



**Universidad de Valladolid**

Facultad de Ciencias

*Trabajo Fin de Grado*

Grado en física

**Algoritmo de detección de nubes a partir de redes neuronales  
perceptrón multicapa en imágenes satelitales**

*Autor: Francisco Javier Montaña García*

*Tutor/es: Isidro Alberto Pérez Bartolomé*

---

El que suscribe, D. **Francisco Javier Montaña García**, alumno del Grado en física en la Universidad de Valladolid, asume la responsabilidad sobre la veracidad y autoría de los datos e informaciones recogidos en el presente trabajo final de grado. Así mismo, cualquier referencia académica como fuente de información queda citada y recogida en la bibliografía correspondiente. Al mismo tiempo, declara y manifiesta que es consciente de las consecuencias académicas que pudieran derivarse de la falsificación de cualesquiera de los datos y/o información anteriormente referidos.

**Fdo: Francisco Javier Montaña García**

Valladolid ,2020

---

**D. Isidro Alberto Pérez Bartolomé**, profesor titular del área de física aplicada de la Universidad de Valladolid

Informa:

Que el presente trabajo, titulado, **Algoritmo de detección de nubes a partir de redes neuronales perceptrón multicapa en imágenes satelitales**, ha sido realizado bajo su supervisión por **Francisco Javier Montaña García**, y autorizo la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expide y firma el presente informe en Valladolid en el año 2020

**El director**

Isidro Alberto Pérez Bartolomé

# Agradecimientos

El presente trabajo me gustaría dedicárselo a todas las personas que me han aportado algo en mi formación, tanto a nivel personal como profesional.

A mi familia, que sin ellos no habría sido posible llegar hasta aquí, y a todas las personas que han estado conmigo pero que, desgraciadamente hoy no están.

A mis profesores de ciencias de educación secundaria, gracias a ellos aprendí que, aunque la física no puede explicar todo, al menos te puede ayudar a entender una pequeña parte del maravilloso diseño que conforma el universo.

A mi tutor Isidro, por su ayuda y disposición a la hora de tutelar este trabajo.

**Ni el más elegante de los diseños sería capaz de paliar el daño causado por un sistema de medida poco fiable....**

**JL Fleis**

## Índice de Contenidos

<b>Resumen</b> .....	4
<b>Abstract</b> .....	4
<b>Palabras clave</b> .....	4
<b>I. Fundamento General</b> .....	5
<b>Introducción</b> .....	6
<b>Planteamiento del problema</b> .....	7
<b>1.1 Métodos de clasificación de coberturas.</b> .....	8
<b>1.2 Plataformas, satélites, sensores.</b> .....	8
<b>1.3 Análisis estadístico de resultados.</b> .....	10
<b>1.4 Software empleado.</b> .....	12
<b>II. Algoritmos basados en análisis de píxeles</b> .....	13
<b>Resumen</b> .....	14
<b>2.1 Introducción</b> .....	14
<b>2.2 Zonas de estudio</b> .....	15
<b>2.3 Metodología.</b> .....	17
<b>2.3.1 Tratamiento preFmask.</b> .....	17
<b>2.4 Flujo de trabajo</b> .....	18
<b>2.4.1 Tratamiento preFmask.</b> .....	18
<b>2.5 Resultados</b> .....	20
<b>2.5.1 Tratamiento preFmask</b> .....	20
<b>2.5.2 Fmask.</b> .....	21
<b>2.6 Análisis de resultados</b> .....	22
<b>2.7 Conclusiones capitulares.</b> .....	23
<b>III. Algoritmos basados en el análisis de objeto</b> .....	24
<b>Resumen</b> .....	25
<b>3.1 Introducción</b> .....	25
<b>3.2 Zonas de estudio</b> .....	26
<b>3.3 Metodología.</b> .....	26
<b>3.3.1 Conceptos previos sobre redes neuronales.</b> .....	26

<b>3.3.2</b>	<b>Estructura de las redes neuronales artificiales.....</b>	<b>27</b>
<b>3.3.3</b>	<b>Parámetros en las redes neuronales artificiales. ....</b>	<b>29</b>
<b>3.3.4</b>	<b>Generación de archivos vectoriales de entrenamiento y validación.....</b>	<b>30</b>
<b>3.4</b>	<b>Algoritmo de enmascaramiento de nubes. ....</b>	<b>35</b>
<b>3.5</b>	<b>Flujo de trabajo.....</b>	<b>37</b>
<b>3.6</b>	<b>Análisis de resultados.....</b>	<b>38</b>
<b>3.7</b>	<b>Conclusiones capitulares.....</b>	<b>40</b>
<b>IV.</b>	<b>Conclusiones.....</b>	<b>41</b>
<b>V.</b>	<b>Referencias.....</b>	<b>43</b>
<b>VI.</b>	<b>Anexos.....</b>	<b>46</b>
	<b>Anexo I. Glosario de términos.....</b>	<b>47</b>
	<b>Anexo II. Proceso Fmask.....</b>	<b>48</b>
	<b>Anexo III. Archivo xml del modelo de entrenamiento de red neuronal.....</b>	<b>51</b>
	<b>Anexo IV. Código Orfeo.....</b>	<b>53</b>
	<b>Anexo V. Relación de imágenes utilizadas.....</b>	<b>55</b>

## Índice de Figuras

Figura 1. Bandas espectrales Sentinel-2 a 10 metros (Copernicus 2020) .....	9
Figura 2. Bandas espectrales Sentinel- 2 a 20 metros (Copernicus 2020) .....	9
Figura 3. Bandas espectrales WorlView-2 y Quickbird (Digital Globe 2020).....	10
Figura 4. Imagen de Sentinel- 2 correspondiente al noroeste de España (Copernicus., 2020) ....	15
Figura 5. Imagen de Sentinel-2 correspondiente al norte de España (Copernicus., 2020) .....	16
Figura 6. Imagen de Sentinel-2 correspondiente al mar Mediterráneo en las cercanías de la costa italiana (Copernicus., 2020).....	16
Figura 7. Diagrama de trabajo.....	18
Figura 8. Imagen comparativa Banda C Sentinel-2.....	20
Figura 9. Imagen Correspondiente a la banda K originada para Sentinel-2 .....	20
Figura 10. Banda S Sentinel -2.....	21
Figura 11. Comparativa de imagen 4 con su respectiva enmascarada. A la izquierda la imagen original, a la derecha la imagen obtenida mediante el algoritmo final Fmask. ....	21
Figura 12. Comparativa de imagen 5 con su respectiva enmascarada. A la izquierda la imagen original, a la derecha la imagen obtenida mediante el algoritmo final Fmask. ....	22
Figura 13. Comparativa de imagen 6 con su respectiva enmascarada. A la izquierda la imagen original, a la derecha la imagen obtenida mediante el algoritmo final Fmask. ....	22
Figura 14. Estructura de una red neuronal artificial multicapa .....	28
Figura 15. Tiempo de convergencia de diferentes funciones de activación.....	29
Figura 16. Proceso de obtención de cromaticidades en zonas nubosas con luz y con sombra ...	32
Figura 17. Diagrama de trabajo del algoritmo.....	37
Figura 18. Imagen de Quickbird relativa a la zona norte de España. A la izquierda la imagen multiespectral, a la derecha la imagen con enmascaramiento nuboso (Digital Globe., 2020). ...	38
Figura 19.. Imagen de WorldView-2 relativa a la zona norte de España. A la izquierda la imagen multiespectral, a la derecha la imagen con enmascaramiento nuboso. (Digital Globe., 2020). ..	38
Figura 22 . Imagen de WorldView-2 relativa a la zona norte de España. A la izquierda la imagen multiespectral, a la derecha la imagen con enmascaramiento nuboso (Digital Globe., 2020). ...	39

## Índice de Tablas

Tabla 1.Indicadores estadísticos. ....	22
Tabla 2. Parámetros de red a utilizar .....	30
Tabla 3. Indicadores estadísticos .....	39



## **Resumen**

---

El enmascaramiento de nubes es de gran utilidad en la clasificación de diferentes coberturas en imágenes de satélite. Este procedimiento favorece la eliminación de ruido haciendo posible la mejora de clasificación del resto de coberturas, facilitando la medida y el análisis de diferentes parámetros estadísticos que aportan información sobre el terreno a estudiar. Hasta ahora, se han propuesto diferentes métodos para el enmascaramiento de las nubes. Sin embargo, presentan variaciones en cuanto a precisión, tiempo de procesamiento y su facilidad de implementación. Tradicionalmente se han empleado técnicas de enmascaramiento basadas en el píxel. Actualmente, nuevas técnicas basadas en inteligencia artificial, centradas en la clasificación a nivel de objeto, están produciendo resultados eficientes y postulándose como la alternativa a la metodología tradicional. En el presente trabajo se expondrá una comparativa de las diferentes técnicas de enmascaramiento, basadas en el píxel mediante el algoritmo Fmask así como basadas en el objeto. Para ello se implementará un algoritmo fundamentado en redes neuronales. Se analizarán las ventajas y defectos de cada técnica y se dará una breve reseña a lo que deparan los próximos años en el tema de enmascaramiento nuboso.

## **Abstract**

---

Cloud masking is very useful in classifying different coverage in satellite images. This procedure favors the elimination of noise, making it possible to improve the classification of the rest of the coverage, favoring the measurement and analysis of different statistical parameters that provide information on the field to be studied. Different methods for cloud masking have been proposed to date. However, they vary in terms of accuracy, processing time and ease of implementation. Traditionally pixel-based masking techniques have been used. Currently, new techniques based on artificial intelligence, focused on object level classification, are producing efficient results and running as the alternative to the traditional methodology. In the present work, a comparison of the different masking techniques, based on the pixel using the Fmask algorithm as well as based on the object, will be presented, for this an algorithm based on neural networks will be implemented. The advantages and defects of each technique will be analyzed, and a brief review will be given to what the coming years will hold in the cloud masking issue.

## **Palabras clave**

---

Enmascaramiento, nubes, sombras nubosas, Sentinel-2, Fmask, bandas, infrarrojo cercano, redes neuronales, perceptrón mutlicapa, GBDX, WorldView-2, Quickbird, capa oculta, validación, entrenamiento.

# **I. Fundamento General**

---

## **Introducción**

---

Un gran inconveniente en los estudios de superficie del suelo es la presencia de nubes en las imágenes de satélite provenientes de sensores ópticos (Alainet et al., 2006). Esto es debido principalmente al brillo que generan las nubes, de manera que los datos de reflectancia obtenidos en la superficie se ven alterados ante tal suceso. En otras ocasiones las nubes son tenues, y aunque es posible detectar la superficie terrestre, los datos presentan ruidos considerables. Por lo tanto, cuando se realizan estudios de la superficie terrestre mediante imágenes satelitales, es necesario emplear técnicas precisas de enmascaramiento de nubes para solventar tales problemas.

Hasta la fecha se han desarrollado diferentes métodos para el enmascaramiento nuboso, los cuales principalmente varían en complejidad y en las características espaciales, espectrales, temporales o radiométricas del tipo de sensor con el que se trabaje. Dentro de los métodos de discriminación de nubes se encuentran diferentes técnicas empleadas; basadas en el análisis a nivel de píxel y en el análisis de objeto (conjunto de píxeles). Dentro de los métodos de enmascaramiento basados en el píxel se encuentran los métodos radiométricos y geométricos, que emplean información de la geometría del sensor como su altura y ángulo de visión (Martinuzzi et al., 2007); métodos basados en umbrales, que excluyen ciertos intervalos en las bandas espectrales donde se supone que hay nubes (Kennedy et al., 2010; Sano et al., 2007); métodos basados en análisis de series de tiempo, cuyo fin es componer una imagen de referencia sin nubes de muchas épocas para comparar con las imágenes con presencia de estas (Goodwin et al., 2013; Lyapustin et al., 2008). Dentro de los métodos basados a nivel de objeto se encuentran técnicas de Machine Learning como pueden ser, redes neuronales (Alain et al., 2006; Bankert, 1994; Lissens et al., 2000), clasificaciones de máxima entropía, así como técnicas de Deep Learning fundamentadas en métodos de aprendizaje automático y recurrente mucho más avanzado que las técnicas de Machine Learning.

A lo largo del presente documento se presentará una comparativa entre un método de enmascaramiento basado en el píxel mediante un pretratamiento previo de imágenes, aplicando después el algoritmo Fmask, así como un método de discriminación nubosa basado en objeto mediante un algoritmo de Machine Learning fundamentado en redes neuronales.

Finalmente se da un análisis comparativo entre los dos métodos de enmascaramiento, tanto a nivel de precisión, rendimiento, operatividad.

El motivo de proponer este trabajo fin de grado corresponde principalmente a las dificultades que generan las nubes en los procesos de teledetección espacial. Con lo que se pretende dar criterios de decisión a los diversos usuarios de imágenes satelitales para identificar qué tipo de metodología se adapta mejor a sus necesidades.

Este documento contiene una descripción de la problemática, los objetivos que se pretenden alcanzar, un marco teórico, la metodología propuesta de los diferentes métodos, los resultados, así como las discusiones, análisis y conclusiones del proyecto.

## **Planteamiento del problema**

---

Obtener información espacializada de los fenómenos terrestres se ha convertido en uno de los objetivos más importantes para los investigadores, esto se ha hecho evidente en el desarrollo de los Sistemas de Información Geográfica, las tecnologías de navegación por satélite y la percepción remota. Provenientes de esta última, las imágenes satelitales se han convertido en una magnífica fuente de información ya sea para la planificación del territorio, el análisis del comportamiento de la superficie terrestre o, en una dimensión mucho mayor, el monitoreo de los cambios en la dinámica terrestre generados ya sea por actividades humanas o por los mismos procesos naturales del planeta.

Sin embargo, extraer información cartográfica categorizada acerca de lo que contiene una imagen de satélite puede llegar a ser un proceso largo y tedioso. Cuando se quiere obtener una determinada información, se necesita eliminar todo aquello que nos es innecesario. En la mayoría de las imágenes de satélite, la presencia de nubes supone una problemática en cuanto a la finalidad de estudio que se busca (Fisher et al., 2009; Sano et al., 2007). Más aún cuando se trata de determinados tipos de nubes, como pueden ser nubes altas tipo cirros, en las cuales se tiene una visión de la cobertura terrestre a través de ellas mismas, de manera que la decisión de otorgar a esa parcela nubosa la etiqueta de información válida o no muchas veces compromete los criterios de clasificación, exponiendo errores de omisión o comisión en función de la decisión tomada.

Las Redes neuronales son procedimientos no lineales fundamentados en la inteligencia artificial, utilizadas para la predicción de datos y clasificación de patrones en las imágenes, las cuales generalmente están conformadas por capas de entrada, capas ocultas de procesamiento y capas de salida que, en el caso particular de las imágenes de satélite, están determinadas por el número de bandas a interpretar y el número de coberturas a identificar o enmascarar.

Las redes son una de las tecnologías recientemente utilizadas para la interpretación y discriminación de coberturas, que tiene como ventaja permitir el “aprendizaje” del algoritmo, a partir de datos de entrenamiento que se le ingresan, los cuales por cada entrada en el cálculo y comparación con semillas de referencia, ajustan el resultado generando una respuesta con niveles de calidad muy precisa.

## 1.1 Métodos de clasificación de coberturas.

Existen diferentes métodos de clasificación de imágenes. Estos métodos se pueden categorizar como métodos de clasificación basados en píxeles, en los cuales se trabaja con los valores de cada pixel, dentro de estos métodos se encuentran los supervisados y los no supervisados y técnicas de clasificación basada en objetos.

Dentro de los métodos de clasificación basada en el pixel se encuentran:

- **Clasificación Supervisada:** es aquella donde el analista tiene una participación activa en el proceso, ya que es la persona que lo realiza y quien indica cómo se hace la clasificación de cada uno de los píxeles de la imagen, mediante la definición de áreas de entrenamiento que se conocen por la naturaleza de la superficie. Los pasos necesarios para llevar a cabo este tipo de clasificación son: el conocimiento del área de estudio para la selección de sitios de entrenamiento de acuerdo con las coberturas presentes en el terreno y el interés del estudio, estos lugares de entrenamiento deben ser representativos de cada cobertura. (Gandía et al., 1991)
- **Clasificación no supervisada,** es aquella en la que el analista valida el resultado de la clasificación, es decir el usuario influencia únicamente la selección del algoritmo de clasificación. Para esta clasificación se utilizan algoritmos, los cuales realizan automáticamente la identificación de aquellos grupos que son diferentes de los ejemplos de entrenamiento en un espacio de características, donde se buscan las diferentes clases espectrales. También se utilizan métodos de comparación con píxeles cercanos. Un ejemplo de este tipo de algoritmos son, máxima verosimilitud, vecinos más próximos, Fmask. (Gandía et al., 1991)

La clasificación Basada en objetos se basa en un análisis basado en grupos de píxeles, con este tipo de clasificación trabajan los algoritmos de redes neuronales, puesto que mejora el reconocimiento de patrones, así como áreas de entrenamiento y validación a utilizar en el proceso de enmascaramiento. Este análisis se realiza en imágenes con una alta resolución, esto justifica el tipo de sensores empleados en el presente estudio. (Gandía et al., 1991)

## 1.2 Plataformas, satélites, sensores.

Para la realización del presente estudio se han empleado imágenes de media- alta resolución. En el caso del procedimiento de discriminación mediante redes neuronales, al trabajar más eficientemente con imágenes de alta resolución, se han utilizado imágenes provenientes de GBDX, tanto de WorldView-2 como de Quickbird (Digital Globe., 2020), las cuales poseen una resolución espacial de 30 centímetros. Para poder realizar una comparativa lo más correcta y

precisa con el enmascaramiento a nivel de píxel dado por Fmask, en esta situación se ha optado por utilizar imágenes de Sentinel- 2, del tipo 1C (Copernicus., 2020), puesto que este tipo son con las que permite trabajar Fmask. Estas imágenes poseen tres resoluciones espaciales diferentes a 10, 20 y 60 metros, en este caso se utilizaron las correspondientes a 10 y 20 metros.

A continuación, se muestran figuras de las diferentes características espectrales de los sensores utilizados para el presente trabajo, tanto de Sentinel-2 como de Worlview-2 y Quickbird. (Digital Globe., 2020)

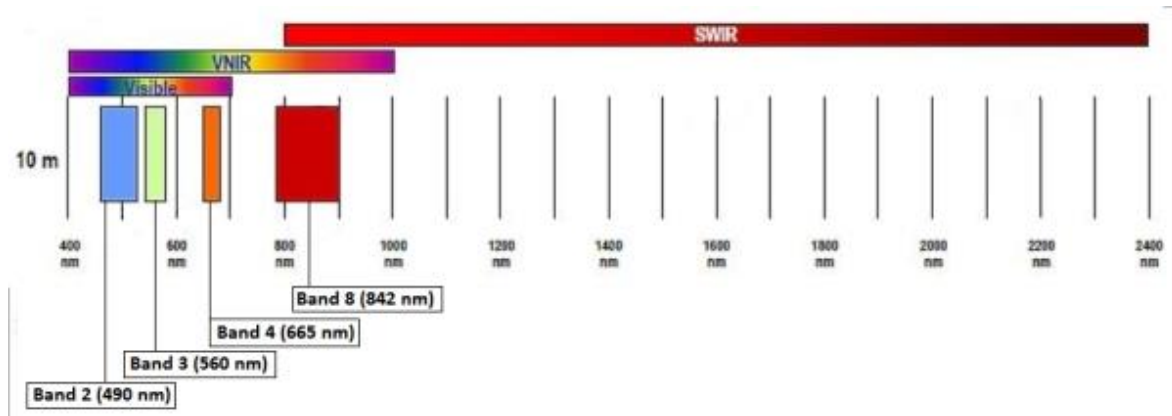


Figura 1. Bandas espectrales Sentinel-2 a 10 metros (Copernicus 2020)

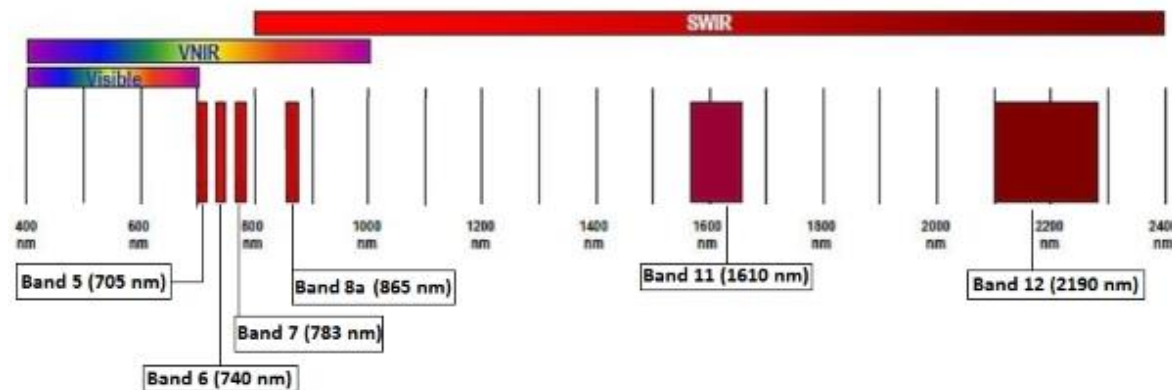


Figura 2. Bandas espectrales Sentinel- 2 a 20 metros (Copernicus 2020)

WorldView-2		QuickBird	
Bands	Spectral Resolution, nm	Bands	Spectral Resolution, nm
Panchromatic	450-800	Panchromatic	525-925
Coastal Blue	400-450	Blue	447-512
Blue	450-510	Green	499-594
Green	510-580	Red	620-688
Yellow	585-625	NIR	755-873
Red	630-690		
Red-Edge	705-745		
NIR1	770-895		
NIR2	860-1040		

Figura 3. Bandas espectrales WorlView-2 y Quickbird (Digital Globe 2020)

### 1.3 Análisis estadístico de resultados.

En el campo del Machine learning y, específicamente, el problema de la clasificación estadística, una matriz de confusión o matriz de error es un diseño de tabla específico que permite la visualización del rendimiento de un algoritmo. Las filas de la matriz representan las instancias en una clase predicha, mientras que las columnas representan las instancias en una clase real (o viceversa). El nombre se deriva del hecho de que hace que sea fácil ver si el sistema está confundiendo dos clases, es decir, comúnmente etiquetando incorrectamente una como otra.

Dicha matriz es un tipo especial de tabla de contingencia, con dos dimensiones "real" y "predicha", y conjuntos idénticos de "clases" en ambas dimensiones, siendo cada combinación de dimensión y clase una variable en la tabla de contingencia (Li et al., 2009).

La forma de matriz de confusión es la siguiente:

$$M = \begin{pmatrix} TP & FP \\ FN & TN \end{pmatrix} \quad (1)$$

- **True positive (TP):** Son los verdaderos positivos, en este caso serán píxeles u objetos que son nubosos y son detectados como nubosos.
- **False positive (FP):** Son los falsos positivos, en este caso serán píxeles u objetos que no son nubosos, pero son detectados como nubosos.
- **False negative (FN):** Son los falsos negativos, en este caso serán píxeles u objetos que son nubosos, pero no se detectan como tal.
- **True negative (TN):** Son los verdaderos negativos, píxeles u objetos que no son nubosos y no se detectan como nubosos.

A continuación, se describen diversos índices estadísticos que derivan de la matriz de confusión, cuyo análisis determinara baremos de precisión y rendimiento de los distintos algoritmos utilizados en el presente trabajo.

**Precisión observada (OP):** informa sobre la efectividad del algoritmo para realizar la tarea correspondiente. Se mide en tanto por ciento o tanto por uno.

$$OP = \frac{TP + TN}{(TP + TN + FP + FN)} \times 100 \quad (2)$$

**Precisión esperada (EP):** informa sobre la efectividad que se espera del algoritmo. Su unidad es el tanto por ciento o tanto por uno.

$$EP = \frac{(TP + FN)(TP + FP) + (FP + TN)(FN + TN)}{(TP + TN + FP + FN)^2} \times 100 \quad (3)$$

**Índice Kappa (K):** representa una medida de la fiabilidad y la concordancia entre dos procesos. En este caso aportará información sobre cómo de fiable y acorde es el algoritmo en el enmascaramiento de nubes. Los valores están comprendidos entre 0 y 1, siendo 0 nada fiable y concordante y 1 perfectamente fiable y acorde. Para ello utiliza los valores de precisión observada (OP) y precisión esperada (EP) en tanto por uno (Cohen, 1960).

$$K = \frac{OP - EP}{(1 - EP)} \quad (4)$$

**F1-Score(F):** es una medida de la precisión de una prueba o experimento. Alcanza su mejor valor en 1 y peor en 0. Se usa a menudo en el campo de la recuperación de información para medir la búsqueda, la clasificación de documentos y el rendimiento de la clasificación de consultas. Es un índice bastante utilizado en el campo del Machine Learning como indicador del rendimiento de diferentes algoritmos de clasificación o enmascaramiento.

$$F = \frac{2TP}{(2TP + FP + FN)} \quad (5)$$



## 1.4 Software empleado.

Para la realización de las diversas pruebas de estudio del presente trabajo se utilizó el siguiente software en su versión para Microsoft Windows 10.

Envi 5.3, Qgis 3.2. Arcgis 10, SAGA.

Adicionalmente la implementación del algoritmo Fmask así como el enmascaramiento mediante redes neuronales fueron realizados en Python 2 (Python Fmask., 2020) y Python 3 respectivamente, a través del entorno de trabajo Anaconda (Anaconda., 2020).

## **II. Algoritmos basados en análisis de píxeles**

---

## Resumen

---

Este capítulo plantea el desarrollo del algoritmo de enmascaramiento de nubes y sombras nubosas sobre imágenes de satélite Sentinel-2. Dicho algoritmo corresponde a un ejemplo de lo que es un método de enmascaramiento o clasificación basado en el nivel de píxeles.

En este caso se realiza un tratamiento previo de las imágenes satelitales para su posterior aplicación del algoritmo Fmask. Todo ello supone el método final de enmascaramiento aquí planteado.

### 2.1 Introducción.

En las últimas décadas, el enmascaramiento de nubes y sombras producidas por estas supone un reto para la teledetección. Algoritmos de diferente índole basados en la multitemporalidad de las imágenes (Goodwin et al., 2013; Lyapustin et al., 2008), así como la implementación de árboles de decisión cuyos elementos se producen mediante álgebra de bandas, han sido los más utilizados hasta el momento en el ámbito de la clasificación de imágenes basadas en el análisis de píxel.

En el presente capítulo se expone el proceso de implementación del algoritmo de enmascaramiento de nubes y sombras nubosas Fmask (Zhu et al., 2012). Dicho algoritmo trabaja sobre el píxel con la mecánica de proximidad y filtración de píxeles cercanos potencialmente nubosos. Utilizando la información espectral de brillo y temperatura, se generan máscaras de nubes en agua y tierra por separado.

A través de un proceso de segmentación, utilizando los píxeles potencialmente nubosos, así como la máscara generada anteriormente, se generan objetos de nubes.

Con la banda del infrarrojo cercano se genera una máscara de sombras potenciales. Utilizando la información del ángulo de toma y el ángulo de iluminación, con los objetos generados se predicen sombras. Todo esto se coteja con la máscara de sombras potenciales para generar la máscara de sombras final.

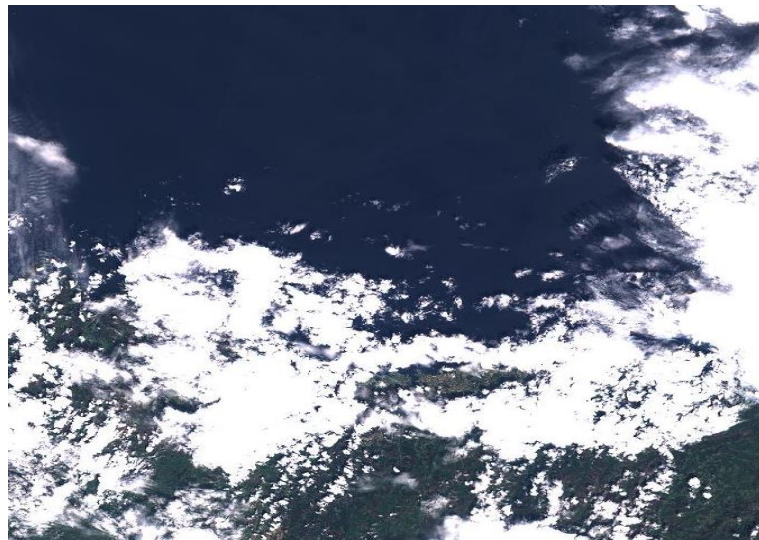
Previamente se realiza un pretratamiento de las imágenes de Sentinel-2 con vistas a mejorar la eficacia de Fmask.

El algoritmo total; pretratamiento y Fmask, se expondrá a continuación.

## 2.2 Zonas de estudio.

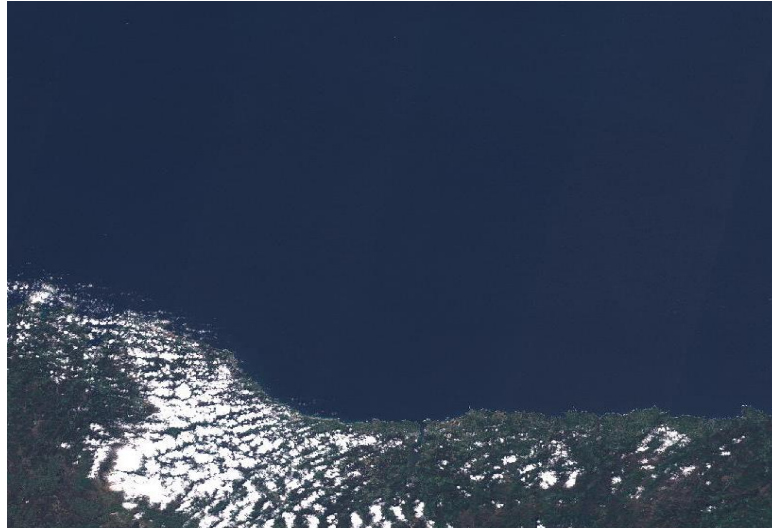
El presente sistema de enmascaramiento se realiza sobre diversas imágenes nubosas de la zona de Galicia. Para el desarrollo del algoritmo no es tan relevante la zona geográfica donde se realice sino la elección de una zona con variedad nubosa, alternando tierra y mar.

A continuación, se presentan fragmentos de las imágenes correspondientes para la prueba del algoritmo. El resto de las imágenes utilizadas son fragmentos de las imágenes descargadas, véase anexo V (Copernicus, 2020).



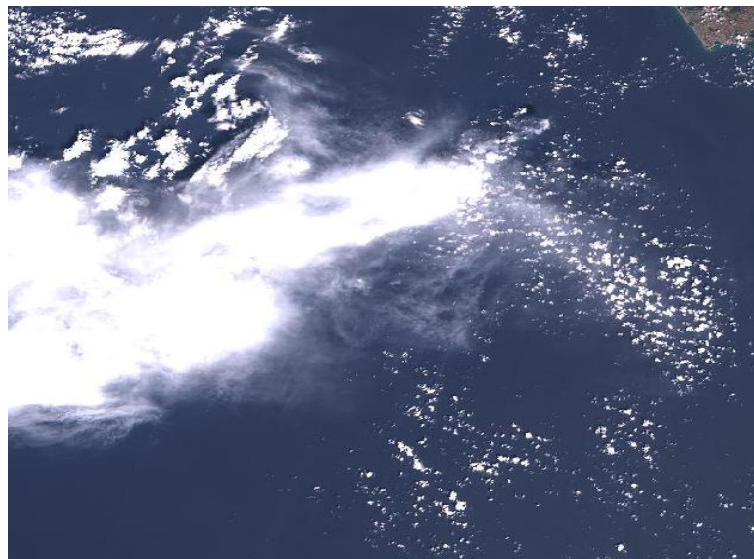
*Figura 4. Imagen de Sentinel- 2 correspondiente al noroeste de España (Copernicus., 2020)*

Este fragmento de imagen ha sido seleccionado por la variedad y numerosa cobertura nubosa que presenta. En este caso se observa que también se trata de abundante nubosidad en diferente entorno natural, contraste tierra – mar donde la bruma marina podría ser detectada o no por el algoritmo. Todo esto puede suponer un buen indicador de precisión y rendimiento del algoritmo de detección nubosa.



*Figura 5. Imagen de Sentinel-2 correspondiente al norte de España (Copernicus., 2020)*

En este caso se trata de nubes tipo altocúmulos, a priori no deben suponer ningún tipo de problema al ser detectadas.



*Figura 6. Imagen de Sentinel-2 correspondiente al mar Mediterráneo en las cercanías de la costa italiana (Copernicus., 2020)*

En esta imagen se puede apreciar que el escenario a trabajar es puramente marino. El agua tiene una respuesta espectral similar a las nubes en muchos casos, por lo que supone un reto añadido para el testeo de la algoritmia. Este fragmento se ha considerado importante por la variedad nubosa que presenta. Todo esto junto con lo mencionado anteriormente supone una prueba considerable de rendimiento y precisión.

## 2.3 Metodología.

El algoritmo Fmask, actualmente está desarrollado para trabajar con imágenes procedentes de Landsat 7,8 así como de Sentinel -2. En este caso se ha decidido utilizar imágenes Sentinel-2 1C, las cuales no tienen corrección atmosférica.

Para la implementación de Fmask, se ha utilizado la versión disponible para Python a través del entorno de Anaconda (Zhu et al., 2015).

El algoritmo completo consta de la aplicación de un tratamiento previo seguido de la posterior ejecución de Fmask.

### 2.3.1 Tratamiento preFmask.

Utilizando el álgebra de bandas se procede a calcular:

- Dos bandas sintéticas denominadas banda C y banda K, ayudando en la visualización y enmascaramiento de nubes bajas, medias y altas.

Una vez realizadas las operaciones algebraicas correspondientes, se procede a ecualizar el histograma para las bandas C y K con el fin de mejorar la visualización de cirros.

- Banda S para la mejora de visualización y enmascaramiento de sombras.

Las 3 bandas originadas, si una vez pasado a código están en formato JP2, se mantiene dicho formato. En el caso en el que se encuentren en formato tiff, se convierten a formato JP2.

Finalmente se llevan las bandas C, K y S a la carpeta donde se encuentra el resto de las bandas de Sentinel-2, la ruta estándar suele tener una dirección similar a esta "...GRANULE/L1C\_XXX.../IMG\_DATA"

## 2.4 Flujo de trabajo.

El flujo de trabajo esquematizado es el siguiente.

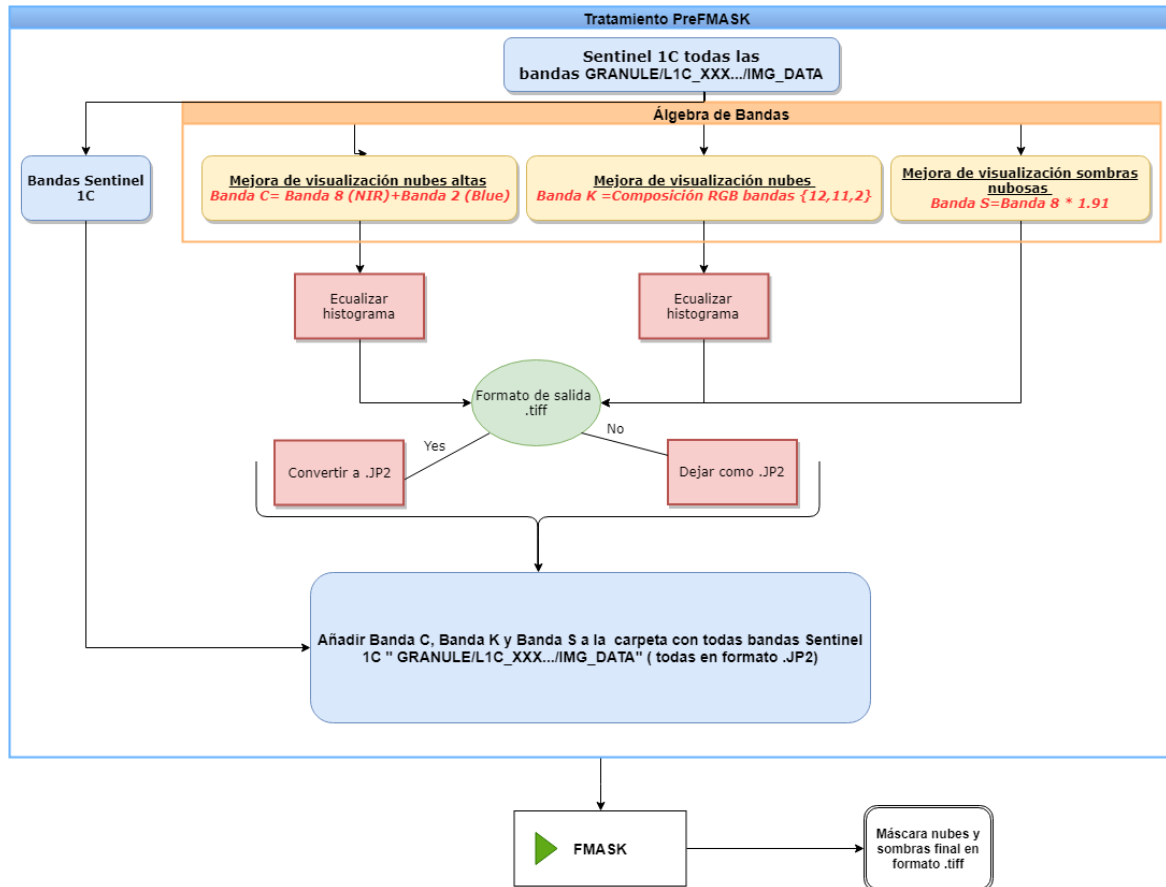


Figura 7. Diagrama de trabajo

### 2.4.1 Tratamiento preFmask.

Inicialmente se utilizan imágenes de Sentinel-2 1C, sin corrección atmosférica puesto que Fmask trabaja con esas y no con las 2A. Este aspecto no afecta al resultado obtenido en el pretratamiento.

A continuación, se procede a construir tres bandas sintéticas, C, K, S.

- La banda C es una banda resultado de sumar la banda 2 (azul) y la banda 8 de Sentinel-2, relativa al infrarrojo cercano, en esta banda las nubes tienen una mayor "reflectividad" aunque en la banda 1C no se usan realmente sus unidades, pero el resultado es el mismo. La banda azul también permite una

buena visualización de la cobertura nubosa puesto que en estas longitudes de onda (entorno a los 450 nm), esta radiación es la que interactúa mejor con las nubes bajas de todo el espectro.

- La banda K es una composición en el esquema de color RGB de las bandas 2, 11–y 12. Dicha composición es bastante favorable para la visualización de nubes puesto que trabaja con la inversión del canal rojo por el azul, así como dota al verde y al azul de los datos de las bandas térmicas 11 y 12, de manera que la percepción de nubes mejora considerablemente. Esto es debido a que en dichas bandas el vapor de agua está bastante presente por lo que, adecuándolo al canal verde y al azul (los canales que mejor visibilidad tienen a nivel fisiológico para el ojo humano), hacen posible una mejora en la visualización nubosa. Hay que recordar que se está trabajando con sensores ópticos, por lo que todo lo que mejore la visualización a nuestro ojo, a nivel de dato, facilitará el enmascaramiento.
- La banda S es una banda resultado del producto de la banda 8 relativa al infrarrojo cercano por una constante de 1,91. El valor de dicha constante hace posible que el resultado final sea una banda S, cercana a las bandas térmicas, pero mejorando la visualización de sombras. Se ha determinado trabajar en esta banda S puesto que en las zonas del espectro cercanas a bandas térmicas (longitudes de onda superiores a 1,5 nm), las sombras adquieren una notable importancia, pero sin llegar a la saturación final que dan las bandas térmicas y podrían interferir en el enmascaramiento de estas.

El paso siguiente pasa por ecualizar el histograma de manera que se obtiene una distribución de píxeles lo más uniforme posible para cada uno de los canales rojo, verde y azul de las bandas C y K. Este proceso es necesario para la visualización de nubes altas tipo cirros o neblinas bajas. Ecualizando el histograma se consigue repartir la intensidad de brillo de los diferentes canales (RGB) en todo el espectro de histograma, de tal forma que los blancos disminuyen su intensidad y los oscuros aumentan. Gracias a esto, la visualización de cirros y pequeñas brumas y neblinas mejora sustancialmente.

Si el resultado obtenido para las tres nuevas bandas, C, K, S está en formato .tiff se debe convertir a formato .JP2 para que Fmask pueda trabajar con ello. En caso de que el formato de salida del pretratamiento sea .JP2, se sigue con este formato.

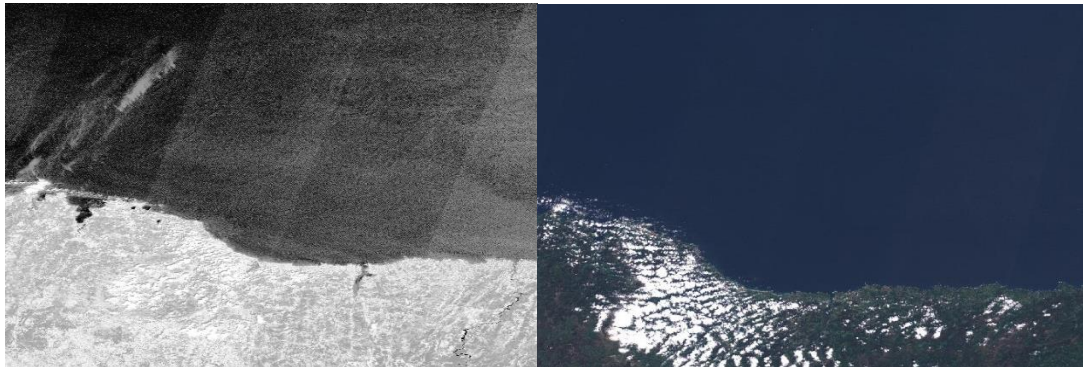
La última parte del pretratamiento se basa en mover las tres nuevas bandas a la carpeta donde se encuentran el resto de las bandas de Sentinel-2, la ruta tendrá una dirección similar a "...GRANULE/L1C\_XXX.../IMG\_DATA".



## 2.5 Resultados

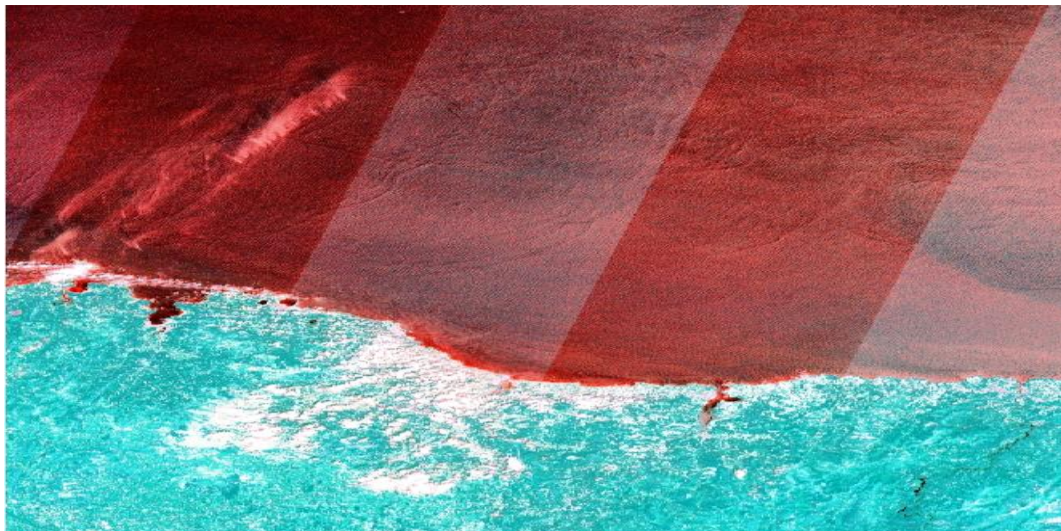
A continuación, se muestran los resultados obtenidos de bandas C, K, S, así como el resultado final del enmascaramiento de nubes y sombras.

### 2.5.1 Tratamiento preFmask



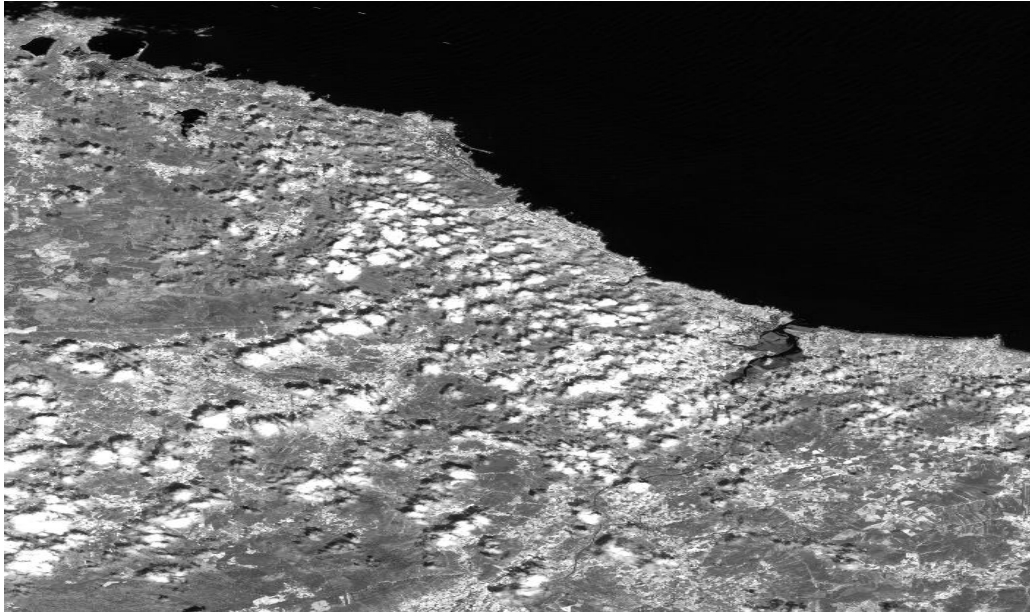
*Figura 8. Imagen comparativa Banda C Sentinel-2*

Banda C en la que se observa la aparición de cirros en la parte superior izquierda de la imagen.



*Figura 9. Imagen Correspondiente a la banda K originada para Sentinel-2*

Banda K en la que se pueden observar cirros y nubes altas en la parte superior izquierda de la imagen.

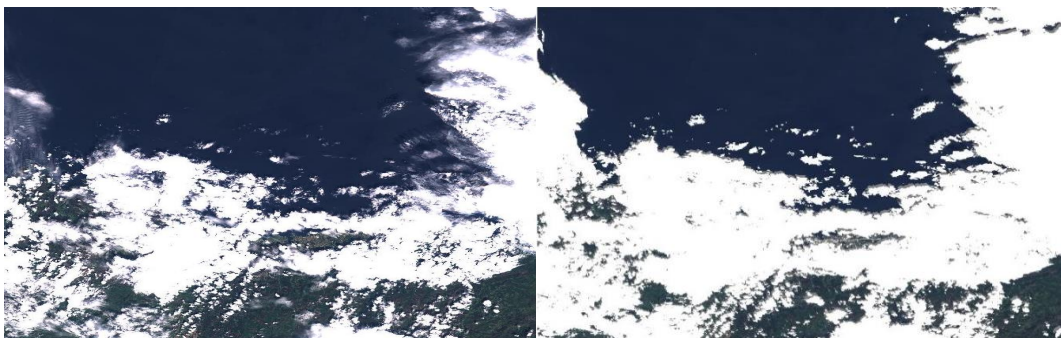


*Figura 10. Banda S Sentinel -2*

Banda S, mejora de visualización de sombras.

## 2.5.2 Fmask.

En las siguientes imágenes se aprecia una comparativa de las imágenes originales con sus respectivas imágenes enmascaradas.



*Figura 11. Comparativa de imagen 4 con su respectiva enmascarada. A la izquierda la imagen original, a la derecha la imagen obtenida mediante el algoritmo final Fmask.*



Figura 12. Comparativa de imagen 5 con su respectiva enmascarada. A la izquierda la imagen original, a la derecha la imagen obtenida mediante el algoritmo final Fmask.

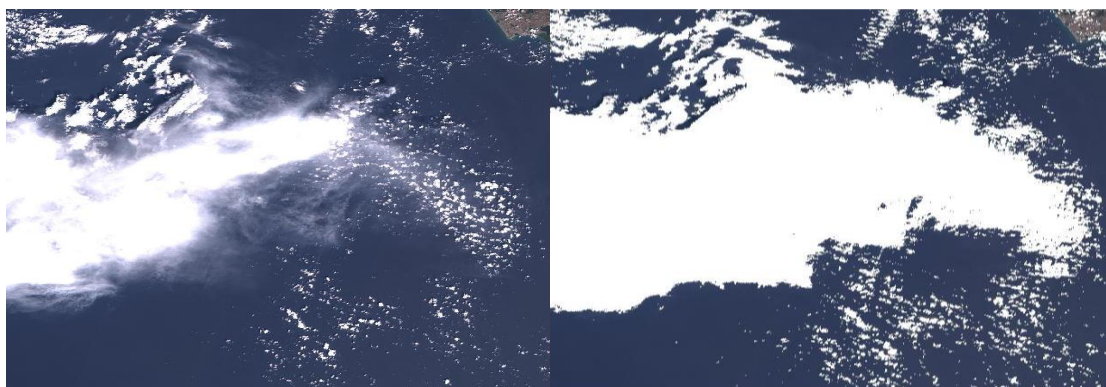


Figura 13. Comparativa de imagen 6 con su respectiva enmascarada. A la izquierda la imagen original, a la derecha la imagen obtenida mediante el algoritmo final Fmask.

## 2.6 Análisis de resultados.

A continuación, se exponen los resultados promedio para los 8 fragmentos de imagen utilizados a partir de las imágenes descargadas que figuran en el anexo V, como los resultados individuales para cada fragmento véase anexo V (Copernicus, 2020).

Indicador	Promedio	Img1	Img2	Img3	Img4	Img5	Img6	Img7
Precisión observada %	<b>87,82</b>	86,56	88,6	85,4	88,61	88,99	88,72	87,9
Precisión esperada %	<b>63,30</b>	60,23	64,34	61,54	63,23	65,91	63,74	64,06
Coefficiente Kappa	<b>0,67</b>	0,63	0,64	0,63	0,69	0,7	0,72	0,67
F1-Score	<b>0,74</b>	0,71	0,73	0,75	0,74	0,75	0,77	0,73

Tabla 1. Indicadores estadísticos.

Tanto el coeficiente Kappa como el índice F-score son buenos estadísticos para determinar el rendimiento de un clasificador. En este caso, el valor del coeficiente Kappa indica un alto nivel de concordancia en la clasificación entre los píxeles potencialmente nubosos o sombríos con los que realmente lo son, de ahí su valor cercano a 0,8.

El índice F-score considera tanto la precisión  $p$  como la recuperación  $r$  de la prueba para calcular la puntuación. En este caso posee un valor alto en el rango [0-1] por lo que indica que el algoritmo de clasificación estudiado posee un buen rendimiento como clasificador.

La precisión observada es alta, por lo que se puede deducir que Fmask junto con su pretratamiento inicial constituyen un buen método de enmascaramiento.

## 2.7 Conclusiones capitulares.

Ante la vista de los resultados obtenidos se ha deducido:

- El algoritmo de enmascaramiento tiene un buen porcentaje de acierto en cuanto al enmascaramiento de nubes bajas y sombras nubosas se refiere, no así en el caso de nubes altas.
- El hecho de implementar un algoritmo de enmascaramiento como Fmask (algoritmo ya hecho) supone pérdida de margen de maniobra y operatividad a la hora de mejorar el resultado final, concretamente en lo que se menciona antes, nubes altas, cirros, brumas.
- Este hecho condiciona que todo el proceso de tratamiento inicial preFmask, no sea del todo aprovechado. Si la imagen no contiene excesivas nubes altas, el tratamiento inicial y la construcción de las bandas C, K y S supone una mejora sustancial en el resultado final. En el caso contrario, el resultado final ira bien encaminado en cuanto a nubes bajas y sombras se refiere, pero mal encaminado cuando se trata de nubes más altas tipo cirros.

Finalmente, el tratamiento inicial está orientado a la mejora del enmascaramiento mediante algún tipo de algoritmo más eficiente que Fmask o que dé más margen a su manipulación.

En el caso de enmascaramiento o clasificación basada en el píxel, este algoritmo constituye un método fiable y preciso, siendo uno de los más utilizados por el momento, en cuanto a nubes y sombras nubosas se refiere.

# **III. Algoritmos basados en el análisis de objeto**

---

## Resumen

---

Este capítulo plantea el desarrollo del algoritmo de enmascaramiento de nubes sobre imágenes de satélite WorldView-2 y Quickbird. Dicho algoritmo corresponde a un ejemplo de lo que es un método de enmascaramiento o clasificación basado en el nivel de objetos.

En este caso se realiza un tratamiento previo de las imágenes satelitales para generar los archivos de entrenamiento y validación necesarios para el entrenamiento y aprendizaje de la red. Finalmente se ejecuta el algoritmo completo de enmascaramiento.

### 3.1 Introducción.

Multitemporalidad de las imágenes, así como en el análisis de píxel, han sido los algoritmos más utilizados hasta el momento para el enmascaramiento nuboso, un ejemplo de ello ha sido Fmask, analizado en el capítulo anterior. En las imágenes satelitales, las nubes limitan la cantidad de información de la superficie terrestre, causando valores atípicos en el análisis de las observaciones de reflectancia en el tiempo. Además, estas complican el uso de datos en el dominio óptico de los satélites de observación de la Tierra.

La información proporcionada por las bandas espectrales de las imágenes no es suficiente para determinar con certeza qué píxeles corresponden a zonas de sombra y cuáles no. Además, el perfil espectral de los píxeles de agua toma valores similares a los píxeles de nube y es por esto que pueden generar confusión al momento de identificar y clasificar los diferentes tipos de objetos presentes en la imagen.

Durante los últimos años, con el auge de la tecnología, tanto el aprendizaje profundo basado en inteligencia artificial como la implementación de redes neuronales, supone una nueva ventana esperanzadora ante tal reto.

En el presente capítulo se expone el proceso de implementación del algoritmo de enmascaramiento de sombras para imágenes de satélite en alta resolución de la plataforma Digital Globe. Al tratarse de imágenes de alta resolución, la eficacia de los algoritmos basados en Machine learning como en Deep learning suponen una ventaja ya que, al ser de un tipo de aprendizaje concreto, la precisión de estos es mayor.

A continuación, se expone la mecánica total del algoritmo.

## 3.2 Zonas de estudio.

El presente sistema de enmascaramiento se realiza sobre diversas imágenes GBDX de la zona de Galicia. Para el desarrollo del algoritmo no es tan relevante la zona geográfica donde se realicen sino el uso de pequeños fragmentos de imagen (chips) para la mejora de la precisión de enmascaramiento, etapa clave en el entrenamiento de la red neuronal. Para ello se han utilizado tres fragmentos de imagen provenientes de WorldView-2 y dos fragmentos provenientes de Quickbird a partir de las imágenes descargadas, véase anexo V (Digital Globe; 2020).

## 3.3 Metodología.

Todas las pruebas han sido realizadas utilizando imágenes de los sensores Worldview-2 y Quickbird (Digital Globe; 2020), de esta manera se obtiene diversidad de trabajo al utilizar 8 y 4 bandas espectrales. Previamente se separan las diferentes bandas espectrales de la imagen multiespectral, para su posterior manipulación.

### 3.3.1 Conceptos previos sobre redes neuronales.

Para el desarrollo de este algoritmo se propone el uso de la metodología correspondiente a la aplicación de Redes Neuronales Artificiales (RNA) (del inglés Artificial Neuronal Network), pertenecientes al conjunto de técnicas de machine learning. Las RNA son un método robusto de clasificación, el cual se basa en el funcionamiento de las neuronas del cerebro humano permitiendo que se realice un aprendizaje que va mejorando la clasificación.

Es un modelo matemático que está inspirado en el comportamiento biológico de las neuronas y en la estructura del cerebro, implementada para la solución de una gran variedad de problemas.

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información relevante, etc.

Las principales ventajas que representan son:

- **Aprendizaje Adaptativo:** Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- **Autoorganización:** Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.

- **Operación en tiempo real:** Los cálculos neuronales pueden ser realizados en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
- **Fácil inserción dentro de la tecnología existente:** Se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello facilitará la integración modular en los sistemas existentes. Su compatibilidad con los actuales lenguajes de programación es elevada, integrándose perfectamente en Python, Matlab, C++.

En su momento se contempló la posibilidad de realizar el algoritmo de enmascaramiento de sombras, mediante técnicas de Deep Learning. En esencia su base es la misma que las RNA con la diferencia que con estas técnicas, las redes neuronales poseen más capas intermedias y su método de aprendizaje es algo diferente. Estos matices les confieren mayor precisión, así como menor tiempo de computación si se hace uso de GPU en vez de CPU. En este caso se consultó la arquitectura computacional a utilizar y ante la evidente afirmación del uso de CPU se procedió a desestimar dicho recurso.

### 3.3.2 Estructura de las redes neuronales artificiales.

**La estructura de una red neuronal** consiste básicamente de tres partes: Una capa de entrada, una capa o varias capas ocultas donde se aplican funciones de activación, y una capa de salida que muestra las respuestas de la red a una entrada.

- **Capa de entrada:** Esta capa recibe las señales desde el entorno, en donde son presentados los datos a la red. Las variables que conforman estos datos de entrada reciben el nombre de inputs, en el caso del enmascaramiento de sombras será una variable que indique la presencia o no de sombra basado en un dato booleano de 0 ó 1.
- **Capa oculta:** Del mismo modo que el cerebro está constituido por neuronas, la unidad básica constituyente de la red neuronal es el nodo, neurona o elemento de procesado. Un nodo es un elemento de cálculo interconectado con otros muchos elementos, imitando las sinapsis nerviosas. A un nodo llegan conexiones desde muchas otras neuronas, proporcionando una respuesta que se muestra en la capa de salida. El nodo de la capa oculta condensa las variables de la capa de entrada con los respectivos pesos asignados aleatoriamente, estos reflejan la conexión entre las neuronas. Posteriormente se aplica una función que recibe el nombre de función de transferencia o de activación; la cual es en parte responsable de la bondad del comportamiento de la red neuronal. De esta manera se obtiene la salida del nodo, que a su vez podrá ser la entrada de algún otro nodo. En este caso, tras diversas pruebas realizadas, se determinó que el número óptimo de capas ocultas es 1 (Delgado & Quintín., 2003) y el número de neuronas de esta capa es 9.



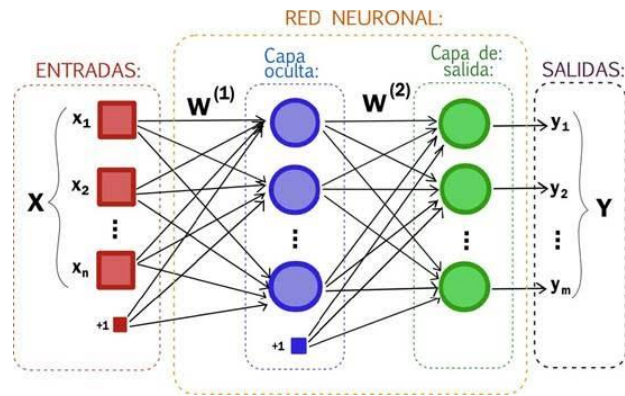


Figura 14. Estructura de una red neuronal artificial multicapa

Se recomienda utilizar el menor número posible de capas ocultas puesto que se reduce considerablemente el tiempo de procesamiento, y se evita el solapamiento de datos en el resultado final. Puesto que en este caso la única variable a manejar son las nubes, el número de capas ocultas es el mínimo.

**La arquitectura de una red** consiste en la disposición y conexionado de las neuronas. Se puede distinguir en una red, el número de capas, el tipo de las capas, que pueden ser ocultas o visibles, de entrada, o de salida y la direccionalidad de las conexiones de las neuronas.

- **Redes Monocapa:** cuentan con una capa de neuronas, que intercambia señales con el exterior y que constituyen a un tiempo la entrada y la salida del sistema.
- **Redes Multicapa:** están formadas por dos o más capas de neuronas conectadas entre ellas. Dependiendo de cómo sean estas conexiones se puede hacer otra subdivisión entre las redes con conexiones hacia delante y las redes con conexiones hacia atrás. Las primeras contienen solo conexiones entre capas hacia delante, esto implica que una capa no puede tener conexiones a una que reciba la señal antes que ella en la dinámica de la computación. Las segundas realizan conexiones de capas hacia atrás y por tanto la información puede regresar a capas anteriores en la dinámica de la red (Serrano et al., 2009). En este caso se utiliza una red multicapa, como ya se comentó anteriormente, debido a la presencia de su capa oculta y al posterior algoritmo de entrenamiento, del cual se hablará más adelante.

### 3.3.3 Parámetros en las redes neuronales artificiales.

**La función de activación empleada** fue la función sigmoide ya que el tiempo de convergencia de dicha función (tiempo que tarda en obtenerse un error mínimo de un orden prefijado en cuanto a la precisión de enmascaramiento) es el más pequeño posible, en comparación con otras funciones (Flórez & Fernández, 2008).

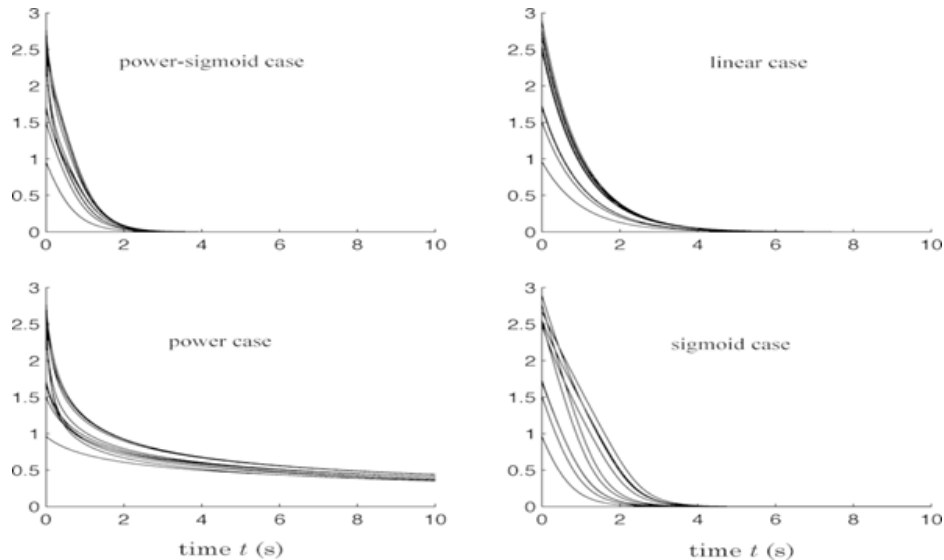


Figura 15. Tiempo de convergencia de diferentes funciones de activación.

- **El aprendizaje** se basa en el entrenamiento de la red con patrones. El proceso de aprendizaje se basa en que la red ejecute los patrones de forma iterativa hasta que se muestren respuestas satisfactorias. Es decir, los pesos sinápticos se ajustan para dar respuestas óptimas para el conjunto de patrones de entrenamiento. Se distinguen tres tipos de aprendizaje:
- **Aprendizaje Supervisado:** la red dispone de los patrones de entrada y de salida que queremos obtener para esa entrada, y en función de ellos se modifican los pesos de las sinapsis para ajustar la entrada a la salida. En el caso que se presenta se utiliza el aprendizaje supervisado, proporcionando diferentes archivos shape tanto para entrenamiento como validación.
- **Aprendizaje No Supervisado:** consiste en no proporcionar a la red los patrones de salida, sino sólo los de entrada y dejar que la red los clasifique en función de características comunes que encuentre entre ellos.

- **Aprendizaje Híbrido:** No se proporcionan los patrones objetivo, sino que sólo se dice si la respuesta acierta o falla ante un patrón de entrada.

En cuanto a los algoritmos de aprendizaje los más utilizados son:

- Back propagation
- Resilient propagation
- Descendent gradient method

Al igual que en el caso de la función de activación, debido al tiempo de convergencia del algoritmo de aprendizaje en función del número de iteraciones, se ha determinado que el algoritmo óptimo es Resilient propagation (Caicedo & Lopez ,2009).

En resumen, para la implementación del algoritmo de enmascaramiento de sombras mediante redes neuronales se han de tener en cuenta los siguientes parámetros.

Tipo de red	Multicapa, 1 capa oculta
Nº neuronas capa oculta	9
Funcion de activación	Sigmoide
Algoritmo de entrenamiento	Resilient propagation
Nº iteraciones entrenamiento	75000
Nº iteraciones validación	50000
Porcentaje entrenamiento/validación	65% entrenamiento, 35% validación
Error epsilon/ iteraciones hasta error epsilon	0.0001/50000

Tabla 2. Parámetros de red a utilizar

### 3.3.4 Generación de archivos vectoriales de entrenamiento y validación.

A continuación, se expone el método empleado para la obtención de archivos de entrenamiento como validación para el método de aprendizaje en la red neuronal.

El principal problema encontrado es que, a diferencia de otros enmascaramientos como pueden ser edificios, carreteras, no hay fuente constante de archivos de extracción de muestras como pueden ser las referencias catastrales. Adicionalmente, el crear un archivo de entrenamiento dinámico y versátil que sea útil para todas las imágenes supondría un tiempo de procesamiento bastante mayor, así como una disminución significativa en la precisión del algoritmo.

Tanto para el entrenamiento como para la validación, el método de enmascaramiento mediante redes neuronales necesita archivos en formato vectorial (shape) que sean objetos, es decir, conjuntos de píxeles agrupados en entidades mayores, en este caso polígonos.

Ante tal problemática se ha optado por el siguiente procedimiento:

- **Separación de bandas raster en imagen multiespectral y posterior extracción de chips:** en este apartado se procede a la extracción de las diferentes bandas espectrales para su posterior manipulación. Se recomienda recortar la imagen en pequeños fragmentos antes de la extracción de bandas, este recorte mejora la precisión de enmascaramiento final de la RNA. No se establece un tamaño determinado ya que cuando se proceda a su implementación se buscará el equilibrio entre precisión y coste computacional que conlleva implementarlo. Este recorte se realiza con el complemento SCP del software Qgis.
- **Cálculo del valor medio de las bandas relativas al visible:** se procede al cálculo valor medio de las bandas del espectro visible, el cual, para imágenes de 8 bandas, WorldView-2 viene dado por las bandas 2, 3 y 5. Para las imágenes de 4 bandas, Quickbird, se corresponden con las bandas 1, 2 y 3.

$$\overline{XVis} = \frac{(Br+Bg+Bb)}{3} \quad (1)$$

- **Cálculo de índice IFW:** como se comentó anteriormente, uno de los principales problemas del enmascaramiento de nubes es su confusión con las masas de agua, para ello se calcula este índice como discriminante de dichas masas acuosas. Este se obtiene a partir de la diferencia entre la primera banda del infrarrojo cercano y la banda visible verde.

$$WT = \sum_{i=r,g,b} \left| \frac{Bi - \overline{XVis}}{\overline{XVis}} \right| \quad (2)$$

- **Cálculo de índice WT:** se utilizará para ajustar la desviación estándar de los datos de las diferentes bandas del visible, más adelante este índice actuará a modo de “filtro” mejorando el enmascaramiento de nubes.

$$IFW = B_{NIR} - B_g \quad (3)$$

- **Ratio entre banda roja y azul del visible**

$$I = \frac{B_R}{B_B}$$

(4)

- **Teoría de Planck de radiación del cuerpo negro**

Una superficie uniforme está iluminada mediante diferentes fuentes de flujo luminoso; iluminación directa e iluminación dispersa. La cromaticidad constante de la superficie permite recuperar dos cromaticidades de los flujos luminosos de tal forma que se puede calcular la temperatura de cuerpo negro para dicha superficie. En este caso, dicha superficie es la nube, la cual posee zonas más luminosas y zonas más sombrías en su gran mayoría por la morfología de nube, así como la retroproyección de otras más elevadas.

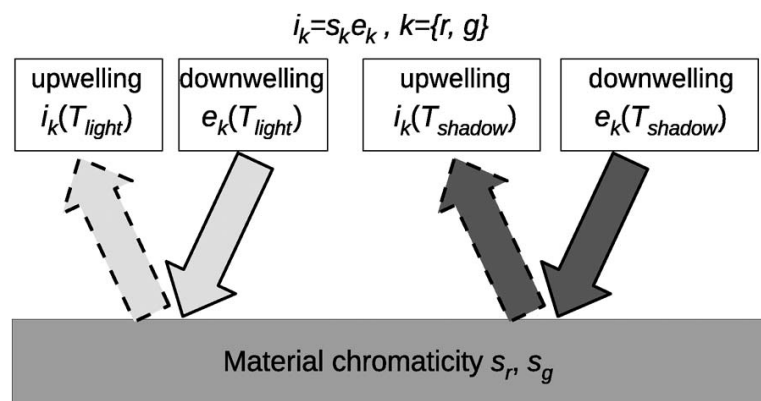


Figura 16. Proceso de obtención de cromaticidades en zonas nubosas con luz y con sombra

La fórmula de Planck para la radiancia espectral de cuerpo negro se define:

$$M(\lambda, T) = C_1 \lambda^{-5} \left[ e^{\frac{C_2}{T\lambda}} - 1 \right]^{-1}$$

(5)

Donde  $M(\lambda, T)$  es la potencia de radiación de cuerpo negro,  $C_1$  y  $C_2$  son dos constantes

( $C_1= 3.7418 \times 10^{-16} \text{ W m}^2$ , y  $C_2= 1.4388 \times 10^{-2} \text{ mK}$ ),  $\lambda$  es la longitud de onda en metros, y  $T$  la temperatura en Kelvin (debida a la luz solar tanto directa como debida a la dispersión de Rayleigh)

Este modelo puede utilizarse para calcular la cromaticidad de los flujos luminosos en el espectro del rojo (se puede usar también el espectro del verde, pero es preferible el rojo ya que al tratarse de imágenes de Digital Globe, se tiene mucha más variedad espectral en las zonas cercanas al rojo, con la banda del amarillo y la banda del borde rojo, está mejor definido el intervalo de esta zona del visible, lo que posibilita más precisión en los cálculos).

La cromaticidad del flujo luminoso puede ser representada por:

$$E_r(T) = \frac{M(\lambda_R, T)}{M(\lambda_B, T)} \tag{6}$$

Siendo  $\lambda_R$  y  $\lambda_B$  las longitudes de onda del rojo y del azul del visible, en metros. Combinando las fórmulas [5] y [6] y distinguiendo entre la zona de sombras y la de luz se obtiene:

$$E_r.Sombra = \frac{\lambda_B^5 \left[ e^{\left(\frac{C_2}{\lambda_B T_{sombra}}\right)} - 1 \right]}{\lambda_R^5 \left[ e^{\left(\frac{C_2}{\lambda_R T_{sombra}}\right)} - 1 \right]} \tag{7}$$

$$E_r.Luz = \frac{\lambda_B^5 \left[ e^{\left(\frac{C_2}{\lambda_B T_{luz}}\right)} - 1 \right]}{\lambda_R^5 \left[ e^{\left(\frac{C_2}{\lambda_R T_{luz}}\right)} - 1 \right]} \tag{8}$$

Utilizando los valores:

$$C_2= 1.4388 \times 10^{-2} \text{ mK}, T_{luz} = 6035\text{K}, T_{sombra}= 7962\text{K}, \lambda_B= 0.4805 \times 10^{-6}\text{m} \text{ y } \lambda_R= 0.665 \times 10^{-6}\text{m}$$

Se obtiene:

$$E_{r,sombra}= 0.584738$$

$$E_{r,luz}= 0.708495$$

**Estos valores son generales para todas las imágenes de GBDX. La minúscula variación de  $T_{luz}$  así como de  $T_{sombra}$  para un tipo u otro de sensor no afecta en el enmascaramiento final, por lo que estos valores se generalizan a todo tipo de sensor GBDX tanto de 8 como de 4 bandas.**

- **Condiciones premáscara raster**

- Obtención de banda denominada B', cuyo valor vendrá dado mediante:

$$B' = \left( \frac{I}{E_{r\text{sombra}}} - \frac{I}{E_{r\text{luz}}} \right) \quad (9)$$

Gracias a las cromaticidades de la luz directa y dispersa se puede obtener una banda B' de forma que permite una mejora de visualización de las nubes acotando significativamente el rango de valores de pixeles de manera que hace más sencilla su clasificación.

- Las condiciones que se deben cumplir para la creación de la máscara de nubes son las siguientes:

$$\left( \frac{WT}{B'} < 0.80 \right) \& (IFW > 900) \quad (10)$$

El ratio entre WT y B' establece una mejora en el enmascaramiento de nubes pues minimiza la desviación estándar de B' afinando aún más el rango de pixeles potencialmente nubosos.

Los parámetros 0.80 y 900 han sido establecidos mediante la realización de diferentes pruebas propias y se han considerado los óptimos.

- **Implementación filtro de mediana de kernel 3** para eliminar ruido y mejorar la visualización de la máscara.
- **Obtención de premáscara raster**: aplicando algebra de bandas con los índices anteriormente calculados, se obtiene una imagen raster de entrenamiento.
- **Trasformación premáscara de raster a vectorial**: se realiza la transformación de la premáscara raster a formato shape. Este archivo será la muestra shape de entrenamiento.
- **Obtención de muestras de validación**: se obtienen dos archivos shape de validación mediante la extracción aleatoria de polígonos de la muestra shape de entrenamiento. Se utilizan dos muestras de validación ya que, como en la ratio de entrenamiento /validación, se procede a emplear mayor porcentaje en entrenamiento para la red, de esta manera se mejora en la precisión de enmascaramiento. Se emplean herramientas del tipo Feature extraction.

### 3.4 Algoritmo de enmascaramiento de nubes.

A continuación, se expone el esquema general del algoritmo de enmascaramiento de nubes considerando todo lo expuesto anteriormente.

Para la implementación del algoritmo basado en redes neuronales se utilizan dos herramientas De Orfeo Toolbox (2020), antes de utilizar dichas herramientas se realiza la generación de los archivos en formato vectorial de entrenamiento:

**Train Images Classifier:** Esta aplicación realiza un entrenamiento clasificador a partir de múltiples pares de imágenes de entrada y datos vectoriales de entrenamiento. Las muestras se componen de valores de píxeles en cada banda. Los datos del vector de entrenamiento deben contener polígonos con un campo entero positivo que represente la etiqueta de la clase. El nombre de este campo se puede configurar con el parámetro "Campo de etiqueta de clase".

Las listas de muestra de capacitación y validación se crean de manera que cada clase esté representada por igual en ambas listas. Un parámetro permite controlar la relación entre el número de muestras en los conjuntos de entrenamiento y validación. Dos parámetros permiten administrar el tamaño de los conjuntos de capacitación y validación por clase y por imagen.

Se pueden establecer varios parámetros de clasificador dependiendo del clasificador elegido. En el proceso de validación, la matriz de confusión se organiza de la siguiente manera: filas = etiquetas de referencia, columnas = etiquetas producidas.

En el encabezado del archivo de salida de matriz de confusión opcional, las etiquetas de clase de validación (referencia) y predichas (producidas) se ordenan de acuerdo con las filas / columnas de la matriz de confusión. Esta aplicación se basa en LibSVM, OpenCV Machine Learning (2.3.1 y posterior) y Shark ML.

El resultado de esta aplicación es un archivo de modelo de texto, cuyo formato corresponde al tipo de modelo ML elegido. No hay imagen ni salida de datos vectoriales.

#### **Entradas de datos y valores en la herramienta Train Images Classifier:**

[Datos de entrada y salida]: este grupo de parámetros permite configurar datos de entrada y salida.

- **Lista de imágenes de entrada:** **Imagen multiespectral en formato tiff.**
- **Lista de datos vectoriales de entrada:** **archivo shape de entrenamiento.**
- **Lista de datos de vector de validación:** **dos archivos shape de validación.**
- **Modelo de salida:** archivo de salida que contiene el modelo estimado (**formato .xml**).
- **Limpieza de archivos temporales:** si está activada, la aplicación intentará limpiar todos los archivos temporales que se crearon.

[Parámetros de muestras de capacitación y validación]: este grupo de parámetros permite establecer parámetros de listas de muestras de entrenamiento y validación.

- **Tamaño máximo de muestra de entrenamiento por clase:** **75000.**
- **Tamaño máximo de muestra de validación por clase:** **50000**
- **Proporción de muestra de entrenamiento y validación:** Relación entre muestras de entrenamiento y validación (0.0 = todo el entrenamiento, 1.0 = toda la validación) (**en este caso 0.35**).



- **Campo que contiene la etiqueta de entero de clase para supervisión:** campo que contiene la identificación de clase para supervisión. Los valores en este campo se convertirán en enteros. **En este caso DN (nivel digital)**

**[Clasificador a usar para el entrenamiento]: Elección del clasificador a usar para el entrenamiento.**

- **Clasificador de la red neuronal artificial:** este grupo de parámetros permite configurar los parámetros del clasificador de la red neuronal artificial. La documentación completa.
- **Método de entrenamiento:** Tipo de método de entrenamiento para la red neuronal perceptrón multicapa (MLP). **Algoritmo Resilient propagation.**
- **Número de neuronas en cada capa intermedia:** **9**
- **Tipo de función de activación de neurona:** esta función determina si la salida del nodo es positiva o no, dependiendo de la salida de la función de transferencia. **Función sigmoide.**
- **Parámetro alfa de la función de activación:** parámetro alfa de la función de activación (utilizado solo con funciones sigmoideas y gaussianas). **En este caso 0.8**
- **Parámetros beta de la función de activación:** Parámetros beta de la función de activación (utilizado solo con funciones sigmoideas y gaussianas). **En este caso 1**
- **Criterios de terminación: todos**
  - Número máximo de iteraciones: 50000**
  - Valor de Epsilon utilizado en los criterios de terminación: 0.0001**

Los demás parámetros que puedan aparecer se deben dejar con su configuración por defecto. Aquí se han expuesto aquellos más determinantes y cuya configuración hay que cambiar para conseguir el resultado buscado.

**Image Classifier:** Esta aplicación realiza una clasificación de imágenes basada en un archivo de modelo producido por la aplicación Train Images Classifier. Los píxeles de la imagen de salida contendrán las etiquetas de clase decididas por el clasificador.

#### **Entradas de datos y valores en la herramienta Images Classifier:**

- **Imagen de entrada:** la imagen de entrada para clasificar en formato tiff, debe ser la misma que la utilizada para entrenar.
- **Máscara de entrada:** dejar vacío por defecto.
- **Archivo de modelo:** **un archivo de modelo .xml (producido por la aplicación TrainImagesClassifier).**
- **Imagen de salida:** **imagen de salida que contiene etiquetas de clase. Imagen tiff que se va a generar.**
- **RAM disponible (Mb):** **memoria disponible para procesamiento (en MB), modificar en función de maquina/s donde se ejecute.**

Los demás parámetros que puedan aparecer se deben dejar con su configuración por defecto. Aquí se han expuesto aquellos más determinantes y cuya configuración hay que cambiar para conseguir el resultado buscado.

### 3.5 Flujo de trabajo.

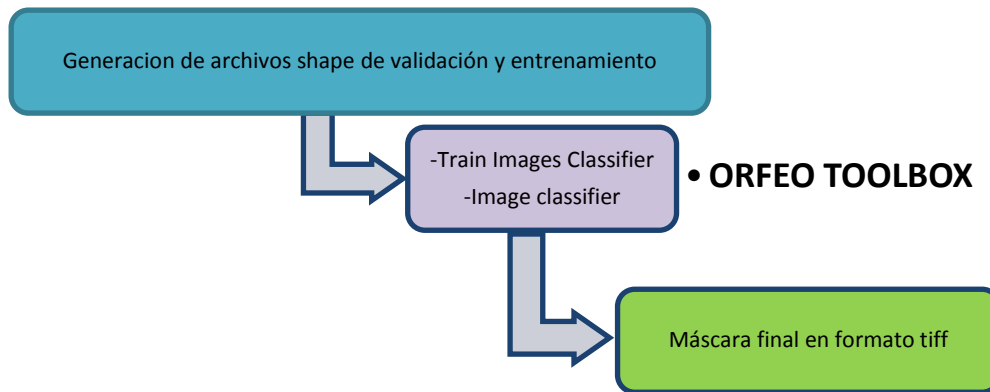


Figura 17. Diagrama de trabajo del algoritmo

Para la implementación a código, si es más simple utilizar la imagen multiespectral, trabajando internamente con sus bandas sin hacer una separación tal cual, de estas, el algoritmo seguiría funcionando correctamente.

### 3.6 Análisis de resultados



*Figura 18. Imagen de Quickbird relativa a la zona norte de España. A la izquierda la imagen multiespectral, a la derecha la imagen con enmascaramiento nuboso (Digital Globe., 2020).*



*Figura 19.. Imagen de WorldView-2 relativa a la zona norte de España. A la izquierda la imagen multiespectral, a la derecha la imagen con enmascaramiento nuboso. (Digital Globe., 2020).*



Figura 20 . Imagen de WorldView-2 relativa a la zona norte de España. A la izquierda la imagen multispectral, a la derecha la imagen con enmascaramiento nuboso (Digital Globe., 2020).

A continuación, se exponen los resultados de los indicadores estadísticos promedios de los tres fragmentos de imagen WorldView-2 y dos fragmentos de imagen Quickbird. Dichos fragmentos están obtenidos a partir de las imágenes que figuran en el anexo V para estos dos satélites.

Indicador	Promedio	Img1 WV-2	Img2 WV-2	Img3 WV-2	Img4	Img5
Precisión observada %	<b>92,23</b>	92,89	91,87	91,88	92,39	92,12
Precisión esperada %	<b>49,80</b>	49,71	49,87	49,72	49,83	49,89
Coefficiente Kappa	<b>0,84</b>	0,82	0,83	0,84	0,84	0,85
F-score	<b>0,92</b>	0,91	0,9	0,92	0,93	0,93

Tabla 3. Indicadores estadísticos

Tanto el coeficiente Kappa como el índice F-score son buenos estadísticos para determinar el rendimiento de un clasificador. En este caso el valor del coeficiente Kappa indica un alto nivel de concordancia en la clasificación entre los píxeles potencialmente nubosos con los que realmente lo son, de ahí su valor cercano a 0,85.

El valor obtenido en F1-Score, 0,92, indica que el algoritmo de enmascaramiento basado en redes neuronales tiene un alto porcentaje de concordancia y desarrolla un buen rendimiento en la tarea de discriminación nubosa.

### 3.7 Conclusiones capitulares.

Las redes neuronales como método de clasificación es un buen método de discriminación y se presenta como una herramienta en el tratamiento digital de imágenes satelitales, permitiendo realizar la discriminación y análisis en el comportamiento de las diferentes coberturas, en este caso se utilizó en nubes, pero también podría ser aplicable a cualquier tipo de cobertura (pastos, nubes, bosques, sombras...etc.).

Es de notar que para los archivos de entrenamiento en general se recomienda que la información de las semillas sean polígonos en vez puntos o de líneas, debido a que se capturan de mejor forma los comportamientos de la cobertura a tratar, balanceando la clase que se quiera discriminar con respecto a todos los comportamientos que tenga en el espacio.

La distribución de las semillas (polígonos) juega un papel importante en el resultado final, así como la proporción de toma de estas.

La búsqueda de la arquitectura más apropiada para una red representa un extenuante trabajo que involucra la realización de numerosas pruebas, un mayor nivel de referencia en el funcionamiento operativo de la misma, aunque los resultados arrojados son bastante satisfactorios.

Para finalizar, hay que añadir que la mejora en este método de enmascaramiento pasa por optimizar los archivos vectoriales de entrenamiento y validación; parte fundamental en el tema de redes neuronales. Una forma de mejorar dicho entrenamiento podría pasar por darle a la red información adicional sobre el relieve del terreno, archivos de elevación de terreno (DEM, SRTM) y utilizar técnicas de máxima probabilidad que condicionen las regiones potencialmente nubosas de manera que permita hacer una selección de polígonos más eficiente con vistas a la mejora de la precisión de la red.

## **IV. Conclusiones**

---

En este trabajo se utilizaron dos metodologías para la detección de nubes en imágenes satelitales de media resolución provenientes de Sentinel-2 y alta resolución proveniente de WoldrView-2 y Quickbird.

En el primer caso se utilizó un método basado en análisis por píxel; Fmask, mientras que en el segundo caso se optó por utilizar redes neuronales artificiales.

El método Fmask fue implementado en este estudio debido a que no requiere de datos de entrenamiento y a su capacidad de considerar la información temporal de cada píxel, permitiendo establecer un umbral dependiendo del comportamiento de la cobertura a partir del supuesto de que existe un punto de inflexión entre las observaciones libres y cubiertas de nubes.

En cuanto a las Redes Neuronales Artificiales, fueron utilizadas, ya que, a diferencia de los enfoques clásicos de inteligencia artificial, tienen la capacidad de aprender de experiencias pasadas mediante datos de entrenamiento, pueden extraer las características principales a partir de entradas que presenten información irrelevante y cuentan con la capacidad de adaptarse y auto-organizar la información, mejorando el rendimiento y precisión en aplicaciones futuras.

No obstante, se requiere cierta experiencia y cuidado al hacer la configuración de la red puesto que una arquitectura mal estructurada hace que se limite su desempeño (problemas de sobre y sub-entrenamiento) y que aumente la carga computacional.

En el enmascaramiento mediante redes neuronales, para las clasificaciones en general se recomienda que la información de las muestras provenga de polígonos en vez puntos o de líneas, debido a que se capturan de mejor forma los comportamientos de las coberturas, en este caso cubierta nubosa, balanceando la clase que se quiera discriminar con respecto a todos los comportamientos que tenga en el espacio. La distribución de las semillas, polígonos de entrenamiento, juega un papel importante en el resultado final, así como la proporción de toma de estas, pues el algoritmo asignará mayor ponderación a la cobertura que más se presente sobre el territorio. Si está cuestión se suma al hecho de la cercanía espectral de las coberturas, generará resultados confusos y coberturas excesivamente superpuestas.

La comparación entre los resultados de máscaras de nubes de cada método implementado permite establecer que tanto RNA como Fmask presentan en promedio valores de concordancia del coeficiente kappa de 0.84 y 0.67 respectivamente. Esto indica que ambos métodos presentan fuerza de concordancia considerable y moderada. Además, se observó que la concordancia aumenta en función del porcentaje de nubosidad, es decir, cuantas más nubes estén presentes en la zona, mayor es la similitud y menor será el error entre estos.

Se recomienda que, para futuros estudios en los que se implementen redes neuronales artificiales para la detección de nubes, se utilice como parte del pre-procesamiento de los datos de entrada de la red la aplicación de filtros espaciales que permitan extraer más información de la contenida en la imagen, de esta manera aumentara la precisión de la red en el proceso de enmascaramiento.

## **V. Referencias**

---



Anaconda. (2020). <https://anaconda.org>

Anaconda Python Fmask. (2020). <https://anaconda.org/conda-forge/python-fmask>

Ayala, R., & Menenti, M. (2001). Metodología para la búsqueda del mejor clasificador de imágenes de satélite. *Teledetección, Medio Ambiente y Cambio Global*: 469-472.

Caicedo, E.F. & López, J.A. (2009). *La Aproximación Práctica a Las Redes Neuronales Artificiales*. Programa Editorial Universidad del Valle.

Chakraborti, S. (2007). Verification of the Rayleigh scattering cross section. *Amer. J. Phys.*, volumen. 75, no. 9: 824–826.

Chuvieco, E. (1996). *Principios de Teledetección Espacial (Rialp, Segunda Ed)*. Madrid.

Chuvieco, E. (2010). *Teledetección Ambiental, La observación de la Tierra desde el Espacio*. (Planeta S.A., Ed.). Barcelona.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educ Psychol Meas* 20: 37-46.

Copernicus. (2020). <https://scihub.copernicus.eu/dhus/#/home>

Digital Globe. (2020). <https://discover.digitalglobe.com/>

Fawcett, T. (2006). An Introduction to ROC Analysis. *Pattern Recognition Letters*. 27 (8): 861–874.

Fleiss, J.L. (2000). *Statistical methods for rates and proportions*. (Wiley). Columbia.

Flórez López, R., & Fernández Fernández, JM. (2008). *Las Redes Neuronales Artificiales, Fundamentos Teóricos Y Aplicaciones Prácticas*. (Netbiblo). La Coruña.

Gandía, S. & Melia, J. (1991). *La teledetección en el seguimiento de los fenómenos naturales. Recursos renovables:Agricultura*.(Universitat de València).Valencia.

Goodwin, N.R., Collett L.J., Denham R. J., Flood N. (2013). Remote Sensing of Environment Cloud and Cloud Shadow Screening across Queensland, Australia: An Automated Method for Landsat TM / ETM + Time Series. *Remote Sensing of Environment* 134.

Hara, K. & Nakayama, K. (1994). Comparison of activation functions in multilayer neural network for pattern classification. *IEEE World Congress on Computational Intelligence*. Volumen 5 :2997 - 3002.

Heremans, S., Bossyns, B., Eerens, H., & Van Orshoven, J. (2011). Efficient collection of training data for sub-pixel land cover classification using neural networks. *International Journal of Applied Earth Observation and Geoinformation*, 13(4): 657-667.

Hilera J. R. & Martínez V. J. (1995) *Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones*.

Irish, R.R., Barker, J.L., Goward, S.N., & Arvidson, T. (2006). Characterization of the Landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm. *Photogrammetric Engineering & Remote Sensing*, 72: 1179–1188.

Kawakami, R. & Takamatsu J. (2007). Color constancy from blackbody illumination. *J. Opt. Soc. Amer. A, Opt. Image Sci*. Volumen. 24: 1886–1893.

Kennedy, A.D., Dong X., Xi B., Minnis P., Wolf A.B., Khaiyer, M.M. (2010). Evaluation of the NASA GISS Single-Column Model Simulated Clouds Using Combined Surface and Satellite Observations. *Journal of Climate*.

Kwok, T., & Yeung, D.Y. (1997). Objective Functions for Training New Hidden Units in Constructive Neural Networks. *IEEE, Transactions on Neural Networks*. Volumen .8. No 5.

Martinuzzi, S., Gould, W., & González O. (2007). *Creating Cloud-Free Landsat ETM+ Data Sets in Tropical Landscapes: Cloud and Cloud-Shadow Removal*.

Open cv. (2020). [http://docs.opencv.org/modules/ml/doc/neural\\_networks.html](http://docs.opencv.org/modules/ml/doc/neural_networks.html)

Orfeo. (2020). <https://www.orfeo-toolbox.org/>

Orfeo Toolbox. (2020). [https://www.orfeo-toolbox.org/CookBook/Applications/app\\_ImageClassifier.html](https://www.orfeo-toolbox.org/CookBook/Applications/app_ImageClassifier.html)

Orfeo Toolbox. (2020). [https://www.orfeo-toolbox.org/CookBook/Applications/app\\_TrainImagesClassifier.html](https://www.orfeo-toolbox.org/CookBook/Applications/app_TrainImagesClassifier.html)

Pao H. (1989). *Adaptive Pattern Recognition and Neural Networks*. (Addison-Wesley). Boston.

Powers, D. M. W. (2011). *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation*. *Journal of Machine Learning Technologies*. 2: 37–63.

Python Fmask (2020) <http://www.pythonfmask.org/en/latest/#downloads>

Riedmiller M. (1994). *Advanced Supervised Learning in Multi-Layer Perceptrons From Backpropagation to Adaptive learning Algorithms*. *Computer Standards and Interfaces*, 5. Special Issue on Neural Networks.

Sasaki, Y. (2007). *The truth of the F-measure*.

Stehman, S. V. (1997). *Selecting and interpreting measures of thematic classification accuracy*. *Remote Sensing of Environment*. 62: 77–89.

Taylor, JR. (1999). *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*.

Xie, F., Shi, M. & Shi. Z. (2017). *Multilevel cloud detection in remote sensing images based on deep learning*. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, volumen 10: 3631–3640.

Zhu, Z., Wang, S. & Woodcock, C.E. (2015). *Improvement and expansion of the Fmask algorithm: cloud, cloud shadow, and snow detection for Landsats 4-7, 8, and Sentinel 2 images*. *Remote Sensing of Environment* 159: 269-277.

Zhu, Z. & Woodcock, C.E. (2012). *Object-based cloud and cloud shadow detection in Landsat imagery*. *Remote Sensing of Environment* 118: 83-94.

## **VI. Anexos**

---

## Anexo I. Glosario de términos

---

**Banda espectral:** es una selección de longitudes de onda con comportamientos electromagnéticos similares.

**Imagen multiespectral:** o multibanda es un conjunto de imágenes, con las mismas propiedades geométricas, cada una de las cuales recoge el valor de reflectancia en un determinado intervalo de longitud de onda del espectro electromagnético.

**Quickbird:** satélite que circunda la tierra en órbita polar a 450 kilómetros de altura. Es un satélite único en su tipo permitiendo obtener imágenes con una resolución (comparable a una fotografía aérea 1:4.000) de 70 cm, pancromático (blanco/negro)-la más alta del mundo- y 2.8 m multiespectral (color), colocando a disposición de la comunidad mundial abundante información acerca de las siempre cambiantes características naturales de la Tierra; siendo una gran contribución para la consolidación final de la industria (de satélites operacionales) de alta resolución

**Reflectancia o reflectividad:** del cuerpo varía según la longitud de onda que incide sobre ellos, normalmente la reflectividad aumenta a medida que la longitud de onda disminuye, no obstante, existen cuerpos donde esta relación es inversa (nieve) o irregular (vegetación). También la reflectividad está influenciada por la textura del cuerpo, para una misma longitud de onda, la reflectividad más baja se encuentra en el agua, aumenta para el suelo, vegetación enferma, vegetación sana y es máxima para la nieve.

**Serie multitemporal:** La comparación de imágenes satélite obtenidas a determinados intervalos de tiempo, permite estudiar fenómenos que implican una variación temporal, como, por ejemplo, el proceso de deforestación amazónica.

**Vectorial:** Modelo de datos de un SIG basado en entidades u objetos geométricos definidos por las coordenadas de sus nodos y vértices. También denominado shape.

**XML:** Extensible Markup Language

## Anexo II. Proceso Fmask

---

### Capturas de la implementación de Fmask mediante consola.

```
MASCARA DE NUBES

('ImagePaths', '/data/images/preprocesado/20170501_20170501_20190809T101853/S2A_MSIL1C_2017
21_N0205_R037_T29TPJ_20170525T112434.SAFE')
('OutputMask', '/data/images/preprocesado/20170501_20170501_20190809T101853/output/S2A_MSIL1C
25T112121_N0205_R037_T29TPJ_20170525T112434_fmask.tif')
('Imagenes .SAFE', '/data/images/preprocesado/20170501_20170501_20190809T101853/S2A_MSIL1C_
112121_N0205_R037_T29TPJ_20170525T112434.SAFE')
('Mascara ', '/data/images/preprocesado/20170501_20170501_20190809T101853/output/S2A_MSIL1C
T112121_N0205_R037_T29TPJ_20170525T112434_fmask.tif')
Making angles image
Making stack of all bands, at 20m pixel size
Cloud layer, pass 1
Saturation mask not supplied - saturated areas may not be detected
  Twater= None Tlow= None Thigh= None NIR_17= 0.196
Cloud layer, pass 2
  landThreshold= 0.8
Cloud layer, pass 3
Potential shadows
Clumping clouds
Making 3d clouds
Making cloud shadow shapes
Matching shadows
No shadow found for 41 of 3484 clouds
Doing final tidy up
finished fmask
El proceso de creación de la mascara de nubes tardó 400.911633015 segundos
('Mascara de nubes reclasificada ', '/data/images/preprocesado/20170501_20170501_20190809T1
put/S2A_MSIL1C_20170525T112121_N0205_R037_T29TPJ_20170525T112434_fmask_reclass.tif')
Done!
```

Implementación en consola de Windows del algoritmo final Fmask en el cual la entrada son imágenes preprocesadas provenientes de Sentinel-2, una vez que se aplica el algoritmo estas son guardadas en el directorio que se indique con la extensión adicional `_fmask` en formato `.tif`

## Generacion de diferentes imágenes enmascaradas mediante Fmask.

```
('Tipo de dato', 'UInt16')
IMAGENES SENTINEL
Todas Bandas Sentinel

  IMAGEN /data/images/preprocesado/20170501_20170501_20190809T101853/S2A_MSIL2A_20170525T11
2121_N0205_R037_T29TPJ_20170525T112434.SAFE/GRANULE/L2A_T29TPJ_A010044_20170525T112434/IMG_
DATA/R60m/L2A_T29TPJ_20170525T112121_B01_60m.jp2

Origen Imagen 1= (600000.0, 4900020.0)
Tamaño pixel = (60.0, -60.0)
('Numero de bandas ', 1, '\n')
Tamaño imagen = (1830, 1830)
('Tipo de dato', 'UInt16')
IMAGENES SENTINEL
Todas Bandas Sentinel

  IMAGEN /data/images/preprocesado/20170501_20170501_20190809T101853/S2A_MSIL2A_20170525T11
2121_N0205_R037_T29TPJ_20170525T112434.SAFE/GRANULE/L2A_T29TPJ_A010044_20170525T112434/IMG_
DATA/R10m/L2A_T29TPJ_20170525T112121_B03_10m.jp2

Origen Imagen 1= (600000.0, 4900020.0)
Tamaño pixel = (10.0, -10.0)
('Numero de bandas ', 1, '\n')
Tamaño imagen = (10980, 10980)
('Tipo de dato', 'UInt16')
IMAGENES SENTINEL
Todas Bandas Sentinel

  IMAGEN /data/images/preprocesado/20170501_20170501_20190809T101853/S2A_MSIL2A_20170525T11
2121_N0205_R037_T29TPJ_20170525T112434.SAFE/GRANULE/L2A_T29TPJ_A010044_20170525T112434/IMG_
DATA/R10m/L2A_T29TPJ_20170525T112121_B08_10m.jp2

Origen Imagen 1= (600000.0, 4900020.0)
Tamaño pixel = (10.0, -10.0)
('Numero de bandas ', 1, '\n')
Tamaño imagen = (10980, 10980)
('Tipo de dato', 'UInt16')
Done!
El preprocesado tardó 6148.68571711 segundos
('Salida en el directorio ', '/data/images/preprocesado/20170501_20170501_20190809T101853')
(tierra3) [root@Tierra3Desarrollo bin]#
```

## Ruta típica de guardado de imagen enmascarada



Se puede observar tanto la ruta ( imagen de abajo) como el archivo generado una vez aplicado Fmask ( ultimo archivo en formato tif, de la imagen de arriba)

## Otro ejemplo de generación de imágenes enmascaradas

```
_____ MASCARA DE NUBES _____
('ImagePaths', '/data/images/L1C//S2A_MSIL1C_20170525T112121_N0205_R037_T29TPJ_20170525T112434.SAFE'
)
('OutputMask', '/data/images/L1C/fmask/S2A_MSIL1C_20170525T112121_N0205_R037_T29TPJ_20170525T112434_
fmask.tif')
('Imagenes .SAFE', '/data/images/L1C//S2A_MSIL1C_20170525T112121_N0205_R037_T29TPJ_20170525T112434.S
AFE')
('Mascara ', '/data/images/L1C/fmask/S2A_MSIL1C_20170525T112121_N0205_R037_T29TPJ_20170525T112434_fm
ask.tif')
Making angles image
Making stack of all bands, at 20m pixel size
Cloud layer, pass 1
Saturation mask not supplied - saturated areas may not be detected
  Twater= None Tlow= None Thigh= None NIR_17= 0.196
Cloud layer, pass 2
  landThreshold= 0.8
Cloud layer, pass 3
Potential shadows
Clumping clouds
Making 3d clouds
Making cloud shadow shapes
Matching shadows
No shadow found for 41 of 3484 clouds
Doing final tidy up
finished fmask
El proceso de creación de la mascara de nubes tardó 394.907718182 segundos
('Mascara de nubes reclasificada ', '/data/images/L1C/fmask/S2A_MSIL1C_20170525T112121_N0205_R037_T2
9TPJ_20170525T112434_fmask_reclass.tif')
Done!
```

```
T29TPJ_A010044_20170525T112434/IMG_DATA/R60m/L2A_T29TPJ_20170525T112121_B01_60m.jp2
-----
Origen Imagen 1= (600000.0, 4900020.0)
Tamaño pixel = (60.0, -60.0)
('Numero de bandas ', 1, '\n')
Tamaño imagen = (1830, 1830)
('Tipo de dato', 'UInt16')
IMAGENES SENTINEL
Todas Bandas Sentinel

  IMAGEN /data/images/L1C/S2A_MSIL2A_20170525T112121_N0205_R037_T29TPJ_20170525T112434.SAFE/GRANULE/L2A_
T29TPJ_A010044_20170525T112434/IMG_DATA/R10m/L2A_T29TPJ_20170525T112121_B03_10m.jp2
-----
Origen Imagen 1= (600000.0, 4900020.0)
Tamaño pixel = (10.0, -10.0)
('Numero de bandas ', 1, '\n')
Tamaño imagen = (10980, 10980)
('Tipo de dato', 'UInt16')
IMAGENES SENTINEL
Todas Bandas Sentinel

  IMAGEN /data/images/L1C/S2A_MSIL2A_20170525T112121_N0205_R037_T29TPJ_20170525T112434.SAFE/GRANULE/L2A_
T29TPJ_A010044_20170525T112434/IMG_DATA/R10m/L2A_T29TPJ_20170525T112121_B08_10m.jp2
-----
Origen Imagen 1= (600000.0, 4900020.0)
Tamaño pixel = (10.0, -10.0)
('Numero de bandas ', 1, '\n')
Tamaño imagen = (10980, 10980)
('Tipo de dato', 'UInt16')
Done!
El preprocesado tardó 4971.24201512 segundos
```

En estas dos imágenes se observa el output devuelto por la consola , al ser lanzado el algoritmo de enmascaramiento nuboso Fmask a través de imágenes de un directorio concreto de trabajo.

## Anexo III. Archivo xml del modelo de entrenamiento de red neuronal

---

El siguiente código pertenece al archivo en formato xml originado mediante el entrenamiento de la red neuronal a través de la herramienta Train Images Classifier de Orfeo.

### Modelo de entrenamiento 1:

```
<?xml version="1.0"?>
<opencv_storage><opencv_ml_ann_mlp><format>3</format><layer_sizes> 8 2
2</layer_sizes><activation_function>SIGMOID_SYM</activation_function><f_param1>8.0000001
192092896e-01</f_param1><f_param2>1.</f_param2><min_val>-9.4999999999999996e-
01</min_val><max_val>9.4999999999999996e-01</max_val><min_val1>-9.7999999999999998e-
01</min_val1><max_val1>9.7999999999999998e-
01</max_val1><training_params><train_method>RPROP</train_method><dw0>1.0000000149011612e
-01</dw0><dw_plus>1.2000000000000000e+00</dw_plus><dw_minus>5.000000000000000e-
01</dw_minus><dw_min>1.0000000116860974e-
07</dw_min><dw_max>50.</dw_max><term_criteria><epsilon>9.999997473787516e-
05</epsilon><iterations>5000</iterations></term_criteria></training_params><input_scale
> 3.6761320232105273e-02 -2.8910307164870819e+00 2.6467742864380701e-02 -
1.8685992694160745e+00 2.1418289907021239e-02 -1.2064498122234364e+00
2.9666269988608947e-02 -2.2570383515916501e+00 3.3909781483303925e-02 -
2.5861033619797533e+00 1.4361608887871771e-02 -8.9410528659680200e-01
1.2124767910821991e-02 -9.0307210886720213e-01 1.2795083767206915e-02 -
8.9673572357100007e-01</input_scale><output_scale> 1. 0. 1.
0.</output_scale><inv_output_scale> 1. 0. 1. 0.</inv_output_scale><weights><_>
4.6893208580987672e-01 -5.1274253458301811e-01 7.1478907211195775e-01 -
6.4558750234246098e-01 -1.8092232156857851e-01 -5.8584587588572257e-01 -
3.9850127126042689e-01 9.8319795946863175e-01 1.0659708484032301e-01 -
1.1704721814497968e-01 4.8556304741057082e-01 -4.0426707752808289e-01
1.5407317896391670e+00 -1.2567389073776443e+00 6.9983366569055838e-01 -
1.2258123476728235e+00 7.3843950304960448e-01 -1.6840680587137400e+00</_><_>
4.7399560502235599e-01 -1.7339837735100365e+00 -3.9043858372008081e+01
3.8440730369146827e+01 6.8720195473234946e-01 -
1.1465354430091164e+00</_></weights><class_labels_type id="opencv-
matrix"><rows>1</rows><cols>2</cols>
<dt>i</dt>
<data> 0 1</data></class_labels></opencv_ml_ann_mlp></opencv_storage>
```



## Modelo de entrenamiento 2:

```
<?xml version="1.0"?>
<opencv_storage><opencv_ml_ann_mlp><format>3</format><layer_sizes> 4 3
2</layer_sizes><activation_function>SIGMOID_SYM</activation_function><f_param1>1.</f_par
am1><f_param2>1.</f_param2><min_val>-9.4999999999999996e-
01</min_val><max_val>9.4999999999999996e-01</max_val><min_val1>-9.7999999999999998e-
01</min_val1><max_val1>9.7999999999999998e-
01</max_val1><training_params><train_method>RPROP</train_method><dw0>1.0000000149011612e
-01</dw0><dw_plus>1.2000000000000000e+00</dw_plus><dw_minus>5.000000000000000e-
01</dw_minus><dw_min>1.0000000116860974e-
07</dw_min><dw_max>50.</dw_max><term_criteria><epsilon>1.0000000474974513e-
03</epsilon><iterations>5000</iterations></term_criteria></training_params><input_scale
> 2.2358134086418392e-02 -2.2385455994510344e+00 1.5961376767979818e-02 -
1.3657201758663893e+00 2.0026097571604666e-02 -1.9880783286949459e+00
1.3963943520437078e-02 -1.1062999959388820e+00</input_scale><output_scale> 1. 0. 1.
0.</output_scale><inv_output_scale> 1. 0. 1. 0.</inv_output_scale><weights><_> -
2.0625616779739775e-01 -3.7511158395442301e-01 4.7854078730528327e-01 -
4.0991327242801168e-01 2.8333469937485356e-01 5.0300539874005679e-01 -
3.5631969476640546e-02 -5.2666841742261605e-02 -1.9449001711376607e-01 -
2.7040304377862339e+00 -2.5703440674784210e+00 2.0590853304827230e+00 -
1.6880459379978063e+00 -8.8785066613022323e-01 6.1274201470698142e-01</_><_> -
1.7766766467075687e+01 1.6752242130376683e+01 -2.8884004584469860e+00
3.8551964020007183e+00 6.5517425225401471e-01 -6.5907992091728951e-01
4.6114110601661157e-01 -5.8599180597047729e-01</_></weights><class_labels
type_id="opencv-matrix"><rows>1</rows><cols>2</cols>
<dt>i</dt>
<data> 0 1</data></class_labels></opencv_ml_ann_mlp></opencv_storage>
```

Se puede observar en la tercera línea, el tipo de función de activación, en este caso sigmoide.

Los números que figuran entre las líneas 16 a la 23, son los pesos finales obtenidos por el modelo una vez ha finalizado el entrenamiento. Este quiere decir que esos pesos son los más idóneos para ser utilizados por la red neuronal a la hora de enmascarar las imágenes.

## Anexo IV. Código Orfeo

---

Ejecución de las herramientas Train Images Classifier e Images Classifier en el lenguaje Python 2.

El siguiente comando, es ejecutado a través de la consola de Windows para lanzar la herramienta Train Images Classifier.

### Train Images Classifier:

```
otbcli_TrainImagesClassifier -io.il prueba_1.tif -io.vd VectorData_QB.shp -io.imstat EstimateImageStatistics.xml -sample.mv 75000 -sample.mt 50000 -sample.vtr 0.5 -sample.vfn Class -classifier ann -classifier.ann.k sigmoidal -classifier.ann.c 1 -classifier.ann.opt false -io.out annQB1.txt -io.confmatout svmConfusionMatrixQB1.csv
```

El siguiente código es el perteneciente a la herramienta Train Image Classifier, escrito en Python 2.

```
#!/usr/bin/python

# Import the otb applications package
import otbApplication

# The following line creates an instance of the TrainImagesClassifier application
TrainImagesClassifier =
otbApplication.Registry.CreateApplication("TrainImagesClassifier")

# The following lines set all the application parameters:
TrainImagesClassifier.SetParameterStringList("io.il", ['prueba_1.tif'])

TrainImagesClassifier.SetParameterStringList("io.vd", ['VectorData_QB.shp'])

TrainImagesClassifier.SetParameterString("io.imstat", "EstimateImageStatistics.xml")

TrainImagesClassifier.SetParameterInt("sample.mv", 75000)

TrainImagesClassifier.SetParameterInt("sample.mt", 50000)

TrainImagesClassifier.SetParameterFloat("sample.vtr", 0.5)

# The following line execute the application
TrainImagesClassifier.ExecuteAndWriteOutput()
```

Caso similar el siguiente comando como código, con la diferencia de que en este caso se ejecuta la herramienta Image Classifier.

Image Classifier:

```
otbcli_ImageClassifier -in prueba_1.tif -imstat EstimateImageStatistics.xml -model  
clsvmModelQB1.svm -out prueba_1mask.tif
```

```
#!/usr/bin/python
```

```
# Import the otb applications package
```

```
import otbApplication
```

```
# The following line creates an instance of the ImageClassifier application
```

```
ImageClassifier = otbApplication.Registry.CreateApplication("ImageClassifier")
```

```
# The following lines set all the application parameters:
```

```
ImageClassifier.SetParameterString("in", "prueba_1.tif")
```

```
ImageClassifier.SetParameterString("imstat", "EstimateImageStatistics.xml")
```

```
ImageClassifier.SetParameterString("model", "clsvmModelQB1.svm")
```

```
ImageClassifier.SetParameterString("out", "prueba_1mask.tif")
```

```
# The following line execute the application
```

```
ImageClassifier.ExecuteAndWriteOutput()
```

## **Anexo V. Relación de imágenes utilizadas**

---

### **IDs imágenes WorldView-2 (Digital Globe 2020)**

10300100941E6900.tif



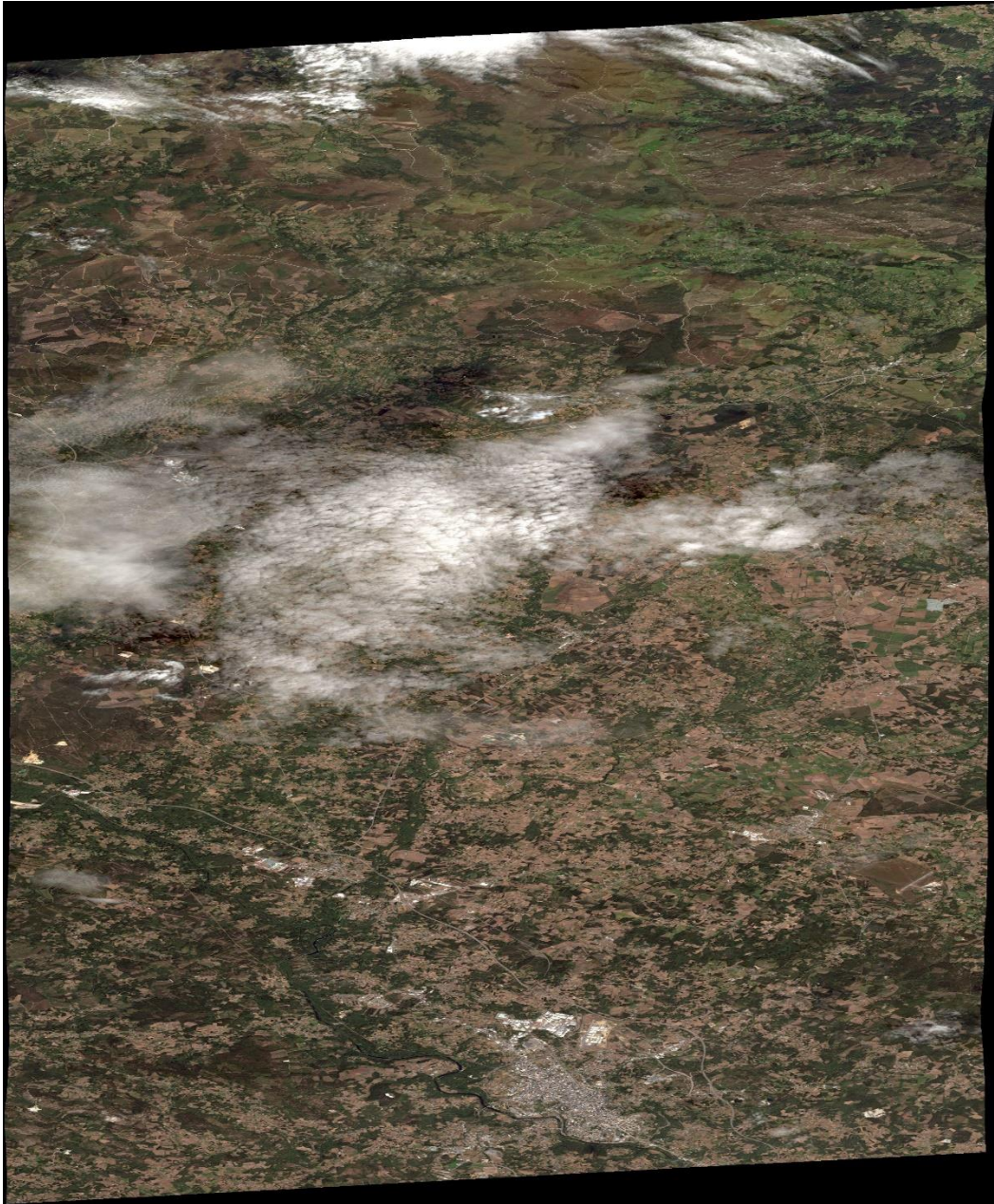
---

*Montaño García Francisco Javier  
Grado en física  
Universidad de Valladolid*

1030010093597200.tif



103001009AB2EB00.tif



103001009195F100.tif

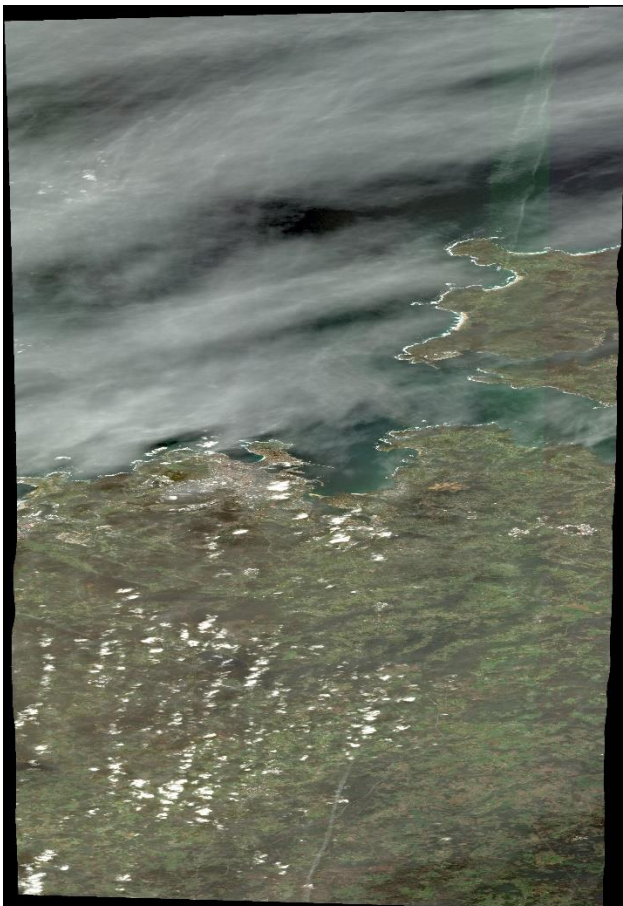


**IDs imágenes Quickbird (Digital Globe 2020)**

105001000EB6AF00.tif



1050010013641800.tif





*Algoritmo de detección de nubes a partir de redes neuronales perceptrón multicapa en imágenes satelitales.*

105001001A03EB00.tif



105001000E6DC000.tif



**IDs imágenes Sentinel-2 (Copernicus 2020)**

L2A\_T29PJ\_20170525T11221.jp2

L2A\_T29PJ\_20170525T112434.jp2