



# **ESCUELA POLITÉCNICA NACIONAL**

## **FACULTAD DE CIENCIAS**

**MODELOS ESTADÍSTICOS PARA LA DETECCIÓN DE  
PATRONES EN MEDIO AMBIENTE Y ECONOMÍA  
DETECCIÓN SEMIAUTOMÁTICA PARA DETERMINAR  
ZONAS CON DERRAMES DE PETRÓLEO UTILIZANDO  
DATOS SAR DEL SENTINEL-1 MEDIANTE ALGORITMOS  
NO SUPERVISADOS PARA CLASIFICACIÓN DE  
IMÁGENES**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERA  
MATEMÁTICA**

**KATHYA VIVIANA GAVILANES GUERRERO**

[viviana.gavilanes.g@gmail.com](mailto:viviana.gavilanes.g@gmail.com)

**DIRECTOR: MIGUEL ALFONSO FLORES SÁNCHEZ**

[miguel.flores@epn.edu.ec](mailto:miguel.flores@epn.edu.ec)

**DMQ, FEBRERO 2022**



## **CERTIFICACIONES**

Yo, KATHYA VIVIANA GAVILANES GUERRERO, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

---

Kathya Viviana Gavilanes Guerrero

Certifico que el presente trabajo de integración curricular fue desarrollado por Kathya Viviana Gavilanes Guerrero, bajo mi supervisión.

---

Miguel Alfonso Flores Sánchez  
**DIRECTOR**



## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el(los) producto(s) resultante(s) del mismo, es(son) público(s) y estará(n) a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Kathya Viviana Gavilanes Guerrero

Miguel Alfonso Flores Sánchez



## **DEDICATORIA**

A Katy y Gino, mis queridos padres, que con su amor incondicional supieron guiar mi vida y apoyarme en todo momento.

A mi abuelita Susana que siempre me motivó a ser valiente.

A mis hermanos Esteban y Elizabeth.

A la niña que soñaba con ser astronauta.

## **AGRADECIMIENTO**

Agradezco a mis padres (Katy Guerrero y Gino Gavilanes) por no dejar que me rinda y proporcionarme los recursos necesarios para estudiar. Gracias por todo el apoyo y amor sin condiciones que pudieron demostrarme.

A mi casa de estudios, la Escuela Politécnica Nacional, por brindarme la oportunidad de estudiar ingeniería matemática.

A mi tutor académico, Miguel Alfonso Flores Sánchez, Ph.D., por confiar en mi potencial, por sus sabios consejos en momentos de indecisión y su tiempo invertido para llevar a cabo el desarrollo de este trabajo.

A mis profesores que supieron guiarme desde el inicio hasta el final de mi carrera. En especial a Polo Vaca, Ph.D.; María Fernanda Salazar, Ph.D.; Fernando Cortez, Ph.D.; Menthor Urvina, Msc.; Ramón Xulvi, Ph.D.; Marco Calahorrano, Ph.D.; Holger Benálcazar, Msc.; Ramiro Torres, Ph.D.; Miguel Yangari, Ph.D.; Luis Miguel Torres, Ph.D.; David Pazmiño, Ph.D., Adriana Uquillas, Ph.D., Carlos Almeida, Ph.D. y Holger Capa, Ph.D.

A Diego Chamorro por expandir mis horizontes y motivarme a apuntar a lo más alto.

A mis amigos que acompañaron mi camino de estudio, en especial a Iván Proaño. Y a mis amigas de toda la vida: Andre Navarrete y Pame Alpala.

A Any Escobar por compartir algunas ideas para la resolución del problema.

A Alexandra Elbakyan por crear Sci-hub y a Guido van Rossum, por Python.



## RESUMEN

Los derrames de petróleo causan perjuicios graves en la naturaleza. La literatura revisada apuesta porque la detección temprana y vigilancia de derrames es la gestión más importante. Resulta trascendental de forma precisa el momento y lugar exactos en el que se producen. Además, conocer qué tipo de hidrocarburo es derramado puede ayudar a los responsables de la toma de decisiones a elegir un método adecuado para su limpieza [10]. La teledetección es considerada uno de los métodos más eficaces para la detección de derrames de petróleo [13]. La teledetección SAR se basa en la interacción de las ondas electromagnéticas del radar con las ondas de materia de la superficie del mar. Los sensores activos de teledetección superan las limitaciones de los sensores pasivos antes las condiciones meteorológicas y temporales. Este es el motivo por el que se seleccionan las imágenes satelitales de las misiones que captura el Sentinel-1 y que se encuentran disponibles en los repositorios de la Agencia Espacial Europea (ESA), para extraer estas imágenes se recopiló una base de datos de registros de incidentes de derrames de petróleo con su ubicación y fecha exacta de ocurrencia. Por otro lado, se propone un método para el preprocesamiento de las imágenes satelitales con especial atención en el tratamiento del ruido speckle. Luego, se calcula la matriz de co-ocurrencia y las texturas de segundo orden mediante el uso de SNAP enlazado a Python. A partir de las características extraídas se realiza un Análisis de Componentes Principales para reducir la dimensión del conjunto de características y por último se aplica el algoritmo de aprendizaje no supervisado  $k$ -means para agrupar píxeles.

**Palabras clave:** teledetección, imágenes SAR, derrames de petróleo, Sentinel-1, k-means, ACP.

## **ABSTRACT**

Oil spills cause serious damage to nature. The literature reviewed argues that early detection and monitoring of spills is the most important management. It is important to know precisely the exact time and place of spills. In addition, knowing what type of oil is spilled can help decision makers choose an appropriate method for cleanup. Remote sensing is considered one of the most effective methods for detecting oil spills. SAR remote sensing is based on the interaction of electromagnetic waves from radar with matter waves from the sea surface. Active remote sensing sensors overcome the limitations of passive sensors in the face of weather and time conditions. This is the reason for selecting the satellite images of the missions captured by Sentinel-1 and available in the repositories of the European Space Agency (ESA). To extract these images, a database of oil spill incident records with their location and exact date of occurrence was compiled. On the other hand, a method for the preprocessing of satellite images is proposed with special attention to the treatment of speckle noise. Then, the co-occurrence matrix and second order textures are calculated using SNAP linked to Python. From the extracted features a Principal Component Analysis is performed to reduce the dimension of the feature set and finally the unsupervised learning algorithm  $k$ -means is applied to cluster pixels.

**Keywords:** Sentinel-1, remote sensing, SAR imaging, oil spills, k-means, PCA

---

# Índice general

---

<b>1. Descripción del componente desarrollado</b>	<b>1</b>
1.1. Objetivo general . . . . .	2
1.2. Objetivos específicos . . . . .	2
1.3. Alcance . . . . .	3
1.4. Marco teórico . . . . .	4
1.4.1. Estatus de los derrames de petróleo . . . . .	4
1.4.2. Introducción a la Teledetección . . . . .	5
1.4.3. Preprocesamiento imágenes SAR . . . . .	8
1.4.4. Matriz de co-ocurrencia de Nivel Gris . . . . .	10
1.4.5. Algoritmos no supervisados . . . . .	13
1.4.6. Análisis de Componentes Principales (ACP) . . . . .	13
1.4.7. Clústering: k-means . . . . .	14
<b>2. Metodología</b>	<b>16</b>
2.1. Base de registros . . . . .	16
2.2. Extracción de imágenes satelitales . . . . .	17
2.3. Preprocesamiento de imágenes satelitales mediante SNAP . . . . .	19
2.3.1. Método semiautomático . . . . .	20
<b>3. Resultados, conclusiones y recomendaciones</b>	<b>32</b>

3.1. Resultados . . . . .	32
3.1.1. Búsqueda de derrames de hidrocarburos . . . . .	35
3.1.2. Descripción de los derrames de hidrocarburos . . . . .	36
3.1.3. Caso: Kuwait . . . . .	40
3.2. Conclusiones y recomendaciones . . . . .	46
3.2.1. Conclusiones . . . . .	46
3.2.2. Recomendaciones . . . . .	48
<b>A. Anexo</b>	<b>49</b>
A.1. Análisis de la base Raw Incident Data . . . . .	49
A.1.1. Lectura de la base de datos . . . . .	49
A.1.2. Tratamiento de los datos de la base . . . . .	51
A.1.3. Base final . . . . .	54
A.2. Gráficos ilustrativos . . . . .	55
A.3. Diccionario de variables . . . . .	59
A.4. Base oil_data . . . . .	61
A.5. Imágenes preprocesamiento . . . . .	62
A.6. Códigos . . . . .	64
<b>Bibliografía</b>	<b>72</b>

---

## Índice de figuras

---

1.1. ITOPF estadísticas. . . . .	4
2.1. Librerías a utilizar . . . . .	20
2.2. Comando <code>gpt - h</code> . . . . .	21
2.3. Pasos para el preprocesamiento de datos SAR y creación de clústers . . . . .	21
2.4. Visualización rápida de la imagen satelital . . . . .	21
2.5. Código para aplicar <code>orbit file</code> . . . . .	22
2.6. Subset con las bandas VV y VH . . . . .	23
2.7. Código para aplicar el filtrado <code>speckle</code> . . . . .	24
2.8. Imagen de $7 \times 7$ con 8 niveles grises . . . . .	24
2.9. Código para realizar el Análisis de Componentes Principales	29
2.10.Código para calcular el número de clústers óptimo . . . . .	30
2.11.Código para aplicar el algoritmo $k$ -means. . . . .	31
3.1. Función para filtrar desde una fecha mínima hasta una má- xima . . . . .	33
3.2. Mapa de los registros derrames de hidrocarburos . . . . .	34
3.3. Histograma del tipo de derrame de hidrocarburos . . . . .	34
3.4. Información sobre las imágenes SAR obtenidas . . . . .	36

3.5. Caso 1: Erupción volcánica en Tonga provoca un derrame de petróleo en Perú . . . . .	37
3.6. Derrame del pozo de petróleo Quarter North, Luisiana, Golfo de México . . . . .	37
3.7. Fuga de gasóleo en el depósito de día, Kivalina, AK . . . . .	38
3.8. Derrame de combustible diesel de NRG Energy, Dagsboro, DE	39
3.9. Derrame de combustible diesel de NRG Energy, Dagsboro, DE	39
3.10.Derrame de combustible diesel de NRG Energy, Dagsboro, DE	40
3.11.Izquierda: imagen SAR sin tratar. Derecha: resultado luego de aplicar Orbit File, quitar ruido térmico, calibrar y corregir el terreno . . . . .	41
3.12.Izquierda: Histograma de la intensidad VV. Derecha: Histograma de la intensidad VV en escala logarítmica. . . . .	41
3.13.De izquierda a derecha: subset, filtrado speckle y transformación logarítmica. . . . .	42
3.14.Intensidad VV e Intensidad VH. . . . .	42
3.15.Contraste, Disimilitud, Homogeneidad, Energía y ASM para la Intensidad VV . . . . .	43
3.16.Máxima probabilidad, Entropía, Media GLCM, Varianza GLCM y Correlación GLCM para la Intensidad VV. . . . .	43
3.17.Contraste, Disimilitud, Homogeneidad, Energía y ASM para la Intensidad VH. . . . .	43
3.18.Máxima probabilidad, Entropía, Media GLCM, Varianza GLCM y Correlación GLCM para la Intensidad VH. . . . .	44
3.19.Análisis de ACP: varianza explicada acumulada . . . . .	45
3.20.Simulación del score de silueta para algunos valores de $k$ . . . . .	45
3.21.Clústers encontrados en la imagen . . . . .	46
A.1. Datos faltantes en las base de datos Raw Incident Data . . . . .	51
A.2. Representación gráfica de la longitud de onda y de la frecuencia . . . . .	55

A.3. Representación gráfica pasivos vs. activo . . . . .	56
A.4. Magnitudes radiométricas medidas por el radar ( $\theta$ es el ángulo de observación del sensor) . . . . .	56
A.5. Búsqueda del área en la que ocurrió el incidente . . . . .	62
A.6. Selección de la misión en el área de interés . . . . .	62
A.7. Output de <code>gpt -h</code> . . . . .	63
A.8. Lectura de las imágenes SAR . . . . .	64
A.9. Código para visualización rápida . . . . .	65
A.10Código para subset de la imagen SAR . . . . .	65
A.11Código para la corrección del terreno . . . . .	66
A.12Código para aplicar la calibración radiométrica . . . . .	67
A.13Código para remover el ruido térmico . . . . .	67
A.14Visualización de los datos de Oil Data . . . . .	68
A.15Código para crear el mapa de registros de derrames de petróleo (parte 1) . . . . .	69
A.16Código para crear el mapa de registros de derrames de petróleo (parte 2) . . . . .	70
A.17Lista de las matrices de textura . . . . .	71

# Capítulo 1

---

## Descripción del componente desarrollado

---

Los derrames de petróleo causan perjuicios graves en los ecosistemas marinos, afectan la vida de las personas que habitan cerca de los incidentes, obstaculizan el turismo y complican la actividad pesquera. Casi todos los estudios, que se han realizado sobre la posible solución a esta problemática, apuntan a que la detección temprana y vigilancia de derrames es la gestión más importante. Resulta trascendental de forma precisa el momento y lugar exactos en el que se producen. Además, conocer qué tipo de hidrocarburo es derramado puede ayudar a los responsable de la toma de decisiones a elegir un método adecuado para su limpieza [10].

La teledetección es considerada uno de los métodos más eficaces para la detección de derrames de petróleo [13]. La teledetección SAR se basa en la interacción de las ondas electromagnéticas del radar con las ondas de materia de la superficie del mar. Los sensores activos de teledetección, como el radar de apertura sintética (SAR) superan las limitaciones de los sensores pasivos antes las condiciones meteorológicas.

Las imágenes satelitales de las misiones del Sentinel-1 se pueden encontrar en algunas plataformas SAR disponibles distribuidas por la Agencia Espacial Europea (ESA), para extraer estas imágenes se necesita una base de datos de registros de incidentes de derrames de petróleo con su ubicación y fecha exacta de ocurrencia. Luego, se crea un procedimiento para el tratamiento de estas imágenes SAR mediante la plataforma SNAP enlazada a Python. Se presta especial atención al ruido speckle que es



inherente a la estructura de un SAR y mediante técnicas de filtrado, se elimina este ruido. Luego, se calcula matriz de co-ocurrencia y las características de segundo orden para realizar un ACP y reducir la dimensión de la matriz. Por último, se realiza un clústering mediante la técnica de  $k$ -means++ para identificar y clasificar los píxeles de la imagen en grupos.

## **1.1. Objetivo general**

Desarrollar un método para determinar las zonas con derrames de petróleo utilizando imágenes satelitales extraídas de los repositorios distribuidos por la European Space Agency (ESA) mediante algoritmos no supervisados programados en Python.

## **1.2. Objetivos específicos**

1. Recopilar el conjunto de datos de registros de derrames de petróleo y sus derivados a través de Resource Watch para filtrar los casos de interés y descargar las imágenes satelitales capturadas por el Sentinel-1 distribuidas por la Agencia Espacial Europea (ESA) a través de diversos repositorios como Copernicus Open Acces Hub, NASA Earth Data, Geodata Mundi, entre otras.
2. Establecer la secuencia de pasos para el preprocesamiento de las imágenes mediante técnicas de tratamiento de imágenes satelitales utilizando SNAP y Python para eliminar la interferencia speckle.
3. Extraer las texturas de la matriz de co-ocurrencia de nivel gris (GLCM) (disimilaridad, entropía, correlación, media y varianza) utilizando la plataforma SNAP.
4. Aplicar un análisis de componentes principales para reducir las capas de la imagen GLCM.
5. Agrupar zonas de la imagen mediante el algoritmo de clasificación no supervisada K-means.

### 1.3. Alcance

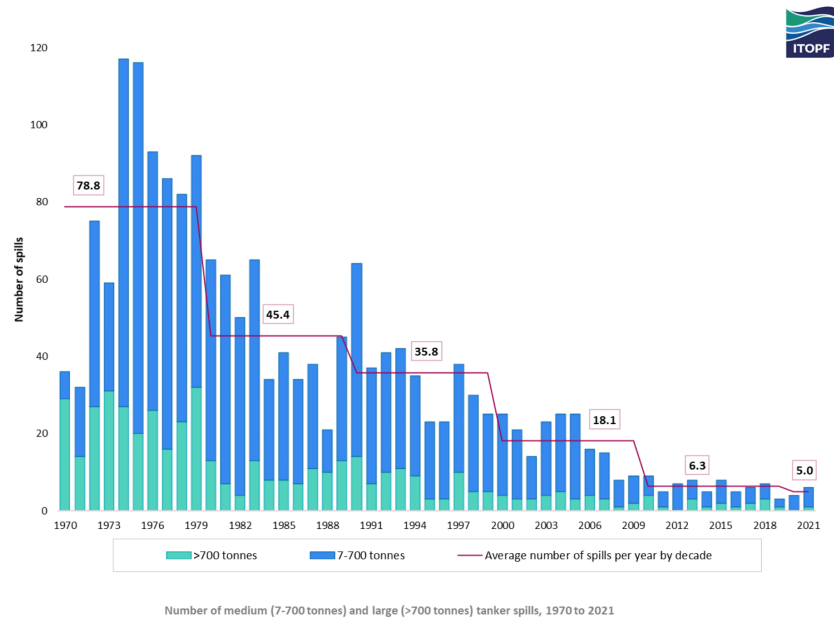
Las aplicaciones de la teledetección están asociadas a múltiples áreas como biología, geología, agronomía, oceanografía, arqueología, entre otras. La observación y análisis de las imágenes SAR proporcionan información realmente valiosa sobre el superficie terrestre del planeta que pueden servir para la toma de decisiones en ámbitos como la agricultura, protección de la atmósfera, tratamiento de suelos, calidad del agua, efectos de desastres naturales, incendios forestales y demás eventos medioambientales que puedan ocurrir. Resulta realmente favorable que se puedan descargar datos SAR de forma gratuita de los repositorios de la ESA así como utilizar plataformas para el tratamiento de estos datos.

En este caso, se tiene especial interés en identificar derrames de hidrocarburos utilizando los datos SAR extraídos de Copernicus Open Access Hub mediante su preprocesamiento en la plataforma SNAP enlazada a Python. Como paso inicial se recopila un conjunto de registros de incidentes asociados a derrames de hidrocarburos que son clasificados de acuerdo al tipo de vertido. Luego, se especifica el procedimiento para descargar las imágenes satelitales asociadas a cada incidente y se crea un método para el tratamiento y la eliminación del inherente ruido speckle.

Posteriormente, a partir de la imagen tratada, se calcula la matriz de co-ocurrencia que permitirá extraer las características que más información condensan de la imagen. Como el orden de estas matrices de características es muy alto y guardan una fuerte correlación, se propone realizar un Análisis de Componentes Principales para utilizar las componentes que acumulen el 99% de varianza. Con esto se busca reducir el número de variables a utilizar sin perder la información. Luego, se realiza un clústering mediante  $k$ -means para detectar a qué grupo pertenece cada píxel y el número óptimo de clústers es determinada mediante el score de Silueta.

## 1.4. Marco teórico

### 1.4.1. Estatus de los derrames de petróleo



Fuente: ITOPF

Figura 1.1: ITOPF estadísticas.

El histórico de derrames de crudo es analizado anualmente por la ITOPF<sup>1</sup>. En la Figura 1.1 se muestra que en 2021 se registraron seis derrames de petróleo de más de 7 toneladas en incidentes con petroleros. Así, se ve un pequeño aumento con respecto al 2020. Sin embargo, se conserva entre la media de la década de 2010. Además, se observa una tendencia decreciente de la cantidad de galones de crudo derramados.

El volumen de petróleo perdido por derrames en 2021 fue de aproximadamente 10 000 toneladas y a pesar de ciertas variaciones anuales, el número y el volumen de los derrames tienden a disminuir desde que se comenzaron a controlar más exhaustivamente estos incidentes, por lo que se debe continuar con procesos preventivos, de detección temprana y de tratamiento.

<sup>1</sup>ITOPF ha respondido a más de 800 incidentes relacionados con derrames de petróleo o productos químicos en todo el mundo.

## 1.4.2. Introducción a la Teledetección

La Teledetección es una técnica de adquisición y tratamiento de datos de la superficie de la tierra empleando las plataformas espaciales que poseen sensores que originan una interacción electromagnética entre el sensor y la tierra. La fuente de radiación proviene del sol o del sensor mismo [4].

Los sistemas de teledetección se componen por un foco energético, la superficie terrestre y un sensor.

Las formas de adquirir información por medio de un sensor son por reflexión que se deriva directamente de la luz solar, emisión o emisión-reflexión [4, 7]. El sol al iluminar la superficie terrestre refleja esa energía de acuerdo al tipo de objeto sobre ella. El sensor transmite el flujo de energía a las estaciones receptoras. La atmósfera ocasiona ruido en este flujo pues dispersa y absorbe parte de la señal original.

Los sensores remotos se clasifican de acuerdo a procedimiento para recibir la energía procedente de la superficie de la tierra, estos son [4]:

- Pasivo: sensor que recibe la energía proveniente de un foco exterior como el sol (limitados).
- Activo: sensor capaz de emitir su propio haz de energía. Estos sensores son más flexibles al no depender solo de un foco exterior.

La energía que dispone la fuente que ilumina al objeto (a menos que la energía detectada sea emitida por el objetivo) se presenta en forma de radiación electromagnética. De acuerdo a la teoría de la onda, se conoce que la energía electromagnética se transmite siguiendo un modelo continuo y armónico y contiene un campo eléctrico (E) y un campo magnético (M) de fuerzas ortogonales entre sí. Ambos campos se desplazan a la velocidad de la luz (c) [7].

Para entender la teledetección se deben considerar dos características intrínsecas de la radiación electromagnética:

- Longitud de onda que es la longitud de un ciclo de onda que puede medirse como la distancia entre los picos de ondas sucesivas. Se

denota como  $\lambda$  a la longitud de onda y se mide en metros (Figura A.2).

- Frecuencia “se refiere al número de ciclos de una onda que pasa por un punto fijo por unidad de tiempo. Se denota como  $f$  y se mide normalmente en hertz (Hz)”. [7](Figura A.2).

La longitud de onda y la frecuencia están relacionadas mediante la fórmula [7]:  $c = \lambda f$ , donde  $\lambda$  es la longitud de onda [m],  $f$  es la frecuencia (ciclos por segundo, Hz) y  $c$  simboliza velocidad de la luz ( $3 \times 10^8 [\frac{m}{s}]$ ).

En esta ecuación, se observa que las dos características se relacionan de forma inversa. Así, mientras más corta sea la longitud de onda, mayor será la frecuencia. Comprender estas características resulta vital para entender la información que se puede extraer de los datos de teledetección ya que se puede definir cualquier tipo de energía en función de su longitud de onda.

## **Espectro electromagnético**

El espectro electromagnético se compone por una sucesión continua de los valores de frecuencia que corresponde a lo que se conoce como bandas en las que la radiación electromagnética presenta un comportamiento similar. Organiza las bandas de longitudes de onda o frecuencia desde las ondas más cortas (gamma, rayos X) hasta las kilométricas. Un resumen de las bandas espectrales se observa en el Cuadro A.1 [7].

Las microondas son las bandas relevantes para el presente estudio y las que más se emplean para la teledetección se observan en el Cuadro A.2. La mayor virtud de esta zona del espectro electromagnético es que es prácticamente transparente a la influencia atmosférica lo que permite recoger información sobre regiones del planeta donde existe mucha nubosidad [4]. La rugosidad de la superficie influye de forma directa en la intensidad de la señal retornada. Esta propiedad depende tanto de la estructura de la superficie terrestre como de la longitud de onda observada.

La importancia de esta relación radica en que cuando la superficie aparece como rugosa, entonces la dispersión retornada se dirige en todas las direcciones y el sensor capta una señal fuerte. Por otro lado, si la

superficie es lisa, la reflexión tenderá a ser especular<sup>2</sup> y en este caso la señal retornada tiende a ser nula, esto se puede observar en el agua del mar en calma que aparece con tonalidad oscura en una imagen radar [4].

## **Polarización**

La polarización se refiere a la orientación del campo eléctrico [4][7]. Existen cuatro combinaciones para transmitir y recibir polarización que se detallan a continuación:

- Combinaciones polarizadas
  - HH - transmisión-recepción horizontal.
  - VV - transmisión-recepción vertical.
- Combinaciones con polarización cruzada
  - HV - transmitida de forma horizontal y recibida verticalmente.
  - VH - transmitida de forma vertical y recibida horizontalmente.

## **Ecuación del radar**

Un radar puede ser modelado por la llamada ecuación de radar que relaciona la potencia recibida con la potencia transmitida en presencia de un objetivo caracterizado por la sección eficaz al radar (Radar Cross Section)  $\sigma^0$ . La ecuación determinada para los SAR es:

$$P_r = P_e \frac{G^2 \lambda^2 \sigma^0}{(4\pi)^3 R^4 a}, \quad (1.1)$$

“donde  $P_r$  es la potencia recibida,  $P_e$  es la potencia enviada,  $G$  es la ganancia de la antena,  $\lambda$ , la longitud de onda,  $a$ , pérdida debido a la absorción del medio de propagación y  $R$  es la distancia entre la antena y el objetivo.  $R^4$  corresponde a la atenuación geométrica debido a la distancia viajada por el pulso electromagnético enviado. El  $\sigma^0$  es una función compleja que depende de la frecuencia y polarización de la onda incidente”. [11].

---

<sup>2</sup>“La reflexión especular se produce cuando un rayo de luz incide sobre una superficie pulida (espejo) y cambia su dirección sin cambiar el medio por donde se propaga”.

Por otro lado, la potencia recibida es proporcional al brillo del objeto ( $\beta^0$ ) que está relacionado con  $\sigma^0$  y  $\gamma^0$  de la siguiente manera:

$$P \propto I \propto A^2 \propto \beta^0 = \frac{\sigma^0}{\sin(\alpha)} = \frac{\gamma^0}{\tan(\alpha)},$$

“donde  $P$  es la potencia,  $I$  es la intensidad,  $A$ , la amplitud,  $\alpha$  es el ángulo de incidencia local,  $\beta^0$  es la reflectividad promedio de centros dispersores por unidad de área del plano de rango oblicuo”. [8],  $\sigma^0$  describe la reflectividad promedio de los dispersores por unidad de área en el plano local de la superficie observada y  $\gamma^0$  es la reflectividad media de centros dispersores distribuidos por unidad de área del frente de onda incidente (Figura A.4). Además  $\sigma^0$  puede expresarse en decibeles mediante escala logarítmica por  $\sigma^0|_{db} = 10 \log_{10}(\sigma^0)$ , esta representación es útil para controlar el rango dinámico que alcanzan los órdenes de magnitud en potencias de diez del radar [8].

### 1.4.3. Preprocesamiento imágenes SAR

#### Textura de una imagen

La textura se refiere a la posible rugosidad o suavidad de una zona de la imagen. Esta textura nace por la relación que existe entre el tamaño de los objetos y la resolución del sensor. Se distinguen tres tipos de texturas en el Cuadro A.3 [4]. Este criterio es importante para discriminar superficies que pueden presentar un comportamiento espectral uniforme.

Considerando que la textura indica qué tan heterogéneamente aparecen los tonos o niveles grises en una imagen, los índices de textura se enfocan en medir la variación espacial entre los niveles de gris de los píxeles. Los índices se pueden dividir entre los que operan sobre toda la imagen o los que se presentan en ventanas móviles desplazándose entre píxeles y modificando el píxel central.

#### Ruido speckle

Desafortunadamente las imágenes satelitales sufren de ruido aditivo y multiplicativo. El primero surge por el ruido térmico y el segundo por

el ruido speckle [21] que perjudica la calidad de la imagen y dificulta su interpretación, por lo que se necesita aplicar un filtro de reducción de ruido speckle en el preprocesamiento de las imágenes SAR. Este filtro suavizará regiones homogéneas preservando la información de la imagen.

Las imágenes SAR tienen propiedades estadísticas y la mayoría de los modelos estadísticos han evolucionado a partir del modelo de ruido multiplicativo. Esto quiere decir que el ruido varía más rápidamente en las regiones en las que el gris de la imagen cambia más rápido, y el efecto speckle es más grave en las regiones más brillantes. El speckle se puede establecer como un modelo de ruido multiplicativo como el siguiente [24].

$$Z(t) = R(t) \cdot \nu(t), \quad (1.2)$$

donde  $Z(t)$  es la señal afectada por el ruido,  $R(t)$  es la imagen original y  $\nu(t)$  el ruido speckle.

### **Filtrado speckle**

El filtrado es una técnica que busca eliminar el ruido en un área uniforme y preservar la textura sin cambiar las características. Además, busca una mejor apariencia visual. Esta técnica funciona con una ventana móvil sobre cada uno de los píxeles de la imagen para suavizar la imagen [21]. El filtrado de interés en este caso es el Filtrado Lee que “calcula una combinación lineal de la intensidad del píxel central en una ventana con su intensidad. Este filtro se basa en el error cuadrático medio mínimo (MMSE)”. [21]

Para producir una imagen libre de ruido speckle, el filtrado Lee se rige por:

$$\hat{R}(t) = \hat{Z}(t) + \left(1 - \frac{C_\nu}{C_Z}\right) (Z(t) - \hat{Z}(t)), \quad (1.3)$$

donde  $\hat{R}(t)$  es el valor de la imagen después de ser filtrada,  $\hat{Z}(t)$  es el valor medio de  $Z(t)$ ,  $C_\nu$  es el coeficiente de varianza de la imagen afectada por el ruido y  $C_Z$  es el coeficiente de varianza de la imagen libre de ruido [21].



#### 1.4.4. Matriz de co-ocurrencia de Nivel Gris

La matriz de co-ocurrencia de nivel gris (GLCM por sus siglas en inglés *Gray-Level Co-Occurrence Matrix*) es una aproximación estadística para obtener información sobre la posición relativa de píxeles vecinos que se encuentran dentro de una imagen [5]. Esta técnica fue propuesta por Haralick en 1973 para caracterizar la textura de una imagen [17] y da origen a una matriz cuadrada de orden  $N_q$  que expresa la frecuencia relativa con la que dos píxeles aparecen a una determinada distancia y orientación, el primero con una intensidad  $i$  y el segundo con una intensidad  $j$  [2].

La matriz de co-ocurrencia  $P$ , se define como:

$$P(i, j) = \sum_{x=1}^m \sum_{y=1}^n \delta_{i,j}(x, y), \quad (1.4)$$

donde la  $\delta$  de Kronecker está dada por

$$\delta_{i,j}(x, y) = \begin{cases} 1, & \text{si } I(x, y) = i \wedge I(x + d_x, y + d_y) = j, \\ 0, & \text{en otro caso.} \end{cases}$$

$I$  es una imagen de tamaño  $m \times n$  y  $d := (d_x, d_y)$  es el vector desplazamiento que indica la distancia entre el píxel de interés y sus vecinos.

Por lo general, la orientación de una matriz GLCM se considera para cuatro direcciones independientes [2]. Horizontal ( $P_H$ ):  $0^\circ; (d_x, 0)$ , Right Diagonal ( $P_{RD}$ ):  $45^\circ; (d_x, d_y)$ , Vertical ( $P_V$ ):  $90^\circ; (0, d_y)$  y Left Diagonal ( $P_{LD}$ ):  $135^\circ; (-d_x, -d_y)$ .

El ángulo asociado al vector de desplazamiento ocasiona que la matriz sea sensible a la rotación. Para solucionar esto y lograr la invarianza rotacional, se considera la matriz de co-ocurrencia en las cuatro direcciones independientes que resulta al sumar las matrices de co-ocurrencia para  $0^\circ, 45^\circ, 90^\circ$  y  $135^\circ$ .

La matriz resultante es la suma de matrices simétricas, por lo que es más conveniente usar los coeficientes de la matriz diagonal superior o inferior. Así, en lugar de tener una longitud de una matriz de  $N_q \times N_q$ , el

tamaño del se reduce a  $\frac{N_q \times N_q + N_q}{2}$  que acelera el rendimiento del proceso sin afectar el resultado [2].

Ahora, se necesita normalizar la matriz GLCM dividiendo cada entrada de la matriz de co-ocurrencia por la suma de todas las entradas se obtiene una matriz de probabilidad. Se define como  $p(i, j)$  a la  $(i, j)$ -ésima entrada de la matriz de co-ocurrencia normalizada y las probabilidades marginales son:

$$p_x(i) = \sum_{j=1}^{N_q} p(i, j) \text{ y } p_y(j) = \sum_{i=1}^{N_q} p(i, j).$$

La media  $\mu = (\mu_x, \mu_y)$ :

$$\mu_x = \sum_{i=1}^{N_q} i p_x(i) \text{ y } \mu_y = \sum_{i=1}^{N_q} i p_y(i);$$

Por último  $\sigma = \sigma_x, \sigma_y$  las desviaciones estándar de  $p_x$  y  $p_x$  como:

$$\sigma_x = \left( \sum_{i=1}^{N_q} p_x(i)(i - \mu_x)^2 \right)^{1/2}, \text{ y } \sigma_y = \left( \sum_{i=1}^{N_q} p_y(i)(i - \mu_y)^2 \right)^{1/2}.$$

## Extracción de texturas características

Con base en la matriz de co-ocurrencia de nivel gris, se pueden calcular las texturas características de una imagen (segundo orden). En el trabajo de Haralick se definen 14 [17]. Las características pueden clasificarse en tres grupos: de contraste, de ordenación y de estadística.

### ■ Grupo de contraste

- **Contraste:** mide la frecuencia espacial de una imagen calculando la diferencia entre los valores más altos y más bajos del conjunto de píxeles adyacentes. Mide las variaciones locales presentes en la imagen.

$$CON = \sum_{n=0}^{N_q-1} n^2 \left\{ \sum_{i=1}^{N_q} \sum_{j=1}^{N_q} p(i, j) \right\}, |i - j| = n. \quad (1.5)$$

- Disimilitud (DIS): es una medida lineal de las variaciones locales de una imagen.

$$DIS = \sum_{i=1}^{N_q} \sum_{j=1}^{N_q} |i - j| p(i, j). \quad (1.6)$$

- Homogeneidad (HOM) (o momento de diferencia inversa): mide la homogeneidad en la imagen, donde asume valores más grandes para diferencias más pequeñas en el nivel de gris dentro de los elementos del par.

$$HOM = \sum_{i=1}^{N_q} \sum_{j=1}^{N_q} \frac{p(i, j)}{1 + (i - j)^2}. \quad (1.7)$$

El contraste y la homogeneidad de la GLCM están fuertemente correlacionados, pero de forma inversa [14].

#### ■ Grupo de ordenación

- Segundo momento angular (ASM): mide las repeticiones en el par de píxeles. Detecta los desórdenes en las texturas de las imágenes [14].

$$ASM = \sum_{i=1}^{N_q} \sum_{j=1}^{N_q} p(i, j)^2. \quad (1.8)$$

- Máxima Probabilidad (MAX) [1]

$$MAX_p = \max_{ij} p_{i,j} \quad (1.9)$$

- Entropía (ENT): medida de aleatoriedad, que tiene su valor más alto cuando los elementos de la matriz son todos iguales.

$$ENT = - \sum_{i=1}^{N_q} \sum_{j=1}^{N_q} p(i, j) (\log p(i, j)). \quad (1.10)$$

#### ■ Grupo de estadísticas

- Media de GLCM: "Se expresa en términos del GLCM. El valor del píxel se pondera no por su frecuencia de aparición por sí

mismo, sino por su frecuencia de aparición en combinación con un determinado valor de píxel vecino". [9]

$$\mu_i = \sum_{i=1}^{N_q} \sum_{i=1}^{N_q} ip(i, j), \quad \mu_j = \sum_{i=1}^{N_q} \sum_{i=1}^{N_q} ip(i, j), \quad (1.11)$$

- Varianza GLCM: "La varianza en la textura realiza se basa en la media, y la dispersión en torno a la media, de los valores de las celdas dentro del GLCM, por lo que se ocupa específicamente de las combinaciones de píxeles de referencia y vecinos". [9]

$$\sigma_i^2 = \sum_{i=1}^{N_q} \sum_{i=1}^{N_q} p(i, j)(i - \mu_i)^2, \quad \sigma_j^2 = \sum_{j=1}^{N_q} \sum_{j=1}^{N_q} p(i, j)(j - \mu_j)^2 \quad (1.12)$$

- Correlación GLCM: "La textura de correlación mide la dependencia lineal de los niveles de gris con respecto a los de píxeles vecinos". [9]

$$CORR = \sum_{i=1}^{N_q} \sum_{i=1}^{N_q} p(i, j) \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right]. \quad (1.13)$$

#### 1.4.5. Algoritmos no supervisados

Los algoritmos no supervisados buscan extraer información de un conjunto de  $p$  características  $(X_1, \dots, X_p)$  y crear una variable respuesta  $Y$  en la que cada observación se asignará a un subgrupo.

#### 1.4.6. Análisis de Componentes Principales (ACP)

El análisis de componentes principales (ACP) es un método algebraico que construye características dadas como combinaciones lineales. Es una técnica de aprendizaje no supervisado que busca conseguir un conjunto de datos con una cantidad menor de variables no correlacionadas guardando la mayor cantidad de información posible del conjunto original [6].

Sea  $X = \{x_1, \dots, x_j\}$  el conjunto de entrenamiento que contiene  $n$  observaciones, entonces el algoritmo produce componentes principales  $P_k$  que son combinaciones lineales de las características originales  $X$  [6]. Se escriben como

$$P_k = a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_{kn},$$

donde  $\sum_i a_{ki}^2 = 1$ .

Los vectores de las componentes principales se forman de modo que sean ortogonales entre sí. El ACP utiliza la optimización de algún criterio de información, como la maximización de la varianza de la observación proyectada.

El ACP puede realizarse mediante el algoritmo de descomposición del valor singular (SVD). En este paso, el PCA se emplea para reducir la dimensionalidad del conjunto de datos y producir un menor número de variables no correlacionadas que se utilizan como características para una mejor clasificación de los datos [6]. Por lo general, las texturas características (o capas) de una imagen satelital son enormes en número, comprenden una gran cantidad de información redundante y están altamente correlacionadas, por lo que utilizar el ACP para reducir el número de capas resulta útil. Además, debido a la simetría de la matriz original existe una base completa de vectores propios y se obtendrá el nuevo grupo de datos que tendrá las características que contribuyen más a su varianza.

#### **1.4.7. Clustering: k-means**

El algoritmo de clustering  $k$ -means es una técnica de aprendizaje no supervisados que agrupa a  $n$  observaciones en  $k$  clúster o subgrupos. Cada observación pertenecerá al clúster con el centro más cercano minimizando iterativamente la suma cuadrada de una norma. El resultado es un conjunto de conglomerados lo más compacto y bien separado posible [15].

MacQueen propuso el siguiente procedimiento [22]:

1. Se seleccionan aleatoriamente  $K$  centros de conglomerados iniciales.

2. Según la similitud, se calcula la distancia de cada instancia restante a cada centro de clúster y se clasifica en la categoría de centro de clúster más cercano.
3. Se actualiza la posición del centro de cada grupo con base en el nuevo centro del promedio de los objetos pertenecientes a dicho grupo.

Se repite iterativamente 2. y 3. hasta que los centros queden fijos o se alcance un umbral predeterminado. El algoritmo  $k$ -means resuelve un problema de optimización que se define como:

$$\min_{\mathbf{S}} E(\mathbf{c}_i) = \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x}_j \in S_i} \|\mathbf{x}_j - \mathbf{c}_i\|^2, \quad (1.14)$$

donde  $\mathbf{x}_j \in \mathbb{R}^d$  para todo  $j \in \{1, \dots, n\}$  y  $\mathbf{S} = \{S_1, S_2, \dots, S_k\}$  es el conjunto de los grupos formados.

Para determinar el  $k$  óptimo se utiliza el Score de Silueta cuyo valor oscila entre  $-1$  y  $1$ , donde  $-1$  implica que los clúster fueron mal asignados,  $0$  la distancia entre clústers no tiene relevancia y  $1$  cuando los clúster está bien separados entre sí. Este coeficiente se calcula como:

$$S_{sc} = \frac{b - a}{\max(a, b)}, \quad (1.15)$$

donde  $a$  es la distancia media entre cada punto dentro de un clúster y  $b$  es la distancia media entre todos los clústers.

# Capítulo 2

---

## Metodología

---

En esta sección se detallan las actividades que se realizaron para cumplir con los objetivos. Como herramienta se utilizó una computadora portátil ASUS VivoBook con las siguientes características:

Característica	Descripción
Nombre	VivoBook 15_ASUS Laptop X54UF
Procesador	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
RAM instalada	16,0 GB (15,9 GB utilizable)
Tipo de sistema	Sistema operativo de 64 bits, procesador x64
Disco principal	HP SSD S700 500 GB (disco sólido)
Disco esclavo	ST1000LM035-1RK172 (disco duro)
Edición SO	Windows 11 Home Single Language

Cuadro 2.1: Características de la computadora utilizada

### 2.1. Base de registros

“*Raw Incident Data*” es la base de registros de incidentes de vertidos petroleros y de productos químicos que se obtuvo de la página <https://incidentnews.noaa.gov/>. Se accedió a través de *Resource Watch* ([www.resourcewatch.org](http://www.resourcewatch.org)), el 29 de enero de 2022. La base de datos cuenta con 17 columnas y 4189 registros desde el 29 de marzo de 1957. En el Cuadro A.4 se observa el tipo y la descripción de las 17 variables. La base recopila incidentes en los que la Oficina de Respuesta y Restauración

(OR&R)<sup>1</sup> de la Administración Nacional Oceánica y Atmosférica (NOAA) se activó para dar un alerta del suceso. Incluye principalmente las zonas marítimas estadounidenses y varias localidades internacionales. El conjunto de datos es de acceso público y libre.

Se consideran las siguientes variables para extraer la información relevante que permitirá buscar y descargar las imágenes satelitales a analizar.

- `id` que corresponde a la identificación del incidente.
- `open_date` que es la fecha en la que se notificó el derrame.
- `name` que es el nombre del incidente.
- `location` que es la localización del incidente.
- `lat` corresponde al valor aproximado de la longitud en la que se identificó el derrame.
- `lon` corresponde al valor aproximado de la latitud en la que se identificó el derrame.
- `threat` tipo de derrame: “Oil”, “Chemical” u “Other”.
- `commodity` representa el tipo de sustancia que se derramó.

El análisis de esta base de datos para clasificar el tipo de vertido se realizó en el lenguaje de programación R mediante el entorno RStudio (Anexo A.1). En la variable `commodity` se obtuvieron 553 tipos de sustancias derramadas, pero se determinaron 14 categorías generales que clasifican a estos incidentes. Se muestran algunos registros de la base de datos analizada en el Cuadro A.5.

## 2.2. Extracción de imágenes satelitales

En este caso de estudio se utilizarán las imágenes del satélite Sentinel-1 que proporciona datos en la banda C (3.75-7.5 cm). Existen diversas fuentes para descargar imágenes SAR como *Copernicus Open Acces Hub*, *NASA Earth Data*, *Geodata Mundi*, entre otras. La extracción de estas imágenes es gratuita y tiene una resolución lo suficientemente alta para sa-

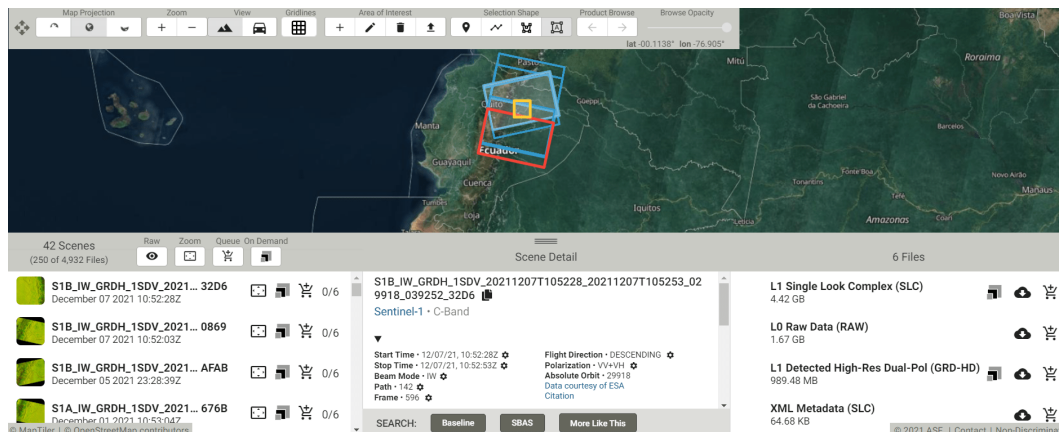
---

<sup>1</sup>OR&R es la organización federal del gobierno de los Estados Unidos con experiencia en la preparación, evaluación y respuesta a las amenazas al medio ambiente costero, incluyendo los derrames de petróleo y químicos.



tisfacer la necesidad del proyecto. Se utilizará el *Alaska Satellite Facility* (ASF) para descargar los datos SAR (este proceso es igual para todas las plataformas).

1. Crear las credenciales de inicio de sesión en el *NASA Earth Data*.
2. Abrir la plataforma de búsqueda de datos del ASF e iniciar sesión.
3. Seleccionar el área geográfica de interés utilizando la longitud, latitud y fecha de ocurrencia recopilados en la base de registros.



Al buscar, aparece la lista de las imágenes disponibles. En la Figura 3 se observa la búsqueda de datos SAR en la región amazónica del Ecuador en el 2021.

En la mayoría de casos, puede suceder que no se encuentre la misión del satélite en la fecha exacta de interés. Sin embargo, se puede descargar las misiones posteriores pues un derrame de petróleo puede demorar décadas en desaparecer. Hay ecosistemas que se recuperan de una manera más rápida porque son más dinámicos [16], pero mientras mayor es el derrame, más tardará en descomponerse. Además, en el tiempo de recuperación influyen otros factores como las técnicas de tratamiento o limpieza empleadas [20].

4. Descargar la imagen SAR del Sentinel-1 para el posterior preprocesamiento.
5. Una vez que haya terminado la descarga, se puede ver que los datos descargados están en formato zip. SNAP-Python lee estos datos SAR.

## 2.3. Preprocesamiento de imágenes satelitales mediante SNAP

### Instalación de SNAP-Python

1. Instalar Anaconda [3].
2. Descargar en la página de la ESA <https://step.esa.int/main/download/snap-download/> la opción *All Toolboxes, Windows 64-bit* e instalar.
3. Crear un entorno en la línea de comandos Anaconda Prompt (anaconda3) que se llame snap y que funcione con la versión 3.6 de Python porque esta versión es compatible con el módulo snappy.

```
(base) C:\Users\Personal> conda create -n snap python = 3.6
```

4. Ingresar al ambiente snap.

```
(base) C:\Users\Personal> conda activate snap
```

5. En la línea de comandos SNAP (SNAP Command Line) ejecutar:

```
> snappy-conf C:\Users\Personal\anaconda3\envs\snap\python.exe  
C:\Users\Personal\anaconda3\envs\snap\lib
```

para cambiar el directorio de la configuración de snap.

6. En la línea de comandos de Anaconda Prompt, ejecutar el siguiente código

```
(snap) C:\Users\Personal> python  
>>> import snappy
```

7. Instalar Jupyter Notebook con la versión de python 3.6 en el ambiente snap.

Con este proceso se logra enlazar python a snap [18] para desarrollar el preprocesamiento de las imágenes satelitales.

### 2.3.1. Método semiautomático

La motivación de utilizar snappy para la detección de derrames de petróleo surge porque SNAP tiene capacidad de memoria limitada para el procesamiento de imágenes y además se demora un tiempo considerable. Debido a que SNAP se enlaza a Python, se puede utilizar una GPU de Jupyter Notebook para disminuir el problema de la capacidad de memoria y rendimiento. Se definen las librerías a utilizar (Figura 2.1).

- Importar las librerías

```
1 # Librería                                # Descripción
2 import matplotlib.colors as colors         # para visualizar data
3 import matplotlib.image as mpimg         # para visualizar data
4 import matplotlib.pyplot as plt          # para visualizar data
5     mpimg
6 from termcolor import colored             # Imprimir texto a color
7 from zipfile import ZipFile              # Manipular archivos tipo
8     zip
9 from os.path import join                  # Acceso a la data en
10    folders
11 from glob import iglob                   # Acceso a la data en
12    folders
13 import pandas as pd                      # Análisis de datos y
14    manipulación
15 import numpy as np                       # Cálculo científico
16 import subprocess                        # Llamadas al sistema (
17    terminal)
18 import snappy                             # API de SNAP para Python
19 import jpy                                # Puente entre Python y
20    Java
21
22 # Cambiar ajustes de módulo
23 pd.options.display.max_colwidth = 80     # Texto más largo en pd.df
```

Figura 2.1: Librerías a utilizar

- Operadores disponibles en snappy

Para saber qué operadores están disponibles en snappy, se debe llamar el comando `gpt -h`.

```

1 # Llamar gpt -h desde la línea de comando
2 print(subprocess.Popen(['gpt', '-h', 'Maximum-Likelihood-Classifier'],
    stdout = subprocess.PIPE, universal_newlines = True).communicate()
    [0])

```

Figura 2.2: Comando `gpt - h`

El output se puede observar en la Figura A.7.

Los pasos que se seguirán para el tratamiento de las imágenes satelitales y la creación de clústers se observan en la Figura 2.3.

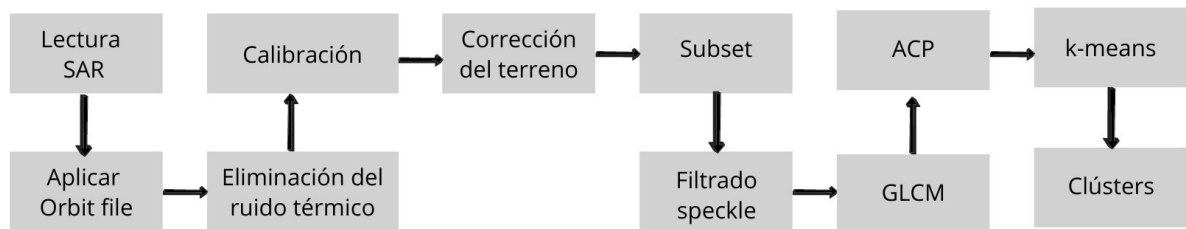


Figura 2.3: Pasos para el preprocesamiento de datos SAR y creación de clústers

## 1. Lectura del producto SAR

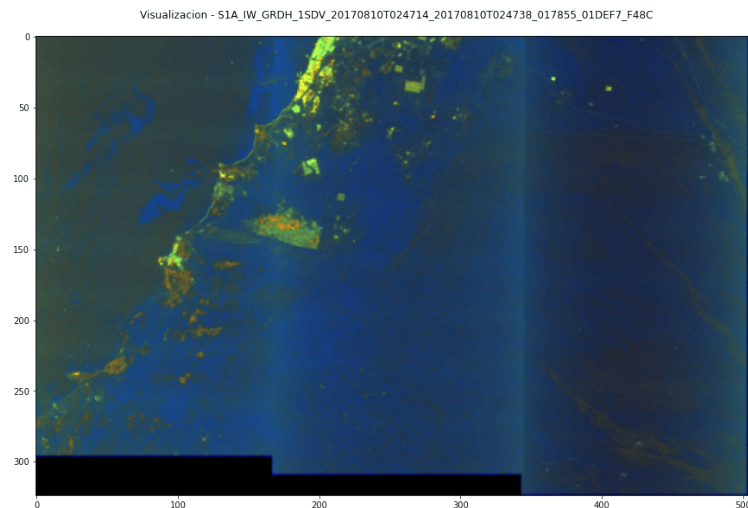


Figura 2.4: Visualización rápida de la imagen satelital

Primero se va a leer los archivos (Figura A.8) que se quieren preprocesar. Luego, se utilizará una imagen SAR para analizarla y será importada con `snappy` que extraerá las bandas  $Amplitude_{VH}$ .

Intensity<sub>VH</sub>, Amplitude<sub>VV</sub> e Intensity<sub>VV</sub> que componen al dato SAR. Además, mediante el comando `mp.img` (Figura A.9 se presenta una visualización de la imagen original (Figura 2.4).

## 2. Aplicar orbit file

Los vectores del estado orbital provistos por la metadata de un producto SAR generalmente no son precisos y pueden ser redefinidos con los archivos del orbital preciso que están disponibles días después de la generación del producto. El archivo orbital provee información precisa de la posición y velocidad satelital y, con base en esta información, los vectores del estado orbital en el extracto de la metadata del producto son actualizados.

```
1 # Aplicar el operador Orbit File - snappy
2 # Se abre la variable de parametros
3 parameters = snappy.HashMap()
4 # Agregamos los parámetros
5 parameters.put('Apply-Orbit-File', True)
6 apply_orbit = snappy.GPF.createProduct('Apply-Orbit-File', parameters,
    subset)
7 print(colored('Actualizacion orbital exitosa', 'green'))
```

Figura 2.5: Código para aplicar orbit file

## 3. Remover ruido térmico

El ruido térmico en las imágenes SAR es la energía de fondo que genera el propio receptor. Muestra los valores más altos de la reflectividad del radar y dificulta la precisión de las estimaciones de la reflectividad del mismo (Figura A.13).

## 4. Calibración radiométrica

El propósito de la calibración del SAR es presentar los valores de píxeles que se relacionen de forma directa con la retrodispersión del radar en la imagen. Cabe destacar que las imágenes SAR que son previamente calibradas resultan esenciales para el uso cuantitativo de los datos SAR que se desarrollará en los siguientes pasos. El procesamiento típico de los datos SAR, que produce imágenes de Level-1, no incluye correcciones radiométricas [12]. La corrección

radiométrica es necesaria para que los valores de los píxeles representen realmente la retrodispersión del radar de la superficie reflectante y, por lo tanto, para la comparación de imágenes SAR adquiridas con diferentes sensores, o adquiridas desde el mismo sensor pero en diferentes momentos, en diferentes modos, o procesadas por diferentes procesadores (Figura A.12).

## 5. Corrección del terreno

Puesto que existen variaciones topográficas y a la inclinación del sensor del satélite, entonces puede existir una distorsión en las imágenes satelitales. Los datos de las imágenes que no se encuentren directamente en la ubicación del punto del sensor van a presentar una distorsión. En este sentido, las correcciones del terreno buscan compensar las distorsiones para que la representación geométrica de la imagen sea lo más parecida posible al mundo real [23] (Figura A.11).

6. **Subset** Luego de leer el producto, se va a extraer un subconjunto (Figura A.10) para reducir el tiempo de procesamiento.

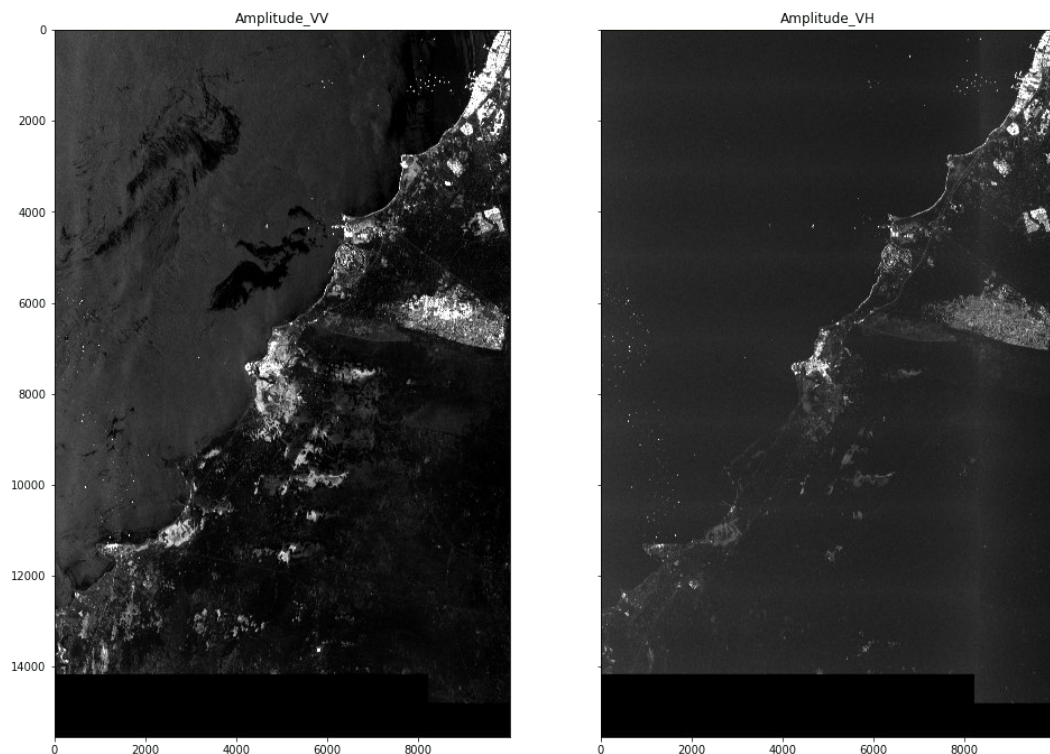


Figura 2.6: Subset con las bandas VV y VH

## 7. Filtrado speckle

Se aplica el filtrado speckle de Lee con un tamaño de ventana de  $10 \times 10$ .

```
1 '''MULTITEMPORAL SPECKLE FILTER'''
2
3 def multitemporal_filter(product):
4     parameters = snappy.HashMap()
5     parameters.put('filter', 'Lee')
6     parameters.put('filterSizeX', 10)
7     parameters.put('filterSizeY', 10)
8     return snappy.GPF.createProduct('Multi-Temporal-Speckle-Filter',
9                                     parameters, product)
10
11 # plot
12 output_bands = ['Sigma0_VV', 'Sigma0_VH']
13 vista_output(speckle, output_bands, 0.00, 0.21, 0.00, 0.048)
```

Figura 2.7: Código para aplicar el filtrado speckle

## 8. GLCM

Luego de obtener la imagen preprocesada en la que se han minimizado los ruidos aditivos y multiplicativos, se procede a calcular la matriz de co-ocurrencia que permitirá extraer las texturas que caracterizan a la imagen satelital. Un ejemplo con un caso reducido se muestra a continuación.

### Cálculo de la matriz de co-ocurrencia

En la Figura 2.8 se observa una imagen de orden 7 que tiene 8 niveles de grises, es decir,  $N_q = 8$ .

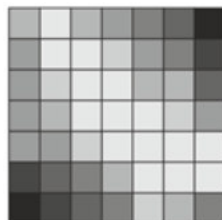


Figura 2.8: Imagen de  $7 \times 7$  con 8 niveles grises

A cada cuadrado de la imagen se le asigna un número entre 1 y  $N_q$  de acuerdo al nivel de gris que le corresponda.

3	1	3	4	5	6	8
4	1	1	2	4	5	7
4	2	1	1	3	3	6
4	3	1	1	1	2	4
4	4	2	1	1	1	1
7	6	5	3	1	1	1
8	7	6	5	2	3	5

Intensidad

Luego, se calculan la matrices de co-ocurrencia  $P_H$  ( $0^\circ$ ),  $P_{RD}$  ( $45^\circ$ ),  $P_V$  ( $90^\circ$ ) y  $P_{LD}$  ( $135^\circ$ ).

Se calcula la matriz  $P_H$  de ocurrencias y la matriz de probabilidad asociadas al desplazamiento horizontal a un paso.

	1	2	3	4	5	6	7	8
1	18	4	5	1	0	0	0	0
2	4	0	1	4	1	0	0	0
3	5	1	2	2	2	1	0	0
4	1	4	2	2	2	0	0	0
5	0	1	2	2	0	3	1	0
6	0	0	1	0	3	0	2	1
7	0	0	0	0	1	2	0	1
8	0	0	0	0	0	1	1	0

**84**

	1	2	3	4	5	6	7	8
1	0.21	0.05	0.06	0.01	0.00	0.00	0.00	0.00
2	0.05	0.00	0.01	0.05	0.01	0.00	0.00	0.00
3	0.06	0.01	0.02	0.02	0.02	0.01	0.00	0.00
4	0.01	0.05	0.02	0.02	0.02	0.00	0.00	0.00
5	0.00	0.01	0.02	0.02	0.00	0.04	0.01	0.00
6	0.00	0.00	0.01	0.00	0.04	0.00	0.02	0.01
7	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.01
8	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00

**1.00**



Se calcula la matriz  $P_{RD}$  de ocurrencias y la matriz de probabilidad asociadas al desplazamiento diagonal derecho a un paso.

	1	2	3	4	5	6	7	8
1	8	5	6	6	2	0	0	0
2	5	0	0	1	1	2	0	0
3	6	0	0	1	1	1	1	0
4	6	1	1	0	0	1	1	0
5	2	1	1	0	0	0	1	1
6	0	2	1	1	0	0	0	1
7	0	0	1	1	1	0	0	0
8	0	0	0	0	1	1	0	0

**72**

	1	2	3	4	5	6	7	8
1	0.11	0.07	0.08	0.08	0.03	0.00	0.00	0.00
2	0.07	0.00	0.00	0.01	0.01	0.03	0.00	0.00
3	0.08	0.00	0.00	0.01	0.01	0.01	0.01	0.00
4	0.08	0.01	0.01	0.00	0.00	0.01	0.01	0.00
5	0.03	0.01	0.01	0.00	0.00	0.00	0.01	0.01
6	0.00	0.03	0.01	0.01	0.00	0.00	0.00	0.01
7	0.00	0.00	0.01	0.01	0.01	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00

**1.00**

Se calcula la matriz  $P_V$  de ocurrencias y la matriz de probabilidad asociadas al desplazamiento vertical a un paso.

	1	2	3	4	5	6	7	8
1	18	5	4	1	1	0	0	0
2	5	0	2	1	1	0	0	0
3	4	2	0	3	2	0	0	0
4	1	1	3	6	1	2	1	0
5	1	1	2	1	0	2	0	0
6	0	0	0	2	2	0	2	0
7	0	0	0	1	0	2	0	2
8	0	0	0	0	0	0	2	0

**84**

	1	2	3	4	5	6	7	8
1	0.21	0.06	0.05	0.01	0.01	0.00	0.00	0.00
2	0.06	0.00	0.02	0.01	0.01	0.00	0.00	0.00
3	0.05	0.02	0.00	0.04	0.02	0.00	0.00	0.00
4	0.01	0.01	0.04	0.07	0.01	0.02	0.01	0.00
5	0.01	0.01	0.02	0.01	0.00	0.02	0.00	0.00
6	0.00	0.00	0.00	0.02	0.02	0.00	0.02	0.00
7	0.00	0.00	0.00	0.01	0.00	0.02	0.00	0.02
8	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00

**1.00**

Se calcula la matriz  $P_{LD}$  de ocurrencias y la matriz de probabilidad asociadas al desplazamiento diagonal izquierdo a un paso.

	1	2	3	4	5	6	7	8
1	22	2	2	0	1	0	0	0
2	2	0	6	1	0	0	0	0
3	2	6	0	3	0	0	0	0
4	0	1	3	4	1	1	0	0
5	1	0	0	1	4	1	0	0
6	0	0	0	1	1	2	1	0
7	0	0	0	0	0	1	2	0
8	0	0	0	0	0	0	0	0

**72**

	1	2	3	4	5	6	7	8
1	0.31	0.03	0.03	0.00	0.01	0.00	0.00	0.00
2	0.03	0.00	0.08	0.01	0.00	0.00	0.00	0.00
3	0.03	0.08	0.00	0.04	0.00	0.00	0.00	0.00
4	0.00	0.01	0.04	0.06	0.01	0.01	0.00	0.00
5	0.01	0.00	0.00	0.01	0.06	0.01	0.00	0.00
6	0.00	0.00	0.00	0.01	0.01	0.03	0.01	0.00
7	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

**1.00**

Luego para lograr la invariancia rotacional, se trabaja con la matriz de co-ocurrencia que suma las cuatro matrices calculadas anteriormente.

	1	2	3	4	5	6	7	8
1	66	16	17	8	4	0	0	0
2	16	0	9	7	3	2	0	0
3	17	9	2	9	5	2	1	0
4	8	7	9	12	4	4	2	0
5	4	3	5	4	4	6	2	1
6	0	2	2	4	6	2	5	2
7	0	0	1	2	2	5	2	3
8	0	0	0	0	1	2	3	0

**312**

	1	2	3	4	5	6	7	8
1	0.21	0.05	0.05	0.03	0.01	0.00	0.00	0.00
2	0.05	0.00	0.03	0.02	0.01	0.01	0.00	0.00
3	0.05	0.03	0.01	0.03	0.02	0.01	0.00	0.00
4	0.03	0.02	0.03	0.04	0.01	0.01	0.01	0.00
5	0.01	0.01	0.02	0.01	0.01	0.02	0.01	0.00
6	0.00	0.01	0.01	0.01	0.02	0.01	0.02	0.01
7	0.00	0.00	0.00	0.01	0.01	0.02	0.01	0.01
8	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00

**1.00**

Este es un caso simple en el que se calcula la matriz de co-ocurrencia para la imagen de la Figura 2.8. Sin embargo, las imágenes satelitales que se procesan tienen una gran dimensión y un número alto de niveles grises, por lo que se utiliza la plataforma SNAP para extraer la matriz de co-ocurrencia de una imagen y las texturas que la caracterizan.

Varios estudios mencionan la elección de ventanas pequeñas puede provocar una pérdida de información importante para la caracterización de la textura y ventanas de tamaño muy grande, pueden acarrear errores en el sistema. Por esta razón se fija la ventana móvil en  $10 \times 10$  píxeles, que se calculan con el vector de desplazamiento en todas las direcciones y con  $d = 4$  pasos. En la sección de resultados se podrá observar las texturas obtenidas.

## 9. ACP

Luego de obtener las texturas: contraste, disimilaridad, homogeneidad, ASM, máxima probabilidad, entropía, media GLCM, varianza GLCM y correlación GLCM tanto para la intensidad  $VV$  como para la  $VH$ , se debe redimensionar las texturas de modo que las matrices de  $N_q \times N_q$  se transformen en vectores  $N_q(N_q) \times 1$ . Así se obtiene la matriz de orden  $N_q^2 \times r$ , con  $r$  el número de texturas extraídas, a la que se le aplicará el ACP para reducir el conjunto de características extraído. Luego, se logra guardar la información más útil en unas pocas capas, lo que acelera el proceso de cálculo. Primero, con ayuda de la librería `sklearn` de Python se normaliza la matriz con la función `MinMaxScaler()` y luego se aplica un ACP a la matriz normalizada. Finalmente, se escogen las componentes principales que acumulen el 99% de la varianza y se guardan los datos en otra matriz.

```
1
2 '''Rescalar '''
3 from sklearn.preprocessing import MinMaxScaler
4
5 normal = MinMaxScaler()
6 dataset= normal.fit_transform(reshapedDataset)
7
8 '''Análisis de componentes principales'''
9 from sklearn.decomposition import PCA
10
11 pca = PCA(n_components= reshapedDataset.shape[1]).fit(dataset)
12 pca_dataset = pca.transform(dataset)
13
14 '''Elegir el primer PC que alcance el 99% de la varianza acumulada'''
15 cumulative_variance = np.cumsum(pca.explained_variance_ratio_)
16 subset99 = cumulative_variance[cumulative_variance<0.99]
17 subset99 = np.append(subset99, cumulative_variance[cumulative_variance
18 >=0.99][0])
19 n_pcs = len(subset99)
20 pc_to_classify = pca_dataset[:, :n_pcs]
```

Figura 2.9: Código para realizar el Análisis de Componentes Principales

## 10. K-means

Para encontrar el número óptimo del  $k$ , es decir, de la cantidad de clústers a considerar, se puede hacer pruebas de ensayo y error. Sin embargo, esto puede resultar tedioso por el tiempo y los recursos computacionales. En este sentido, se plantea un score de silueta para elegir el  $k$  más adecuado. Se debe escoger el  $k$  que maximice el score de la silueta pues esto implica que la separación de los subgrupos es más relevante.

```
1 '''Silueta Score'''
2 k_space = list(range(2,10))
3 results, scores = [], []
4 sample = np.random.choice(range(colxrig), size=100000)
5 for k in k_space:
6     print("Running k-means with k=%d..." % k)
7     model_k = KMeans(k, init='k-means++', algorithm='full')
8     model_k.fit(pc_to_classify)
9     clusters_k = model_k.labels_
10    results.append(clusters_k.reshape(list_bands[0].shape))
11    silhouetteScore = silhouette_score(reshapedDataset[sample,:],
12    clusters_k[sample])
13    scores.append(silhouetteScore)
14    print("Encontró %d clusters con un score de Silueta promedio de %.3
15    f" % (k, silhouetteScore))
```

Figura 2.10: Código para calcular el número de clústers óptimo

La librería scikitlearn ofrece el algoritmo  $k$ -means++ que resuelve el inconveniente de que los resultados dependen de la elección inicial de los centros. Este algoritmo busca optimizar la versión estándar del  $k$ -means porque para la elección de los clústers se basa en la medida de probabilidad de la distribución ponderada. Este método mejora el rendimiento y agrupa de mejor manera los píxeles con menos recursos.

```
1 '''K-MEAN classification'''
2 model = KMeans(n_clusters=7, init='k-means++', algorithm='full')
3 model.fit(pc_to_classify)
4 clusters = model.labels_
```

Figura 2.11: Código para aplicar el algoritmo  $k$ -means.

# Capítulo 3

---

## Resultados, conclusiones y recomendaciones

---

### 3.1. Resultados

La base de datos que se recopiló inicialmente contaba con 4189 registros y 17 variables. Luego del tratamiento de los datos, se obtuvo una base de datos con 2328 registros y 10 variables que fue exportada en la carpeta Resultados con el nombre `oil_data.xlsx` y corresponde a registros de derrames de hidrocarburos que fueron reportados desde el 13 de marzo de 1968 hasta el 28 de enero de 2022.

Con el tratamiento de datos, se logró pasar de 553 categorías en la variable que determina el tipo de derrame a 14 categorías que se pueden observar en el Cuadro 3.1 y en el Anexo A.14.

Commodity	Cantidad	Commodity	Cantidad
asphalt	11	look alikes	3
condensate	14	lube oil	20
crude oil	357	tar balls	23
fuel oil	1126	transformed oil	6
heating oil	19	oil	162
heavy oil	103	otros	335
light crude oil	21	suspected, posible or unkown	128

Siguiendo el proceso de Extracción de imágenes de la sección ante-

rior se puede conseguir los nombres de las imágenes satelitales usando los registros con especial atención en la localización, fecha de registro, longitud, latitud y tipo de derrame como se observa en el Cuadro 3.1

location	open_date	lon	lat	commodity
Hampton, VA, USA	10/27/2019	-76.343628	36.980325	fuel oil
Atlantic Ocean	11/1/2019	-79.6525	25.300278	fuel oil
Lake Charles, LA	11/7/2019	-93.320556	30.131667	crude oil
off Northern CA	11/10/2019	-123.6	38.45	fuel oil
Chassell, MI	3/2/2020	-88.436389	46.98861	suspected, posible or unkown
Area F Rd, Highlands, NJ 07732	2/22/2020	-73.965833	40.4444	fuel oil
Stikine Channel, AK	3/27/2020	-132.5616	56.43701	otros
1317 SW Spokane St, Seattle	3/28/2020	-122.35	47.57	lube oil
Red Bank, NJ	1/20/2022	-74.0644	40.35463	fuel oil
Lima, Peru	1/19/2022	-77.1427	-11.9205	crude oil
Ocracoke, NC	1/28/2022	-76.0012	35.06674	fuel oil

Cuadro 3.1: Subconjunto de `oil_data` con las variables relevantes para las imágenes SAR

Como el Sentinel-1 únicamente proporciona registros desde el 2014, se filtró la base de datos desde el 1 de enero de 2014 hasta el 28 de enero de 2022.

```

1 def select_incident(min_date, max_date, type_incident, raw_data):
2     incident_base = raw_data[(raw_data.threat == type_incident) & (
3         raw_data.open_date >= min_date) & (raw_data.open_date <= max_date)]
4     return(incident_base)

```

Figura 3.1: Función para filtrar desde una fecha mínima hasta una máxima

Se obtiene que existen 1134 derrames de petróleo entre las fechas mencionadas. Finalmente, se filtra de la base `oil_data` únicamente los



derrames de hidrocarburos: crude oil, fuel oil, heavy oil, light crude oil y oil, lo que reduce a la base a 896 derrames de petróleo entre las fechas 2014-01-09 y 2022-01-28.

En Python, se creó un mapa interactivo para visualizar las ubicaciones de los 896 registros de derrames de petróleo y sus derivados como se observa en la Figura 3.2. Este mapa es exportado en la carpeta maps como un archivo html con el nombre `oil_spills_class.html`. El código para desarrollar este gráfico se encuentra en Anexo A.15 y Anexo A.16.



Figura 3.2: Mapa de los registros derrames de hidrocarburos

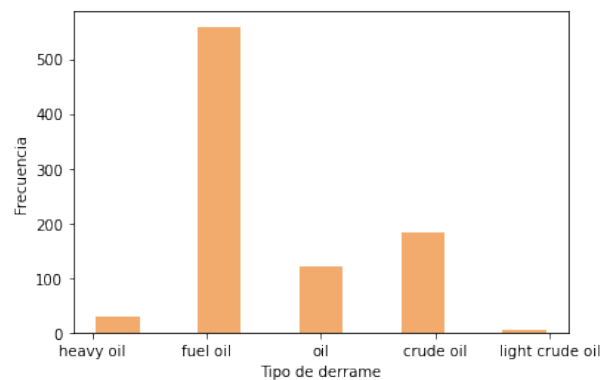


Figura 3.3: Histograma del tipo de derrame de hidrocarburos

Se observa que existen más registros de derrames de “fuel oil” en la base de datos, seguido por “crude oil” y “oil”. La cantidad de reportes de “heavy oil” y “light crude oi” es relativamente baja.

### 3.1.1. Búsqueda de derrames de hidrocarburos

De la base de registros de incidentes de petróleo se seleccionan los casos que se observan en el Cuadro 3.2

id	location	open_date	lon	lat	commodity
10408	Lima, Peru	2022-01-19	-77.142699	-11.9205	crude oil
10403	LA, Gulf of Mexico	2022-01-10	-89.500278	28.97056	crude oil
10398	Kivalina, AK	2021-12-17	-164.53444	67.72556	fuel oil
10352	Dagsboro, DE	2021-09-08	-75.236778	38.58132	fuel oil
10339	1 River Rd, Staten Island, NY 10314, USA	2021-08-18	-74.200287	40.62021	heavy oil
10299	Baytown, TX	2021-06-17	-94.979886	29.71164	light crude oil

Cuadro 3.2: Derrames de hidrocarburos seleccionados

id	SAR_ID	Memoria
10408	S1A_IW_GRDH_1SDV_20220125T104757_20220125T104819_041616_04F352_6EA6	611 MB
10403	S1A_IW_GRDH_1SDV_20220120T000200_20220120T000225_041537_04F09F_CE8A	725 MB
10398	S1B_IW_GRDH_1SDV_20211220T042708_20211220T042733_030104_03982F_5C0E	832 MB
10352	S1A_IW_GRDH_1SDV_20210917T225859_20210917T225924_039728_04B2BC_7424	889 MB
10339	S1A_IW_GRDH_1SDV_20210819T225123_20210819T225148_039305_04A43C_CF9B	949 MB
10299	S1A_IW_GRDH_1SDV_20210621T002641_20210621T002706_038431_0488F1_75AA	877 MB

Cuadro 3.3: Nombre de las imágenes SAR y memoria en MB

Con la plataforma ASF Data Search se determinó el nombre exacto

de la imagen satelital que contiene a cada uno de los derrames y se descargó las imágenes SAR, los nombre y la memoria que requieren para almacenarse en la computadora se observan en el Cuadro 3.3.

Se lee en el script de python los productos y se muestra el detalle de estos en la Figura 3.4 (incluyendo el caso de Kawait).

	Nombre	Modo sensor	Tipo de producto	Polarizacion	Height	Width	Bandas
0	S1A_IW_GRDH_1SDV_20170810T024714_20170810T024738_017855_01DEF7_F48C	GRDH	1SDV	1SDV	15842	25671	[Amplitude_VH, Intensity_VH, Amplitude_VV, Intensity_VV]
1	S1A_IW_GRDH_1SDV_20210621T002641_20210621T002706_038431_0488F1_75AA	GRDH	1SDV	1SDV	16741	25680	[Amplitude_VH, Intensity_VH, Amplitude_VV, Intensity_VV]
2	S1A_IW_GRDH_1SDV_20210819T225123_20210819T225148_039305_04A43C_CF9B	GRDH	1SDV	1SDV	16689	25914	[Amplitude_VH, Intensity_VH, Amplitude_VV, Intensity_VV]
3	S1A_IW_GRDH_1SDV_20210917T225859_20210917T225924_039728_04B2BC_7424	GRDH	1SDV	1SDV	16701	25496	[Amplitude_VH, Intensity_VH, Amplitude_VV, Intensity_VV]
4	S1A_IW_GRDH_1SDV_20220120T000200_20220120T000225_041537_04F09F_CE8A	GRDH	1SDV	1SDV	16743	25711	[Amplitude_VH, Intensity_VH, Amplitude_VV, Intensity_VV]
5	S1A_IW_GRDH_1SDV_20220125T104757_20220125T104819_041616_04F352_6EA6	GRDH	1SDV	1SDV	15386	25250	[Amplitude_VH, Intensity_VH, Amplitude_VV, Intensity_VV]
6	S1B_IW_GRDH_1SDV_20211220T042708_20211220T042733_030104_03982F_5C0E	GRDH	1SDV	1SDV	16650	25641	[Amplitude_VH, Intensity_VH, Amplitude_VV, Intensity_VV]

Figura 3.4: Información sobre las imágenes SAR obtenidas

Todas las imágenes poseen las bandas Amplitude VH, Intensity VH, Amplitude VV e Intensity VV.

### 3.1.2. Descripción de los derrames de hidrocarburos

#### Caso 1: Erupción volcánica en Tonga provoca un derrame de petróleo en Perú

Derrame de “crude oil” registrado en Lima, Perú (Figura 3.5) en las coordenadas lon-latitudinales (-77.142699 -11.9205). En la descripción de la base de datos se expresa que “el 15 de enero de 2022, unas olas inusualmente grandes resultantes de las erupciones volcánicas en tonga llegaron a la costa de Lima (Perú). Estas olas desestabilizaron el petrolero italiano mare doricum mientras trasladaba un cargamento de crudo a la refinería la Pampilla, operada por Repsol. Se estima que se derramaron 6 000 barriles de petróleo”.

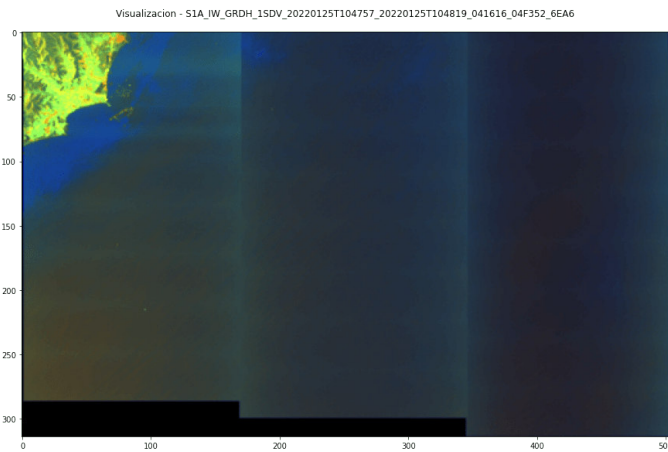


Figura 3.5: Caso 1: Erupción volcánica en Tonga provoca un derrame de petróleo en Perú

**Caso 2: Derrame del pozo de petróleo Quarter North, Luisiana, Golfo de México**

Derrame de “crude oil” registrado en el Golfo de México (Figura 3.6), LA en las coordenadas lon-latitudinales (-89.50027778, 28.97055556). En la descripción de la base de datos se expresa que “el 10 de enero de 2022, la división de gestión de incidentes del sector USCG de Nueva Orleans recibió la notificación de un vertido de petróleo crudo y gas natural procedente de un pozo de quarter north en el Golfo de México. El pozo y la plataforma fueron dañados por el huracán ida el año pasado”.

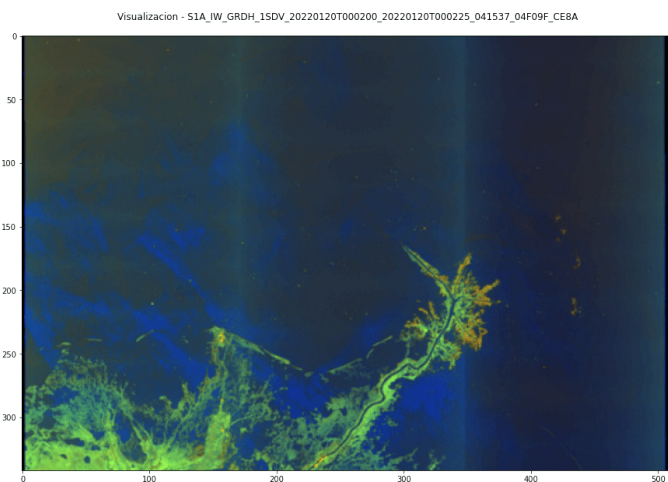


Figura 3.6: Derrame del pozo de petróleo Quarter North, Luisiana, Golfo de México

### **Caso 3: Fuga de gasóleo en el depósito de día, Kivalina, AK**

Derrame de “fuel oil” registrado en Kivalina, AK, en las coordenadas lon-latitudinales (-164.53444, 67.72556 ). En la descripción de la base de datos se expresa que “el 16 de diciembre de 2021, el distrito escolar de northwest arctic borough informó de que se vertieron 1 900 galones de combustible diésel de un tanque diurno debido a un agujero en la tubería. La tubería se encuentra bajo 14 pies de nieve. Se sospecha que el combustible se ha filtrado en el suelo bajo la tubería. USCG sector ancho- rage solicitó apoyo telefónico de NOAA SSC en relación con la respuesta”. Una breve visualización de la imagen se observa en la Figura 3.7.

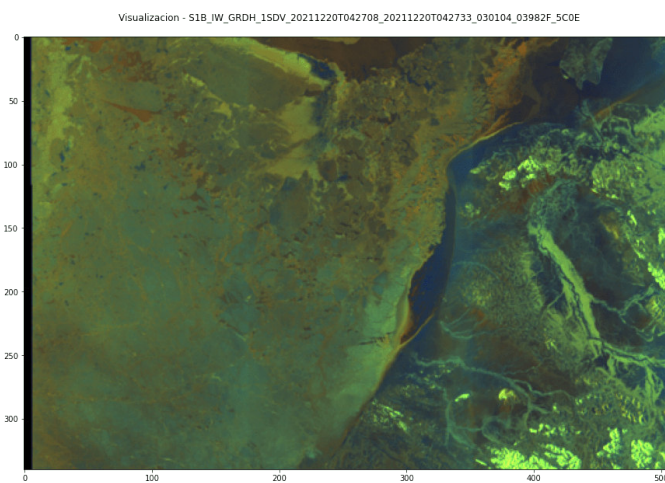


Figura 3.7: Fuga de gasóleo en el depósito de día, Kivalina, AK

### **Caso 4: Derrame de combustible diesel de NRG Energy, Dagsboro, DE**

Derrame de “fuel oil” registrado en Dagsboro (Figura 3.8) en las coordenadas lon-latitudinales (-75.23677826, 38.58131844). En la descripción de la base de datos se expresa que “a las 4:00 P.M. del 8 de septiembre, el USCG del sector de la Bahía de Delaware y el USEPA, notificaron al SSC de la NOAA acerca de un derrame de 26 000 galones de combustible diesel en un oleoducto de las instalaciones de NRG energy, ubicadas adyacentes al Río Indio, Delaware. En este momento, todo el producto ha sido contenido dentro de la contención secundaria, pero el EPA está preocupado por el posible desbordamiento en el Río Indio como resultado de una tormenta que se está moviendo en el área esta noche”.

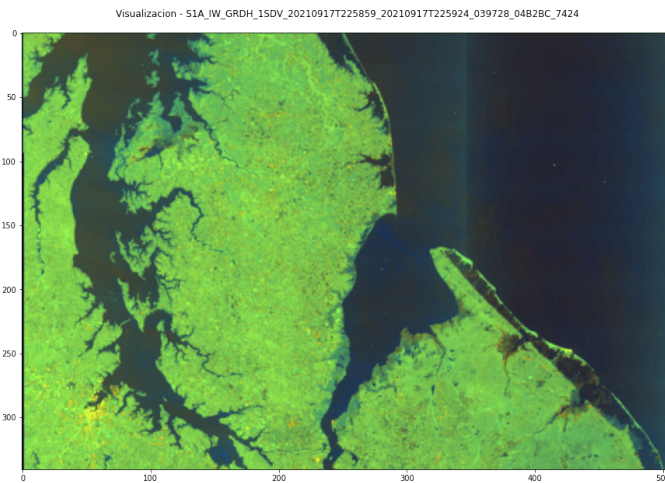


Figura 3.8: Derrame de combustible diesel de NRG Energy, Dagsboro, DE

**Caso 5: Phillips 66 Linden, NJ Heavy Fuel Spill; 1 River Rd, Staten Island, NY 10314, USA**

Derrame de “heavy oil” registrado en Dagsboro (Figura 3.9) en las coordenadas lon-latitudinales (-74.20028687, 40.62020705). En la descripción de la base de datos se expresa que “el 18 de agosto de 2021, el USCG sector new york notificó al SSC de la NOAA que se habían derramado entre 100 y 200 galones de fueloil pesado de una barcaza en el Canal de Arthur Kill durante una operación de abastecimiento de combustible. En la Refinería Phillips 66. No se solicitaron productos. El SSC proporcionó una descripción verbal de los recursos en riesgo”.

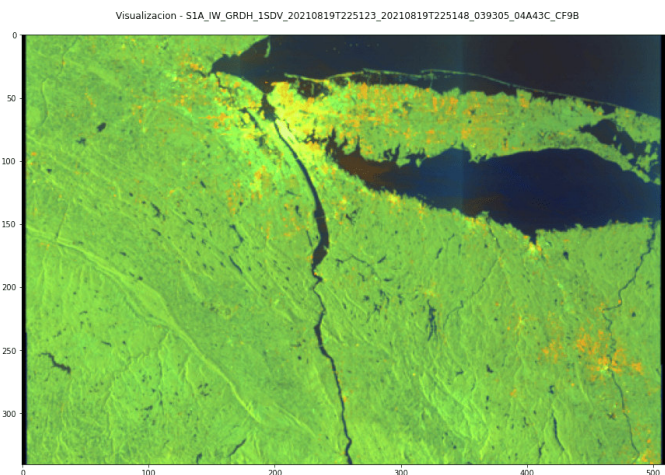


Figura 3.9: Derrame de combustible diesel de NRG Energy, Dagsboro, DE

## Caso 6: Descarga de la zanja de Baytown; Baytown, TX

Derrame de “light crude oil” registrado en Baytown, TX en las coordenadas lon-latitudinales (-94.97988611, 29.71163889). En la descripción de la base de datos se expresa que “el 17 de junio de 2021 el sector houston-galveston se puso en contacto con el SSC de la NOAA para solicitar una evaluación del lavado de la vegetación a baja presión para un crudo ligero estimado de 20-25 barriles vertido en una zanja (NRC#1307083). el vertido es adyacente a las aguas navegables, pero no las amenaza. La recuperación del petróleo está en curso en la zanja/estanque”. Una breve visualización de la imagen se observa en la Figura 3.10.

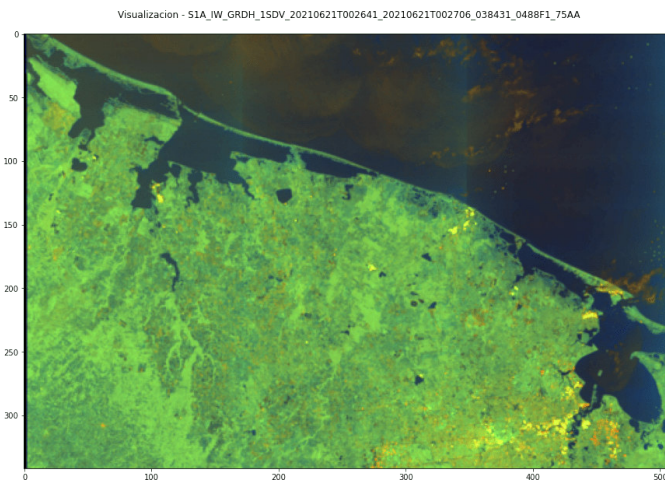


Figura 3.10: Derrame de combustible diesel de NRG Energy, Dagsboro, DE

### 3.1.3. Caso: Kuwait

El 10 de agosto de 2017 ocurrió un gran derrame de petróleo en el sur de Kuwait [19]. La causa del incidente no es clara, pero se filtraron aproximadamente 133000 litros según SkyTruth<sup>1</sup>.

<sup>1</sup>SkyTruth es una organización sin fines de lucro con sede en los Estados Unidos.

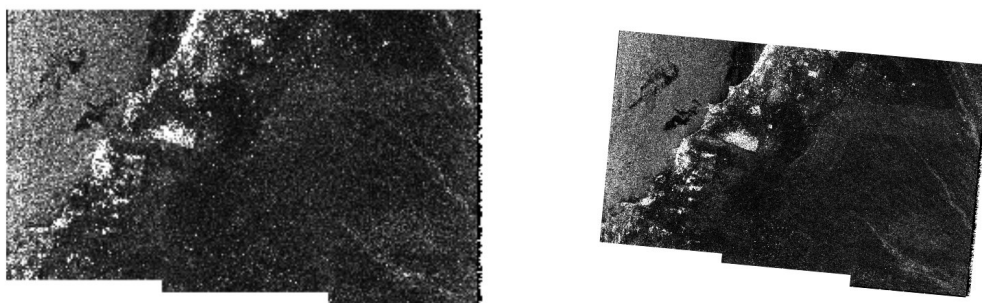


Figura 3.11: Izquierda: imagen SAR sin tratar. Derecha: resultado luego de aplicar Orbit File, quitar ruido térmico, calibrar y corregir el terreno

## Data

- Imagen de Sentinel-1 con fecha 10 de agosto de 2017 extraída de la plataforma Copernicus

La imagen es de tamaño  $25671 \times 15742$ . Se aplica la corrección orbit file, se remueve el ruido aditivo correspondiente al ruido térmico, se realiza la calibración radiométrica y una corrección del terreno. En la Figura 3.11 se observa el cambio.

Luego, se extrae un subconjunto de la imagen original para aplicar el filtrado speckle de Lee y reducir el ruido multiplicativo. En este caso se decidió ocupar una ventana de  $10 \times 10$  por las razones explicadas en la sección de metodología y se realiza una transformación de Intensidad a Decibelios en escala logarítmica con lo que se controlan la escala frecuencia en número de píxeles como se observa en la Figura 3.12.

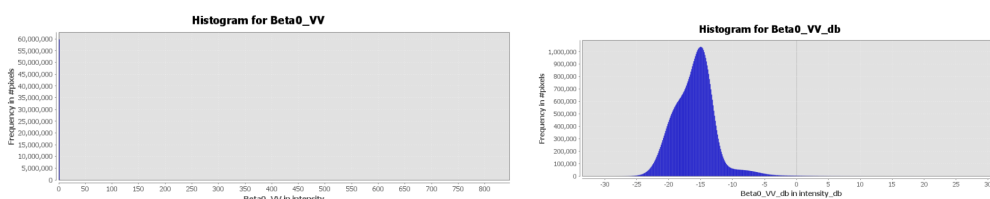


Figura 3.12: Izquierda: Histograma de la intensidad VV. Derecha: Histograma de la intensidad VV en escala logarítmica.



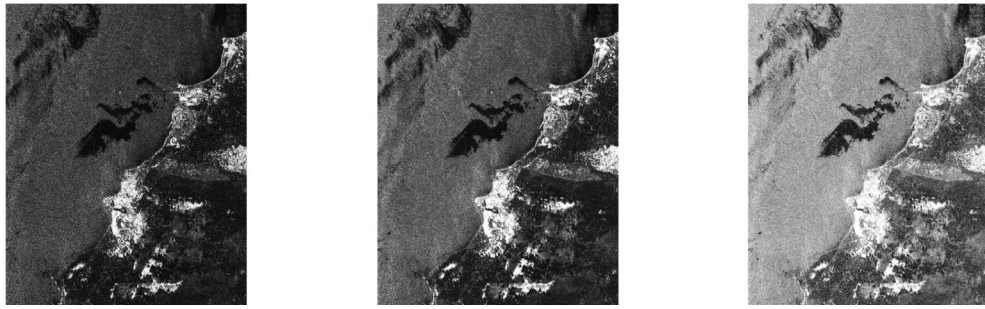


Figura 3.13: De izquierda a derecha: subset, filtrado speckle y transformación logarítmica.

Ahora, se van a extraer las texturas de un subset de la imagen en escala logarítmica mediante el procedimiento propuesto en la metodología. En el subset que se observa en la Figura 3.14, se tiene la Intensidad de polarización VV y la Intensidad de polarización VH que son de tamaño  $2577 \times 2577$  que si bien todavía sigue siendo un número grande ha disminuido considerablemente en comparación a la imagen original.

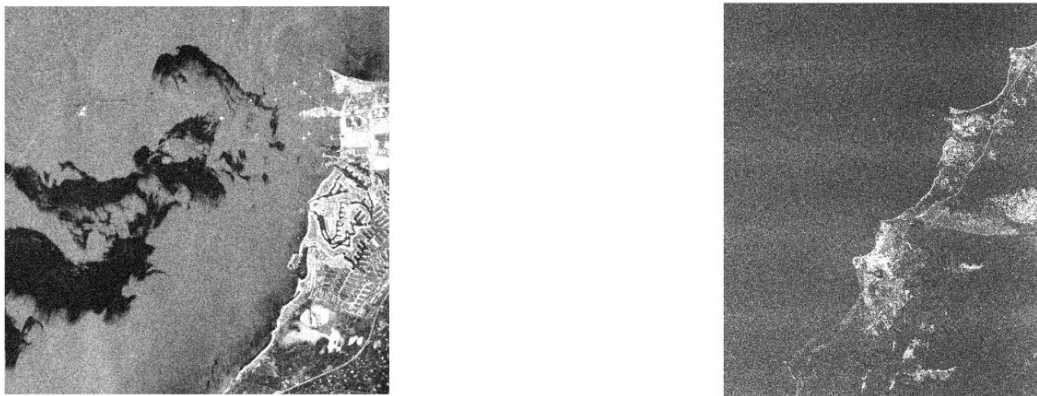


Figura 3.14: Intensidad VV e Intensidad VH.

Con la ayuda de la plataforma SNAP enlazada a Python, se extraen las texturas de las dos intensidades. En la Figura 3.15 y la Figura 3.16 se observan las texturas características para la polarización VV y en las Figuras 3.17 y 3.18, para VH.

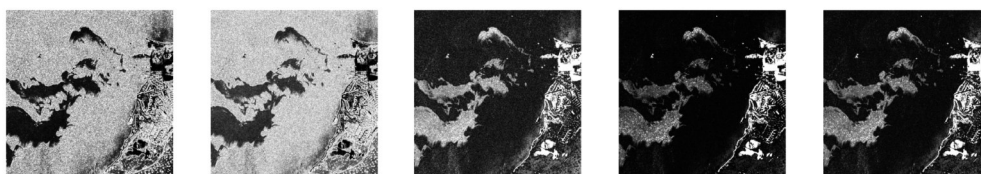


Figura 3.15: Contraste, Disimilitud, Homogeneidad, Energía y ASM para la Intensidad VV

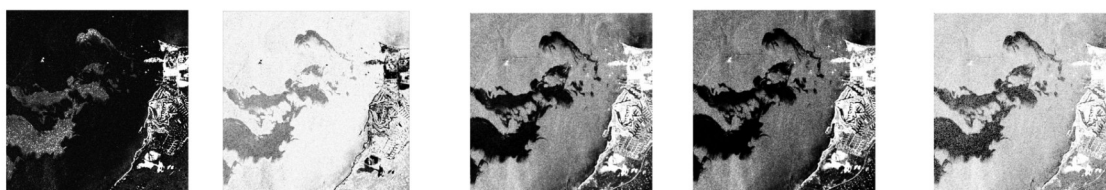


Figura 3.16: Máxima probabilidad, Entropía, Media GLCM, Varianza GLCM y Correlación GLCM para la Intensidad VV.

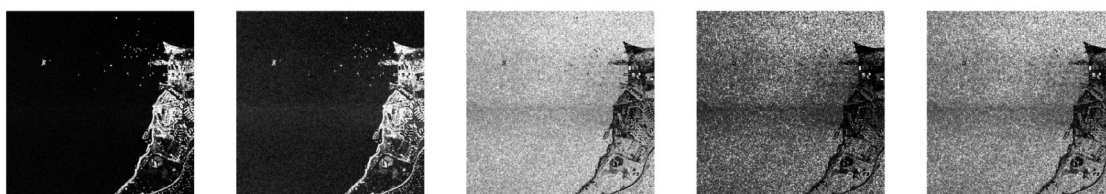


Figura 3.17: Contraste, Disimilitud, Homogeneidad, Energía y ASM para la Intensidad VH.

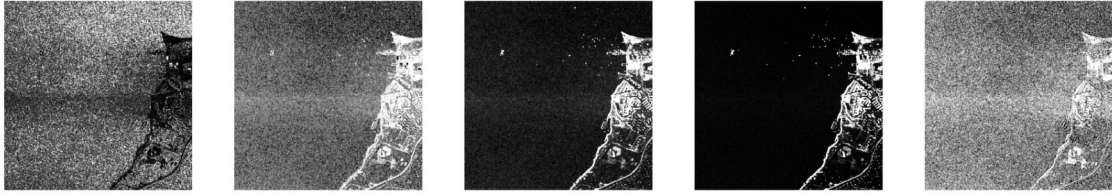


Figura 3.18: Máxima probabilidad, Entropía, Media GLCM, Varianza GLCM y Correlación GLCM para la Intensidad VH.

Ahora, se leen las 20 matrices de Orden 2577 en Python. En la Figura 1.12 se observa la lista de matrices.

Luego se redimensiona cada textura y en este caso se obtienen vectores de tamaño  $2577^2 \times 1 = 6640929 \times 1$ . Así, la matriz a la que se le aplicará el ACP, previamente normalizada, en este caso es de orden  $6640929 \times 20$ .

Al aplicar el ACP, se obtiene que con las ocho primeras componentes principales se explica el 99% de la varianza.

	Varianza explicada
1	0.468608
2	0.763221
3	0.87621
4	0.925997
5	0.955964
6	0.976841
7	0.985315
8	0.990871

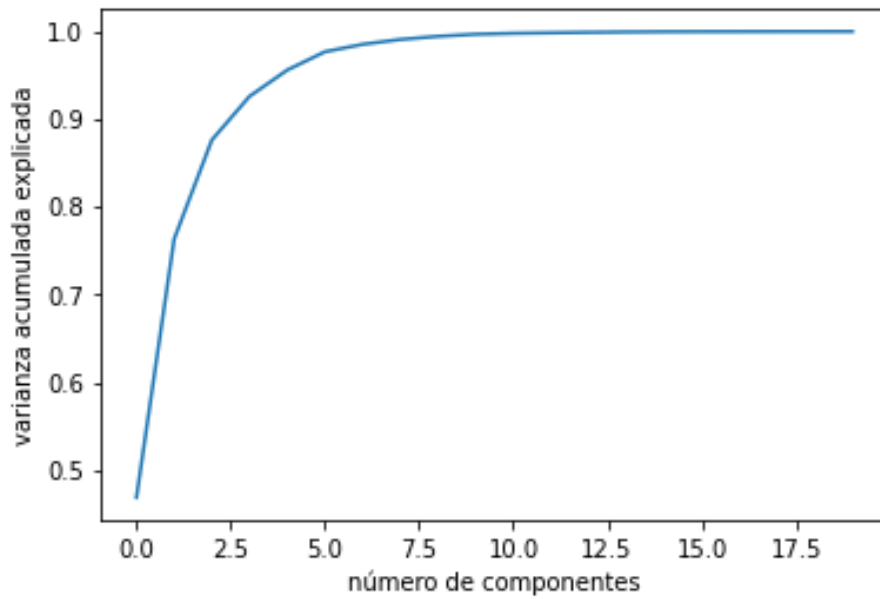


Figura 3.19: Análisis de ACP: varianza explicada acumulada

Finalmente, para determinar el  $k$  óptimo para implementar el algoritmo  $k$ -means se analiza para  $k \in \{2, \dots, 9\}$  y se obtiene que el valor que maximiza el score de silueta es  $k = 3$  como se observa en la Figura 3.20.

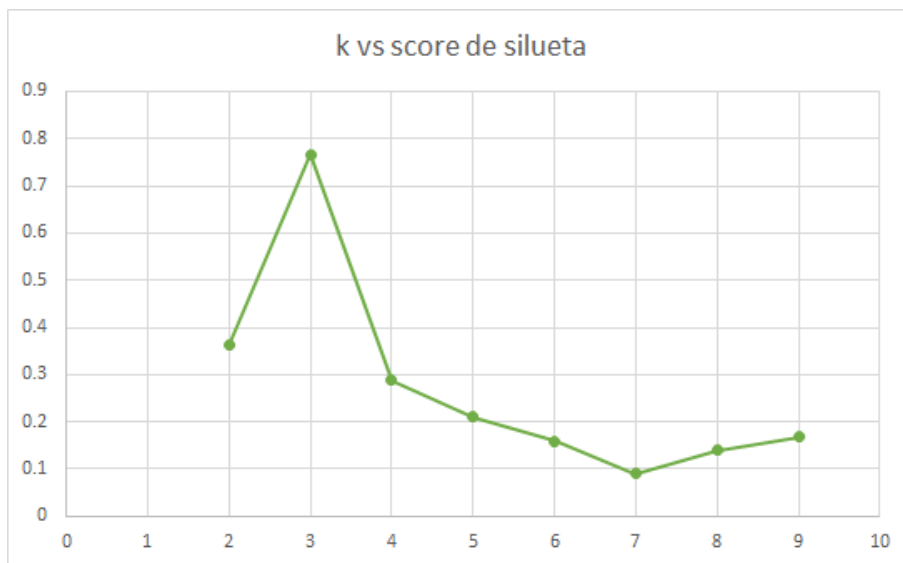


Figura 3.20: Simulación del score de silueta para algunos valores de  $k$

Con este procedimiento se etiqueta cada píxel de la imagen original que se muestra en la Figura 3.21. La zona amarilla, en su gran mayoría, agrupa a píxeles cuya intensidad es similar a la de un derrame de petró-

leo. La zona morada agrupa a píxeles que corresponden al mar y la zona celeste agrupa a píxeles de la superficie costera.

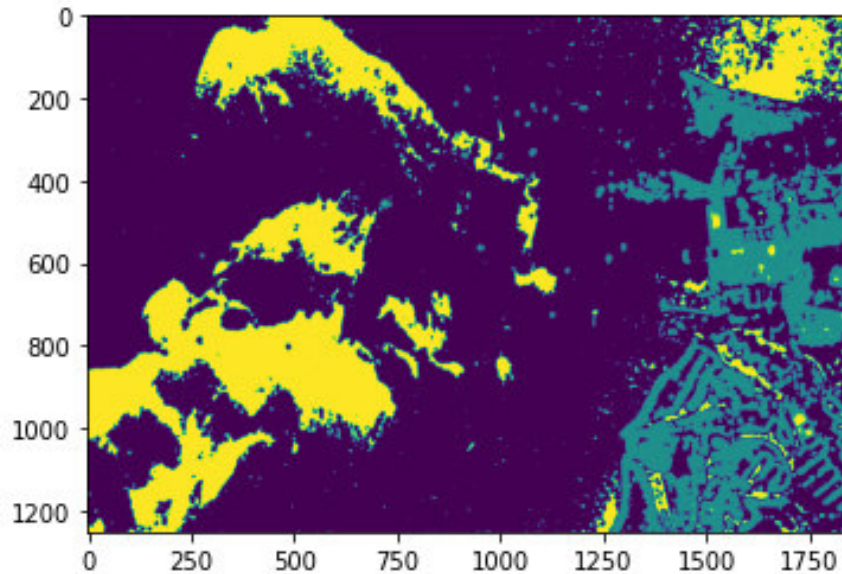


Figura 3.21: Clústers encontrados en la imagen

## 3.2. Conclusiones y recomendaciones

### 3.2.1. Conclusiones

Se logró consolidar una base de registros de incidentes asociados a hidrocarburos, entre ellos derrames de petróleo o sus derivados en la que se encuentra la fecha en que ocurrió el suceso, las coordenadas geográficas (longitud y latitud), el tiempo de derrame de petróleo que corresponde a las 5 categorías (“heavy oil”, “fuel oil”, “crude oil”, “oil” y “light crude oil” que engloban a las observaciones analizadas, es importante determinar el tipo de vertido que ocurrió pues esto ayuda a quienes deben tomar decisiones para tratar de remediar sus efectos perjudiciales. Luego del análisis de la base de datos `Raw incident data` que tiene 4189 registros y 17 variables, se obtuvo un conjunto de datos exclusivamente asociado a incidentes de hidrocarburos con un total de 896 registros y las variables más relevantes para su posterior análisis.

Las imágenes satelitales de sensor activo como las capturadas por el Sentinel-1 tienen ventajas meteorológicas y temporales con respecto a las imágenes ópticas ya que estas últimas se ven afectadas por condiciones atmosféricas y dependen de la luminosidad del sol. Los radares SAR se alimentan de la potencia que emiten y reciben para caracterizar la superficie terrestre. Entre las desventajas del SAR se encuentra la aparición del ruido aditivo (ruido térmico) y del ruido multiplicativo (ruido speckle) que en este método se eliminó mediante el filtrado de Lee con una ventana de  $10 \times 10$  debido a que si la ventana fuese más pequeña, se perdería información valiosa y en caso de ser más grande, puede traer problemas de cómputo. Es importante aclarar que se expresó la intensidad en escala logarítmica para evitar problemas con valores de intensidad demasiado grandes.

Una vez que se logra preprocesar la imagen satelital y con especial atención eliminar los ruidos, se procede a calcular la matriz de de nivel gris de co-ocurrencia que permite caracterizar a una imagen mediante el cálculo de las características de segundo orden que pertenecen a los grupos de contraste, de ordenación y de estadística. Estas características conocidas como texturas guardan información de relevancia para clasificar grupos de píxeles de la imagen. Debido a que se obtuvieron 20 matrices características de orden 2577, mediante el Análisis de Componentes Principales se redujo la matriz de orden  $6640939 \times 20$  tomando las 8 primeras componentes principales que explican el 99% de la varianza. Finalmente, se aplicó el algoritmo  $k$ -means para agrupar los grupos de píxeles que se encuentran a la menor distancia de los centros y mediante el score de siluetas se concluye que el  $k$  óptimo para el análisis es igual a 3.

Además de este documento, se adjunta un proyecto en R desarrollado para analizar la base de datos de registros, un cuaderno de Jupyter de Python: `Oil_Data_Map.ipynb` que exporta un html con los registros de derrames de hidrocarburos en un mapa mundial y el código `Sar_images.py` en donde se encuentra el script para el preprocesamiento de las imágenes satelitales.

### **3.2.2. Recomendaciones**

Las imágenes SAR demandan una ingente capacidad de memoria y computacional. Las siete imágenes que se mencionaron en este trabajo pesan 5.33 GB, por lo que se sugiere tener un computador con la memoria suficiente para almacenar y procesar las imágenes. Además, se recomienda contar con un disco sólido porque ayuda a que el procesamiento de las imágenes sea más rápido.

Finalmente, se recomienda utilizar los repositorios disponibles como Copernicus Open Acces Hub, NASA Earth Data, Geodata Mundi, entre otras de forma simultánea pues algunas de ellas guardan los productos Offline y puede tomar entre 2 a 48 horas en activarse para descargarlos. Cabe mencionar que es una gran oportunidad para continuar en el aprendizaje de la teledetección que las misiones de los satélites sean totalmente gratuitas y que se pueda utilizar plataformas de procesamiento de imágenes de forma libre. Con respecto a la extracción de las imágenes satelitales asociadas a derrames de hidrocarburos, en caso de que no exista una misión registrada en la fecha exacta de la base de registros, se recomienda buscar la de la fecha posterior más próxima que sirve para el análisis pues un derrame de petróleo se demora entre meses y años para desaparecer o atenuar sus efectos.

# Capítulo A

---

## Anexo

---

### A.1. Análisis de la base Raw Incident Data

#### A.1.1. Lectura de la base de datos

##### 1. Importar las librerías

```
1 library(readxl)
2 library(readr)
3 library(naniar)
4 library(VIM)
5 library(FSA)
6 library(data.table)
7 library(dplyr)
8 library(comprehenr)
9 library(kableExtra)
10 library(ggplot2)
11 library(TSstudio)
```

##### 2. Fijar directorios del proyecto



```

1 # Direcciones
2 # Consigo direccion actual del proyecto
3 dir.p <- getwd()
4 # Directorio de la carpeta Base
5 dir.d <- paste(dir.p, "Base/Oil_spills", sep = "/")
6 # Directorio de la carpeta Resultados
7 dir.r <- paste(dir.p, "resultados", sep = "/")

```

**3. Se realiza la lectura de datos mediante el comando `read.csv`. La base de datos se llamará `'raw_data'`.**

```

1 setwd(dir.d)
2 raw_data <- setDT(read.csv(file = "incidents.csv", header = TRUE, sep =
  ", ", dec = "."))
3 dim(raw_data)
4 output [1] 4189 17

```

Esta base de datos cuenta con 17 variables y 4189 observaciones. Se define la función VP para calcular el número total de datos faltantes y tomando en cuenta que existen espacios en blanco en la base de datos, se llenan estos espacios vacíos con `'NA'`.

```

1 #Calcula el número total de valores perdidos en una columna
2 VP <- function(x) {sum(is.na(x))}
3 raw_data[raw_data == " "] <- NA
4 raw_data_aux = raw_data

```

## A.1.2. Tratamiento de los datos de la base

### Datos faltantes

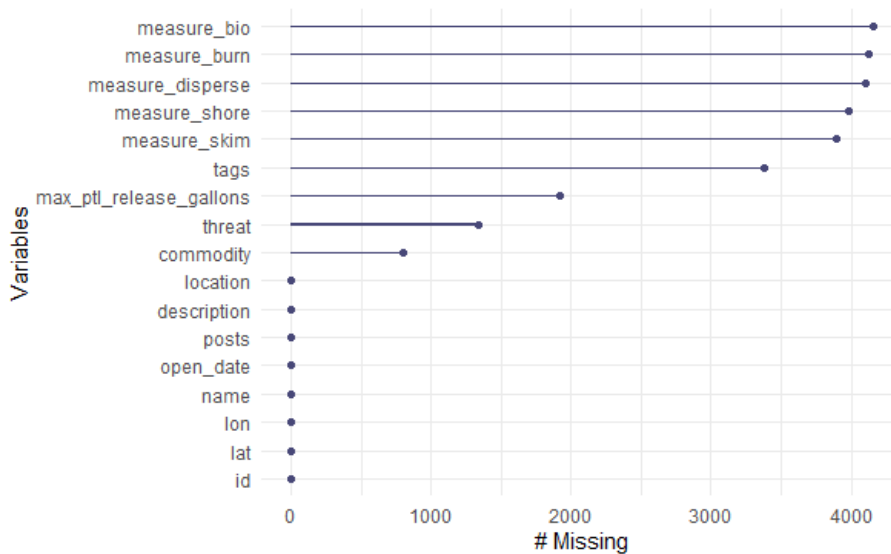


Figura A.1: Datos faltantes en las base de datos Raw Incident Data

Mediante el comando `gg_miss_var` se observan los datos faltantes de la base de datos. Se observa que la cantidad de NA en las variables de monitoreo es alta y como el objetivo del presente proyecto no es el tratamiento de los derrames sino la detección, entonces se retiran de la base de datos.

Luego, se observan datos faltantes también en las variables ‘commodity’ y ‘threat’ que sí son importantes para el estudio. Se considera el conjunto de palabras :

$$M = \{\text{oil, crude, diesel, diesel fuel, gasoline, crude oil}\}$$

que permitirán determinar la cantidad de veces que un elemento de  $M$  se repite en los párrafos de la variable ‘description’ que se define como el conjunto  $P$  que contiene las palabras del párrafo de descripción.

1. Se transforman a minúsculas las letras de las palabras de las variables que corresponden a la descripción y al tipo de derrame porque R discrimina entre mayúsculas y minúsculas.

```

1 v_minus <- c("description", "commodity")
2
3 minusculas <- function(DATA, VECTOR){
4   for (i in 1:length(VECTOR)){
5     DATA[[VECTOR[i]]] <- tolower(DATA[[VECTOR[i]]])
6   }
7   return(DATA)
8 }
9
10 raw_data <- minusculas(raw_data, v_minus)

```

2. Se define la función `contar_letras` que permite determinar la cantidad de veces que las palabras en  $M$  se repiten en el párrafo de descripción. Para ello se calcula la cardinalidad del conjunto para cada  $i \in \{1, \dots, |M|\}$ ,

$$C = \{m_i \in M : m_i \subset P\}.$$

Este conjunto está bien definido pues en caso de que no existan palabras de  $M$  en  $P$ ,  $C = \emptyset$  y por tanto  $|C| = 0$ .

```

1 contar_letras <- function(str1, pat){
2   lengths(regmatches(str1, gregexpr(paste0("\\b", pat), str1, ignore.
3     case = TRUE)))
4 }
5 count_oil <- contar_letras(raw_data$description, "oil")
6 count_crude <- contar_letras(raw_data$description, "crude")
7 count_diesel <- contar_letras(raw_data$description, "diesel")
8 count_diesel_fuel <- contar_letras(raw_data$description, "diesel fuel")
9 count_gasoline <- contar_letras(raw_data$description, "gasoline")
10 count_crude_oil <- contar_letras(raw_data$description, "crude oil")
11 count_oil_threat_original <- contar_letras(raw_data$threat, "Oil")

```

3. Mediante el paquete `comprehenr` creamos las variables indicadoras que asignen el valor de 0 en caso de que no se haya detectado la palabra al menos una vez en el párrafo y 1 en caso contrario.

```

1 id_oil = to_vec(for(x in count_oil) if (x == 0 ) 0 else 1 )
2 id_crude = to_vec(for(x in count_crude) if (x == 0 ) 0 else 1 )
3 id_diesel = to_vec(for(x in count_diesel) if (x == 0 ) 0 else 1 )
4 id_crude_oil = to_vec(for(x in count_diesel_fuel) if (x == 0 ) 0 else
  1 )
5 id_gasoline = to_vec(for(x in count_gasoline) if (x == 0 ) 0 else 1 )
6 id_diesel_fuel = to_vec(for(x in count_crude_oil) if (x == 0 ) 0
  else 1 )

```

Se obtiene lo siguiente:

- La cantidad de incidentes con oil son: 1891.
- La cantidad de incidentes con crude son: 447.
- La cantidad de incidentes con diesel son: 1166.
- La cantidad de incidentes con crude oil son: 387.
- La cantidad de incidentes con gasoline son: 157.
- La cantidad de incidentes con diesel fuel son: 329.

Además, como la variable `threat` etiqueta con “Oil” a los incidentes asociados a un derrame de petróleo, aceite o sus derivados, también se crea una variable indicadora a partir de esta variable.

```

1 count_oil_threat_original <- contar_letras(raw_data$threat, "Oil")
2 id_oil_threat_original = to_vec(for(x in count_oil_threat_original)
  if (x == 0 ) 0 else 1 )

```

Y se obtiene que la cantidad de incidentes con “Oil” en `threat` son: 2328.

4. Se añaden las variables indicadoras a la base de datos.
5. Se crea la variable `oil_id_threat` que sumará las variables indicadoras. Luego, se crea la variable `threat_aux` que asignará 0 en caso de que la entrada de `oil_id_threat` sea 0 y “Oil” en caso contrario. Con esto se está determinando qué incidentes tienen un derrame de aceite involucrado. Luego, se incluye esta variable en la base de datos.

```

1 oil_id_threat = id_oil + id_crude + id_diesel + id_crude_oil +
  id_gasoline + id_diesel_fuel + id_oil_threat_original
2 threat_aux0 = to_vec(for(x in oil_id_threat) if (x == 0 ) 0 else 1 )
3 threat_aux = to_vec(for(x in oil_id_threat) if (x == 0 ) 0 else "Oil
  " )
4 cat("La cantidad de incidentes que involucran un derrame de aceite es",
  sum(threat_aux0))
5
6 raw_data <- data.table(raw_data, threat_aux)

```

6. Se crea la base `raw_oil_data` en la que se filtran los datos tales que la variable `threat_aux` contenga la palabra “Oil” y se cuenta cuántos registros corresponden a un derrame de petróleo o sus derivados.

```

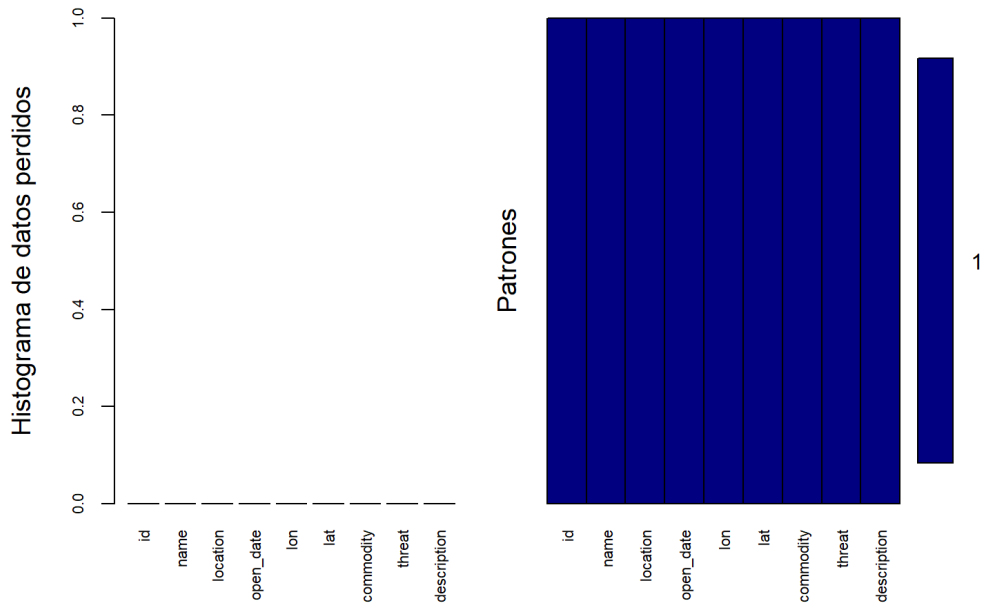
1 n_incidentes = dim(raw_data)[1]
2 raw_oil_data = raw_data[raw_data$threat_aux == "Oil" & raw_data$threat
  != "Other" & raw_data$threat != "Chemical"]
3 n_oil_incidentes = dim(raw_oil_data)[1]
4 cat("Se tienen", dim(raw_oil_data)[1], "derrames de petróleo o sus
  derivados, esto es el", round(n_oil_incidentes/n_incidentes*100,2),
  "% de la base de datos.")

```

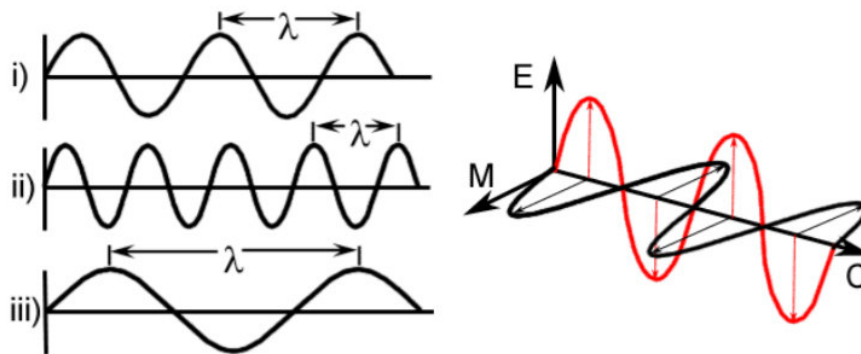
Se tienen 2328 derrames de petróleo o sus derivados, esto es el 55.57% de la base de datos original.

### A.1.3. Base final

En la variable `commodity` se obtienen 553 tipos de derrames, pero se determinaron 14 categorías que pueden englobar a cada uno de estos incidentes. Ahora, se puede observar en la Figura A.1.3 que ya no existen datos faltantes para `oil_data`.

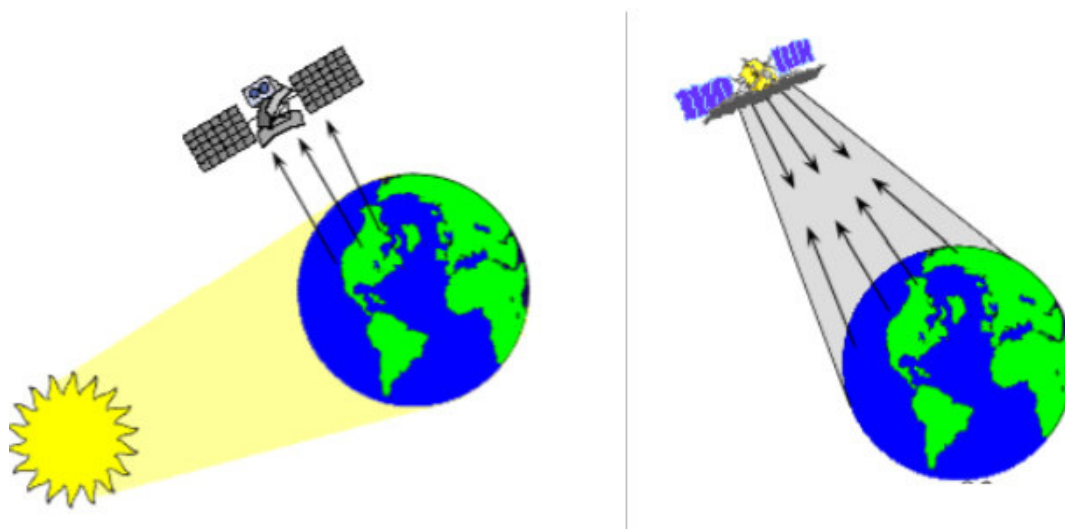


## A.2. Gráficos ilustrativos



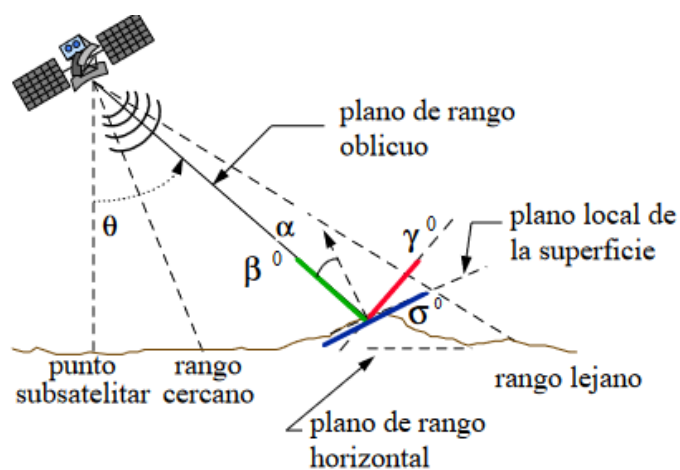
Fuente: Canada Centre for Remote Sensing

Figura A.2: Representación gráfica de la longitud de onda y de la frecuencia



Fuente: Canada Centre for Remote Sensing

Figura A.3: Representación gráfica pasivos vs. activo



Fuente: Laura Frulla, et al. [8].

Figura A.4: Magnitudes radiométricas medidas por el radar ( $\theta$  es el ángulo de observación del sensor)

<b>Banda</b>	$\lambda$	<b>Descripción</b>
Espectro visible	0.4-0.7 $\mu m$	La única que percibe el ojo humano. Distingue tres bandas elementales: azul (0.4-0.5 $\mu m$ ); verde (0.5-0.6 $\mu m$ ) y rojo (0.6-0.7 $\mu m$ )
Infrarrojo próximo	0.7-1.3 $\mu m$	Capaz de discriminar vegetación y/o concentraciones de humedad.
Infrarrojo medio	1.3-8 $\mu m$	Se presentan procesos de reflexión de la luz solar y de emisión de la superficie terrestre puede detectar focos de alta temperatura.
Infrarrojo térmico	8-14 $\mu m$	Presenta la emisión del espectro terrestre para detectar el calor de la mayor parte de los objetos de la superficie.
Micro-ondas	1 mm-1 m	Es interesante por su transparencia ante las características nubosas o efectos meteorológicos.

Cuadro A.1: Serie de bandas espectrales

<b>Banda</b>	$\lambda$	<b>f</b>
Banda-X	2.4-3.75 cm	12.5 a 8 GHz
Banca-C	3.75-7.5 cm	14 a 8 GHz
Banda-S	7.5-15 cm	4 a 2 GHz
Banda-L	15-30 cm	2 a 1 GHz
Banda-P	30-100 cm	1 a 0.3 GHz

Cuadro A.2: Clasificación de las bandas microondas

<b>Tipo</b>	<b>Descripción</b>
Textura gruesa	0.25 - 1 $mm^2$
Textura media	0.04 - 0.25 $mm^2$
Textura fina	< 0.04 $mm^2$

Cuadro A.3: Tipos de texturas de acuerdo a la superficie





### A.3. Diccionario de variables

Variable	Tipo	Descripción
id	int	Identificación del incidente
open_date	date	Fecha en que se notificó el derrame por la OR&R
name	str	Nombre del incidente
location	str	Localización del incidente
lat	float	Latitud. Norte es positivo y sur es negativo.
lon	float	Longitud. Norte es positivo y sur es negativo.
threat	str	Tipo de amenaza: "Oil", "Chemical" u "Other". "Oil" significa un derrame de petróleo. "Chemical" es un derrame de agentes químicos o biológicos y "Other" es algún otro tipo de incidente.
tags	str	Causas del incidente.
commodity	str	El item derramado. (Texto libre.)
is_measure_skim	boolean	Verdadero si se recogió en el agua (skimming) como contramedida.
is_measure_shore	boolean	Verdadero si se utilizó la limpieza de la línea costera como contramedida.
is_measure_bio	boolean	Verdadero si se utilizó la Biorremediación como contramedida.
is_measure_disperse	boolean	Verdadero si se utilizaron dispersantes como contramedida.
is_measure_burn	boolean	Verdadero si se utilizó la quema in situ como contramedida.
max_ptl_release_gallons	float	Máxima emisión potencial en galones. Puede ser una cantidad conocida o el límite superior de una estimación mínimo-máximo.
posts	integer	Número de publicaciones en IncidentNews.
description	str	Descripción del incidente.

Cuadro A.4: Diccionario de las variables de la base de datos "Raw Incident Data"

Fuente: IncidentNews



## A.4. Base oil\_data

id	open_date	lon	lat	commodity	threat	description	max_ptl_release_gallons
10407	1/20/2022	-74.0644	40.35463	fuel oil	Oil	at 9:30 pm on 20...	100
10394	12/15/2021	-118.159	33.75542	fuel oil	Oil	on 15-dec-2021,...	100
10395	12/15/2021	-118.233	33.7505	fuel oil	Oil	on 15-dec-2021,...	500
10377	11/1/2021	-153.285	56.99582	fuel oil	Oil	on 1-nov-2021, ...	4000
10375	10/26/2021	-165.972	54.675	otros	Oil	on 26-oct-2021,...	300
10371	10/21/2021	-76.0209	35.07297	fuel oil	Oil	at 3:45 pm on 21...	2000
10369	10/15/2021	-124.775	48.4092	fuel oil	Oil	on october 15,...	400
10361	10/2/2021	-118.087	33.63806	crude oil	Oil	at 0840 hrs on...	140028
10357	9/21/2021	-122.335	47.54641	otros	Oil	on the afternoon,...	2
10347	8/27/2021	-166.127	64.57194	suspected, posible or unkown	Oil	on august 27, 2021....	55
10345	8/26/2021	-133.39	57.42183	fuel oil	Oil	on august 26, 2021,...	80
10343	8/25/2021	-70.2275	43.64347	oil	Oil	on august 25, 2021...	500
10339	8/18/2021	-74.2003	40.62021	heavy oil	Oil	on 18 august 2021...	200
10336	8/15/2021	-93.8286	29.75417	condensate	Oil	on august 14, 2021...	13944

Cuadro A.5: Registros de la base de datos creada las variables name y location están ocultas para que se pueda visualizar correctamente en la hoja.



```

Usage:
gpt Maximum-Likelihood-Classifier [options]

Description:
Maximum Likelihood classifier

Parameter Options:
-PclassLevels=<int> ...
-PclassValStepSize=<double> ...
-PdoClassValQuantization=<boolean> ...
-PdoLoadClassifier=<boolean> ...
-PevaluateClassifier=<boolean> ...
-PevaluateFeaturePowerSet=<boolean> ...
-PfeatureBands=<string, string, string, ...> ...
-PlabelSource=<string> ...
-PmaxPowerSetSize=<integer> ...
-PminClassValue=<double> ...
-PminPowerSetSize=<integer> ...
-PnumTrainSamples=<int> ...
-PsavedClassifierName=<string> ...
-PtrainingBands=<string, string, string, ...> ...
-PtrainingVectors=<string, string, string, ...> ...
-PtrainOnRaster=<boolean> ...

```

Figura A.7: Output de gpt -h

## A.6. Códigos

```
1
2
3 # Asignamos el camino para extraer la metadata
4 import os
5 cwd = os.getcwd()
6 path = "\\Base/"
7
8 # Expresion regular para importar los archivos que empiecen con S1 y
   terminen con .zip
9 input_S1_docs = sorted(list(iglob(join(cwd+path, '**', '*S1*.zip'),
   recursive = True)))
10
11 nombre, modo_sensor, tipo_producto, polarizacion, h, w, nombre_bandas =
   ([] for file in range(7))
12
13 for file in input_S1_docs:
14     modo_sensor.append(file.split("_")[3])
15     tipo_producto.append(file.split("_")[4])
16     polarizacion.append(file.split("_")[-6])
17     # Lectura con snappy
18     s1_lectura = snappy.ProductIO.readProduct(file)
19     nombre.append(s1_lectura.getName())
20     h.append(s1_lectura.getSceneRasterHeight())
21     w.append(s1_lectura.getSceneRasterWidth())
22     nombre_bandas.append(s1_lectura.getBandNames())
23
24 # Definicion de un dataframe para trabajar con la metadata
25 df_s1_lectura = pd.DataFrame({'Nombre': nombre, 'Modo sensor':
   modo_sensor, 'Tipo de producto': tipo_producto, 'Polarizacion':
   polarizacion, 'Height': h, 'Width': w, 'Bandas': nombre_bandas})
26 display(df_s1_lectura)
```

Figura A.8: Lectura de las imágenes SAR

```

1 i = 0 # Visualizar brevemente la imagen tatelital
2 img = mpimg.imread(cwd + '\\Base\\Quick_visualization\\' + nombre[i] +
   '.png')
3 plt.figure(figsize = (15,15))
4 plt.title('Visualizacion - ' + nombre[i] + '\n')
5 #plt.axis('off')
6 plt.imshow(img)

```

Figura A.9: Código para visualización rápida

```

1 # Esquina superior
2
3 x, y, w, h = 175, 300, 10000, 18000
4
5 # Operador subset en snappy
6
7 # Se abre un objeto de snappy
8 parameters = snappy.HashMap()
9 # Parametros que se pueden visualizar en gpt -h
10 parameters.put('copyMetadata', True)
11 parameters.put('region', '%s,%s,%s,%s' % (x,y,w,h))
12 # Parametros de Snappy estan disponibles con la funcion createProduct
13 # Como se quiere realizar un subset, se coloca como primera entrada
14 # Se coloca los parametros a analizar en este caso de subset
15 # Finalmente, el producto
16 subset = snappy.GPF.createProduct('Subset', parameters, s1_lectura)
17 list(subset.getBandNames())
18
19 # Plot subset
20 output_bands = ['Amplitude_VV', 'Amplitude_VH']
21 vista_output(subset, output_bands, 41, 286, 20, 160)

```

Figura A.10: Código para subset de la imagen SAR



```

1 proj = '''PROJCS["UTM Zone 31 / World Geodetic System 1984",
2   GEOGCS["World Geodetic System 1984",
3     DATUM["World Geodetic System 1984",
4       SPHEROID["WG5 84", 6378137.0, 298.257223563, AUTHORITY["
5   EPSG", "7030"]],
6     AUTHORITY["EPSG", "6326"]],
7     PRIMEM["Greenwich", 0.0, AUTHORITY["EPSG", "8901"]],
8     UNIT["degree", 0.017453292519943295],
9     AXIS["Geodetic longitude", EAST],
10    AXIS["Geodetic Latitude", NORTH]],
11  PROJECTION["Transverse_Mercator"],
12  PARAMETER["central_meridian", 3.0],
13  PARAMETER["latitude_of_origin", 0.0],
14  PARAMETER["scale_factor", 0.9996],
15  PARAMETER["false_easting", 500000.0],
16  PARAMETER["false_northing", 0.0],
17  UNIT["m", 1.0],
18  AXIS["Easting", EAST],
19  AXIS["Northing", NORTH]]'''
20 # Operador de Correccion del terreno - snappy
21
22 parameters = snappy.HashMap()
23 parameters.put('demName', 'SRTM 3Sec')
24 parameters.put('imgResamplingMethod', 'NEAREST_NEIGHBOUR')
25 parameters.put('pixelSpacingInMeter', 10.0)
26 parameters.put('mapProjection', proj)
27 parameters.put('nodataValueAtSea', False) # No marca areas con
28   elevaciones
29 parameters.put('saveSelectedSourceBand', True)
30 terrain_correction = snappy.GPF.createProduct('Terrain-Correction',
31   parameters, speckle)

```

Figura A.11: Código para la corrección del terreno

```

1 # Operador Calibracion - snappy
2
3 def calibration(product):
4     parameters = snappy.HashMap()
5     parameters.put('selectedPolarisations', 'VH,VV')
6     parameters.put('outputBetaBand', True)
7     parameters.put('outputSigmaBand', False)
8     parameters.put('outputImageScaleInDb', False)
9     return snappy.GPF.createProduct('Calibration', parameters, product)
10
11 # Plot calibracion
12 output_bands = ['Sigma0_VV', 'Sigma0_VH']
13 vista_output(calibrated, output_bands, 0.00, 0.28, 0.00, 0.05)

```

Figura A.12: Código para aplicar la calibración radiométrica

```

1 # Aplicar el operador Thermal Noise - snappy
2 parameters = snappy.HashMap()
3 parameters.put('removeThermalNoise', True)
4 thermal_noise = snappy.GPF.createProduct('ThermalNoiseRemoval',
5     parameters, apply_orbit)
6
7 # Plot
8 output_bands = ['Intensity_VV', 'Intensity_VH']
9 vista_output(thermal_noise, output_bands, 0.02, 99836.52, 0.27,
10     18471.83)

```

Figura A.13: Código para remover el ruido térmico

## Oil Data ¶

Conjunto de datos que contiene registros de derrames de hidrocarburos.

```
In [12]: 1 # https://resourcewatch.org/data/explore/8746e75d-2749-405e-8f3b-0c12097860a1?section=Discover&selectedCollection=&zoom=3.88
2 raw_data = pd.read_excel('Base/Oil_spills/oil_data.xlsx', sheet_name='Data', engine='openpyxl')
3 print(raw_data.shape)
4 raw_data.head(50)
```

(2328, 10)

Out[12]:

	id	name	location	open_date	lon	lat	commodity	threat	description	max_ptl_release_gallons
0	6205	Witwater, Galeta Island, Canal Zone, Panama	Galeta Island, Canal Zone, Panama	1968-12-13	-80.666700	9.583330	fuel oil	Oil	on december 13, 1968, the oil tanker witwater ...	588000.0
1	6206	Santa Barbara Well blowout, Santa Barbara, Cal...	Santa Barbara, California	1969-01-28	-119.750000	34.166700	crude oil	Oil	on january 28, 1969, the union oil company wel...	4200000.0
2	6207	Hamilton Trader, Liverpool Bay, England	Liverpool Bay, England	1969-04-30	3.333330	53.500000	fuel oil	Oil	early on the morning of april 30, 1969, the ha...	168000.0
3	6208	Arrow, Nova Scotia, Canada	Nova Scotia, Canada	1970-02-04	-61.100000	45.466700	heavy oil	Oil	on february 4, 1970, at 0935, the steam tanker...	3470000.0
4	6209	Chevron Main Pass Block 41, 11 miles E of the ...	11 miles E of the Mississippi River delta, Lou...	1970-02-10	-88.983300	29.383300	crude oil	Oil	the chevron main pass block 41c platform caught...	NaN

Figura A.14: Visualización de los datos de Oil Data

```

1 import pandas as pd
2 import folium
3
4 world = folium.Map(
5     zoom_start=2,
6     location=[13.133932434766733, 16.103938729508073])
7 raw_data = pd.read_excel('Base/Oil_spills/oil_data.xlsx', sheet_name='
    Data', engine='openpyxl')
8 def select_incident(min_date, max_date, type_incident, raw_data):
9     incident_base = raw_data[(raw_data.threat == type_incident) & (
    raw_data.open_date >= min_date) & (raw_data.open_date <= max_date)]
10    return(incident_base)
11
12 min_date = "2014-01-01"
13 max_date = "2022-01-28"
14 type_incident = "Oil"
15 oil_spills = select_incident(min_date, max_date, type_incident,
    raw_data)
16 print("Existen", oil_spills.shape[0], "derrames de petróleo entre las
    fechas:")
17 print(min(oil_spills.open_date), "y", max(oil_spills.open_date), ".")
18
19 oil_spills = oil_spills[(oil_spills.commodity == 'crude oil') | (
    oil_spills.commodity == 'fuel oil') | (oil_spills.commodity == '
    heavy oil') | (oil_spills.commodity == 'light crude oil') | (
    oil_spills.commodity == 'oil')]
20 print("Existen", oil_spills.shape[0], "derrames de petróleo entre las
    fechas:")
21 print(min(oil_spills.open_date), "y", max(oil_spills.open_date), ".")

```

Figura A.15: Código para crear el mapa de registros de derrames de petróleos (parte 1)

```

1 def select_marker_color(row):
2     if row['commodity'] == 'crude oil':
3         return 'darkgreen'
4     elif row['commodity'] == 'fuel oil':
5         return 'orange'
6     elif row['commodity'] == 'heavy oil':
7         return 'red'
8     elif row['commodity'] == 'light crude oil':
9         return 'beige'
10    elif row['commodity'] == 'oil':
11        return 'blue'
12
13    oil_spills['colors'] = oil_spills.apply(select_marker_color, axis=1)
14    world_all_cities_colored = folium.Map(
15        zoom_start=2,
16        location=[13.133932434766733, 16.103938729508073]
17    )
18
19    for _, city in oil_spills.iterrows():
20        folium.Marker(
21            location=[city['lat'], city['lon']],
22            popup=[city['lat'], city['lon'], city['open_date']],
23            tooltip=[city['location'], city['name']],
24            icon=folium.Icon(color=city['colors'], prefix='fa', icon='
circle')
25        ).add_to(world_all_cities_colored)
26
27    world_all_cities_colored

```

Figura A.16: Código para crear el mapa de registros de derrames de petróleo (parte 2)

list\_bands - Lista (20 elementos)

Índice	Tipo	Tamaño	Valor
0	Array of float32	(2577, 2577)	[[0.5 0.34 0.2780612 ... 0.68622446 0.83 0.7361111 ...
1	Array of float32	(2577, 2577)	[[12.333333 10.7 9.357142 ... 6.428571 2.1 3. ...
2	Array of float32	(2577, 2577)	[[3.3333333 3.3 3.2142856 ... 2.142857 1.3 1.6666666]
3	Array of float32	(2577, 2577)	[[0.70710677 0.5830952 0.5273151 ... 0.82838666 0.91104335 0.8579691 ...
4	Array of float32	(2577, 2577)	[[3.1780539 3.9460027 4.220346 ... 2.6471772 2.0253263 2.2706373]
5	Array of float32	(2577, 2577)	[[0.17777778 0.30967742 0.33502537 ... 0.4 0.7407407 0.7567568 ...
6	Array of float32	(2577, 2577)	[[2.1666667 2.55 2.5357144 ... 1.7857143 1.65 2.1666667]
7	Array of float32	(2577, 2577)	[[ 7.5 7.75 7.035714 ... 5.357143 4.05 6.166666 ...
8	Array of float32	(2577, 2577)	[[1.0559578 0.9835747 0.93112475 ... 1.2912087 1.43 1.3 ...
9	Array of float32	(2577, 2577)	[[0.5 0.4 0.2857143 ... 0.5714286 0.5 0.5 ...
10	Array of float32	(2577, 2577)	[[0.2361111 0.125 0.08418367 ... 0.09183674 0.13 0.1666666 ...
11	Array of float32	(2577, 2577)	[[298. 257.6 224. ... 117.42857 72.9 93.33333 ...
12	Array of float32	(2577, 2577)	[[20.666666 19.2 17.714285 ... 12.285714 9.3 10.666667 ...
13	Array of float32	(2577, 2577)	[[0.48591265 0.35355338 0.29014423 ... 0.30304575 0.36055514 0.4082483 ...
14	Array of float32	(2577, 2577)	[[4.3921905 5.644891 6.4168563 ... 6.268325 5.5755763 4.9698133]
15	Array of float32	(2577, 2577)	[[0.67934 0.741832 0.7723577 ... 0.9022825 0.9377508 0.9286079 ...
16	Array of float32	(2577, 2577)	[[27. 28.4 28.5 ... 32.642857 32.65 34.166668]
17	Array of float32	(2577, 2577)	[[ 464.66666 498.9 492. ... 600.8571 585.55 653.6667 ...
18	Array of float32	(2577, 2577)	[[0.20841175 0.17274667 0.18233137 ... 0.350564 0.47262177 0.3367881 ...
19	Array of float32	(2577, 2577)	[[0.16666667 0.1 0.07142857 ... 0.07142857 0.1 0.0833333 ...

Figura A.17: Lista de las matrices de textura

---

## Referencias bibliográficas

---

- [1] Maher Arebey, M. A. Hannan, Rawshan Begum, and Hassan Basri. Solid waste bin level detection using gray level co-occurrence matrix feature extraction approach. *Journal of environmental management*, 104:9–18, 04 2012.
- [2] Stefania Barburiceanu, Romulus Terebes, and Serban Meza. Texture feature extraction and classification using glcm and lbp-based descriptors. *Applied Sciences*, 11(5), 2021.
- [3] Jason Bergner. Anaconda install and uninstall (powershell). <https://silentinstallhq.com/anaconda-install-and-uninstall-powershell/>, 2022. Recuperado el: 2022-01-30.
- [4] Emilio Chuvieco. *Fundamentos de Teledetección espacial*. RIALP, 3 edition, 1996.
- [5] Alaa Eleyan and Hasan Demirel. Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turkish Journal of Electrical Engineering and Computer Sciences*, 19:97–107, 01 2011.
- [6] Peter Flach. *Machine learning: the art and science of algorithms that make sense of data*. Cambridge university press, 2012.
- [7] Canada Centre for Remote Sensing. *Fundamentals of Remote Sensing*. RIALP, 3 edition, 1996.
- [8] Laura Frulla, Jorge Milovich, Haydee Karszenbaum, Patricia Kandus, and Julián Alvarez. Metodologías de pre-procesamiento y pro-

cesamiento utilizadas en el tratamiento cuantitativo de datos sar para el estudio de ambientes en el bajo delta del río paraná, argentina. 02 2022.

- [9] Mryka Hall-Beyer. *GLCM Texture: A Tutorial v. 3.0* March 2017. 03 2017.
- [10] Anurag Kulshrestha. *Dark spot detection for characterization of oil spills using polsar remote sensing*. Faculty of Geo-Information Science and Earth Observation of the University of Twente, 2018.
- [11] Henri Maitre. *Processing of Synthetic Aperture Radar*. SAR Images Digital Signal & Image Processing Series Iste-DSP. Wiley-ISTE, 1 edition, 2008.
- [12] Henri Maitre. *Informe final. Determinación de los impactos de los recursos hidrobiológicos y en los ecosistemas marinos presentes en el área de influencia del derrame de hidrocarburo de Bahía Quintero, V Región*. Instituto de Fomento Pesquero, 2016.
- [13] Carl Brown Merv Finga. *Review of oil spill remote sensing*. *Marine Pollution Bulletin*. Faculty of Geo-Information Science and Earth Observation of the University of Twente, 2018.
- [14] Uferah Shafi y Syed Mohammad Hassan Zaidi Naveed Iqbal, Rafia Mumtaz. Gray level co-occurrence matrix (glcm) texture based crop classification using low altitude remote sensing platforms. *PeerJ Computer Science*, 7, 2021.
- [15] Michael O'Byrne, Franck Schoefs, Bidisha Ghosh, and Vikram Parkrashi. Texture analysis based detection and classification of surface features on ageing infrastructure elements. 09 2012.
- [16] Laura Plitt. ¿cuál es el verdadero impacto de un derrame de petróleo? [https://www.bbc.com/mundo/internacional/2010/04/100428\\_derrame\\_petroleo\\_claves\\_lp](https://www.bbc.com/mundo/internacional/2010/04/100428_derrame_petroleo_claves_lp), 2010. Recuperado el: 2022-01-30.
- [17] Its' Hak Dinstein Robert M Haralick, Karthikeyan Shanmugam. *Textural Features for Image Classification*. 6. IEEE Transactions on systems, man, and cybernetics, 1973.



- [18] Aditya Sharma. Snap-python (snappy) installation on windows. <https://www.youtube.com/watch?v=14YM1kKdgA8>, 2020. Recuperado el: 2022-01-30.
- [19] Serco Italia SPA. Oil spill mapping with sentinel-1 (version 1.2). *RUS Lectures*, 2019.
- [20] Ignacio Vergara and Francisco Pizzaro. *Control de derrames de petróleo*. Apuntes para Cursos en América Latina de IMCO - CPPS - PNUMA, 1 edition, s.a.
- [21] Ardhi Wicaksono, Dwi Pebrianti, Luhur Bayuaji, and Jasni Mohamad Zain. Performance of various speckle reduction filters on synthetic aperture radar image. 08 2015.
- [22] Jin Xu, Xinxiang Pan, Baozhu Jia, Xuerui Wu, Peng Liu, and Bo Li. Oil spill detection using lbp feature and k-means clustering in shipborne radar image. *Journal of Marine Science and Engineering*, 9(1), 2021.
- [23] Angie Paola Hurtado Acosta y Paola Andrea Suarez Jaimes. *Estimación de la subsidencia de la ciudad de Bogotá mediante imágenes de radar y técnicas de interferometría diferencial DINSAR*. Universidad Dsitrital Francisco José de Caldas, 2008.
- [24] Junling Zhu, Jianguo Wen, and Yafeng Zhang. A new algorithm for sar image despeckling using an enhanced lee filter and median filter. volume 1, pages 224–228, 12 2013.