



# VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL



TRABAJO FIN DE MÁSTER:

***IDENTIFICACIÓN DE AEROGENERADORES MEDIANTE  
COMPARATIVA DE IMÁGENES PARA GENERAR  
ESPACIOS AÉREOS SEGUROS.***

Autora: Natalia Martín Jiménez

Tutores: Diego González Aguilera, Mario Soilán Rodríguez

septiembre 2021



## ÍNDICE

<b>AGRADECIMIENTOS</b> .....	4
<b>RESUMEN</b> .....	5
<b>ÍNDICE DE FIGURAS</b> .....	6
<b>INDICE DE TABLAS</b> .....	7
<b>INTRODUCCIÓN</b> .....	8
<b>1. OBJETIVOS</b> .....	11
<b>2. ESTADO DEL ARTE</b> .....	12
2.1. ORIGEN DEL DEEP LEARNING:.....	12
2.2. CONCEPTOS GENERALES Y APLICACIÓN DEL DEEP LEARNING.....	14
2.3. CLASIFICACION DE IMÁGENES DE SATÉLITE CON DEEP LEARNING.....	18
<b>3. MATERIAL EMPLEADO</b> .....	20
HARDWARE.....	20
SOFTWARE.....	20
<b>4. NORMATIVAS AESA Y OACI</b> .....	20
4.1. AESA.....	20
4.2. OACI.....	21
<b>5. METODOLOGÍA</b> .....	24
5.1. METODOLOGÍA GENERAL.....	24
5.1.1. CREACIÓN DE UNA RED NEURONAL.....	24
5.1.2. ENTRENAMIENTO DE LA RED NEURONAL.....	25
5.1.3. DETECCIÓN OBJETOS.....	25
5.2. METODOLOGÍA CON ARCGIS PRO.....	26
5.2.1. UTILIZACIÓN DE LA HERRAMIENTA DE CLASIFICACIÓN DE IMÁGENES.....	26
5.2.2. CREACIÓN DEL MODELO DE APRENDIZAJE PROFUNDO.....	32
5.2.3. DETECCIÓN DE OBJETOS UTILIZANDO DEEP LEARNING.....	34
<b>6. RESULTADOS</b> .....	37
<b>7. CONCLUSIONES</b> .....	52
<b>GLOSARIO DE TÉRMINOS</b> .....	55
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	56
<b>REFERENCIAS DE IMÁGENES</b> .....	57
<b>BIBLIOGRAFÍA</b> .....	58
<b>ANEXOS</b> .....	61



## AGRADECIMIENTOS

*En primer lugar, a mis padres, sin su ayuda no estaría redactando estas palabras.*

*A Sergio por su apoyo incondicional.*

*A mis compañeros, Jorge y Elena, gracias por la ayuda mostrada durante esta época de estudio.*

*A los compañeros de mi padre del CECAF, por darme la idea para este trabajo.*

*A Diego y Mario, por haberme guiado para que este TFM pueda salir adelante.*

## RESUMEN

Los obstáculos que se encuentran los pilotos de aviones, helicópteros, drones, etc. es una problemática que se produce con bastante frecuencia. En la actualidad existen una serie de normas y decretos que establecen como se deben señalar los obstáculos terrestres que tienen repercusión en la aviación, sin embargo, no existen datos sobre el papel de todos y cada uno de esos objetos que los pilotos pueden encontrarse a la hora de realizar un vuelo.

En este trabajo se pretende detectar objetos mediante la herramienta Deep Learning de ArcGIS Pro. En concreto la búsqueda se centra en la localización de aerogeneradores en el territorio de la Península Ibérica para generar espacios aéreos seguros.

Para ello hay que seleccionar un número determinado de imágenes que incluyan estos patrones y generar modelos de entrenamiento que detecten los objetos señalados.

Con los objetos ya detectados se estudiará el porcentaje de localización y en una zona concreta se compararán los resultados con los elementos que ya están registrados.

Por último, se darán varias soluciones de aplicación de estos datos obtenidos para lograr que los resultados sean de gran utilidad.

Para la detección de los aerogeneradores se hará uso del centro de descargas del CNIG, en concreto de ortofotos del PNOA de máxima actualidad del MTN50 (Mapa Topográfico Nacional 1:50.000).

## ÍNDICE DE FIGURAS

Figura 1: Calibración en vuelo.....	8
Figura 2: Obstáculos en aeronavegación. ....	9
Figura 3: Zona del territorio que contiene aerogeneradores representados. ....	9
Figura 4: Figura 3 ampliada. ....	10
Figura 5: Zona con aerogeneradores sin representar en las cartas de navegación aérea. ....	10
Figura 6: Línea temporal Deep Learning. Fuente: .....	13
Figura 7: Esquema IA y ML. ....	13
Figura 8: Composición Deep Learning. ....	15
Figura 9: Algoritmo de Regresión. ....	15
Figura 10: Algoritmo basado en Instancias. ....	16
Figura 11: Algoritmo de Árbol de Decisión.....	16
Figura 12: Algoritmo Bayesiano. ....	16
Figura 13: Algoritmo de Clustering. ....	17
Figura 14: Algoritmo Redes Neuronales. ....	17
Figura 15: Algoritmo de Reducción de Dimensión. ....	17
Figura 16: Librerías instaladas en ArcGIS Pro. Elaboración propia. ....	18
Figura 17: algunas soluciones del proyecto. ....	19
Figura 18: detección de aerogeneradores.....	19
Figura 19: ejemplo localización aerogeneradores. Elaboración propia. ....	22
Figura 20: ejemplo aerogeneradores sin registrar. Elaboración propia. ....	23
Figura 21: ejemplo espacio sin aerogeneradores. Elaboración propia.....	23
Figura 22: Esquema Red Neuronal.....	24
Figura 23: herramienta de clasificación. Elaboración propia. ....	26
Figura 24: Zona de España con la capa de aerogeneradores geolocalizados. Elaboración propia. ....	26
Figura 25: Ventana de clasificación de imagen. Elaboración propia. ....	27
Figura 26: Ejemplo Clasificación.....	27
Figura 27: Ejemplo Clasificación.....	27
Figura 28: Ejemplo Clasificación.....	27
Figura 29: Exportación de datos de formación. Elaboración propia. ....	28
Figura 30: Funcionamiento RCNN. ....	30
Figura 31: Funcionamiento Faster RCNN. ....	30
Figura 32: Funcionamiento Mask RCNN.....	30
Figura 33: Exportación datos de formación. Elaboración propia. ....	31
Figura 34: Zoom a un elemento perteneciente a la capa de las etiquetas creadas. Elaboración propia. .	32
Figura 35: Creación Modelo de entrenamiento profundo. Elaboración propia. ....	33
Figura 36: Carpeta de salida del modelo. Elaboración propia. ....	34
Figura 37: Parámetros de Detect Objects- Elaboración propia. ....	34
Figura 38: Argumentos para detección de objetos. Elaboración propia. ....	35
Figura 39: Entornos del Detect Objects. Elaboración propia. ....	35
Figura 40: Resultados obtenidos en la fase de experimentación. Elaboración Propia. ....	36
Figura 41: conjunto de imágenes de muestra. Elaboración propia. ....	37
Figura 42: Model Characteristics. Elaboración propia. ....	38
Figura 43: Ejemplo Model Characteristics. ....	39
Figura 44: Resultados iniciales hoja 0001 con el modelo de 100 imágenes. Elaboración propia. ....	39
Figura 45: Primeros resultados hoja 0024 con el modelo de 100 imágenes. Elaboración propia.....	40
Figura 46: Primeros resultados hoja 0980 con el modelo de 100 imágenes. Elaboración propia.....	40
Figura 47: Primeros resultados hoja 0958 con el modelo de 100 imágenes. Elaboración propia.....	41
Figura 48: Imagen 47 ampliada. Elaboración propia.....	41
Figura 49: Ejemplo de zona detectada con un falso positivo. Elaboración propia. ....	42
Figura 50: ejemplo de zonas desplazadas. Elaboración propia. ....	43
Figura 51: exportación de datos de formación. Elaboración propia.....	44
Figura 52: etiquetas para el modelo de entrenamiento. Elaboración propia. ....	44
Figura 53: informe de la detección de objetos por aprendizaje profundo. Elaboración propia. ....	45
Figura 54: localización del parque eólico más grande de España. Elaboración propia. ....	46
Figura 55: localización del parque eólico en las hojas del MTN50. Elaboración propia. ....	46
Figura 56: división de la hoja 0958 por cuadrantes. Elaboración propia. ....	47

Figura 57: división de la hoja 0980 por cuadrantes. Elaboración propia. ....	47
Figura 58: elementos sin localizar. Elaboración propia. ....	48
Figura 59: elementos din localizar tras ralizar un segundo intento. Elaboración propia. ....	48
Figura 60: Ejemplo de falsos positivos en la deteccion de objetos. Elaboración propia. ....	49
Figura 61: resultados de búsqueda en el cuadrante 3 de la hoja 0980. Elaboración propia. ....	50
Figura 62: Atributos de la capa generada. Elaboración propia. ....	51
Figura 63: informe del proceso. Elaboración propia. ....	51
Figura 64: contenido de la carptena generada. Elaboración propia. ....	52
Figura 65: Arquitectura ResNet-34. ....	55

## INDICE DE TABLAS

Tabla 1: Resultados obtenidos. Elaboración propia. ....	50
---	----

## INTRODUCCIÓN

En el mundo de la aviación existen múltiples tareas y trabajos que conllevan la realización de vuelos a bajas alturas. Un ejemplo de esto pueden ser las calibraciones aéreas.

Estas ayudas en vuelo son obligatorias para todos los aeropuertos civiles y militares y tienen como objetivo la calibración de los sistemas de radio ayuda a la navegación y las comunicaciones con el sistema de control en tierra. En la [figura 1](#) se puede contemplar la máquina que se utiliza para realizar las calibraciones en vuelo de los aeropuertos. Vemos que el aparato utiliza las cartas de navegación aéreas como referencia para guiarse por el espacio aéreo.



*Figura 1: Calibración en vuelo. Referencia: Artículo De Revista Mapping. La inspección en vuelo en el CECAF. Fuente: [http://revistamapping.com/wp-content/uploads/2018/03/Revista-MAPPING-185\\_A1.pdf](http://revistamapping.com/wp-content/uploads/2018/03/Revista-MAPPING-185_A1.pdf)*

Este tipo de vuelos suelen producirse a bajas alturas, no sobre los 10.000 pies de los vuelos comerciales. Debido a esto, los pilotos suelen encontrarse con obstáculos durante el vuelo, uno de estos obstáculos son los aerogeneradores.

Estos elementos, deberían estar debidamente georreferenciados y señalados en las cartas de navegación aérea para poder ser detectados y que los pilotos tengan constancia de su existencia.

A continuación, se muestra un ejemplo para entender la necesidad de realizar este proyecto.

En la página oficial del Ministerio del Interior, en el apartado de Protección Civil y Emergencias encontramos un documento en el que se muestran las variantes de representación de los aerogeneradores (obstáculos) en función de sus características y que deben representarse en las cartas o mapas de navegación aérea.

Obstáculos			
Obstáculo y grupo de obstáculos / Obstacle and group of obstacles		Obstáculo y grupo de obstáculos iluminado / Lighted obstacle and group of obstacles	
Obstáculo y grupo de obstáculos de más de 100 m / Obstacle and group of obstacles higher than 100 m		Obstáculo determinante Determining obstacle	
Elevación de la cota / Spot elevation	1755	Elevación obstáculo / Obstacle elevation	998
Turbina eólica no iluminada e iluminada / Not illuminated and illuminated wind turbine			

Figura 2: Obstáculos en aeronavegación. Fuente: [Proteccioncivil.es](http://Proteccioncivil.es) y <https://www.proteccioncivil.es/catalogo/carpeta02/carpeta24/vademecum17/vade01.htm>

En la [figura 2](#) vemos que los aerogeneradores deberían representarse con esos símbolos en las cartas aéreas de navegación. Hemos comprobado si esto se cumple. Para realizar esta comprobación, en la página oficial del Ministerio de Transporte, Movilidad y Agenda Urbana existe un visor que muestra los diferentes mapas aeronáuticos que contiene el territorio de la Península Ibérica. En este visor llamado [INSIGNIA](#), existen varias capas de información, una de ellas es la de obstáculos, que es la que nos interesa comprobar.

En la [figura 3](#) se muestra una zona que contiene aerogeneradores y que están representados con la simbología correcta. Haciendo zoom a esa zona ([Figura 4](#)), se comprueba visualmente que cada símbolo coincide con un objeto. Esta zona mostrada corresponde con el parque eólico Monte gordo, situado en Villablanca, Huelva.

Sin embargo, nos hemos dirigido a otra zona de la que tenemos constancia que existen aerogeneradores y éstos no están representados en el mapa. ([Figura 5](#)). Esta zona pertenece al complejo eólico De Andévalo, en Huelva.

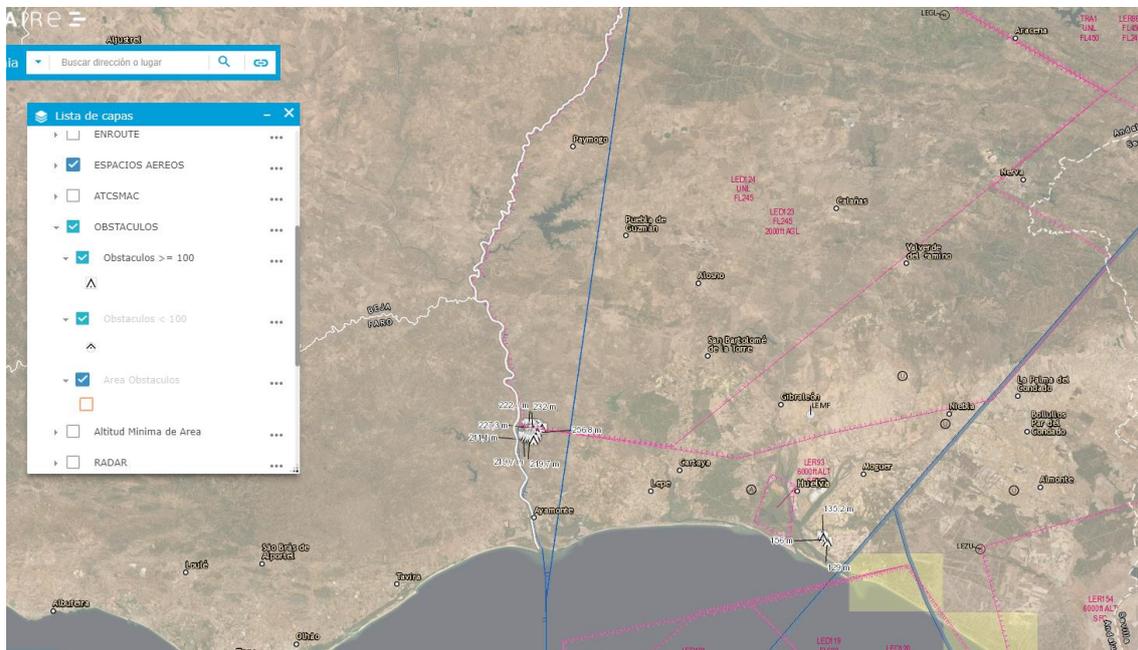


Figura 3: Zona del territorio que contiene aerogeneradores representados. Fuente: <https://insignia.enaire.es/>



Figura 4: Figura 3 ampliada. Fuente: <https://insignia.enaire.es/>

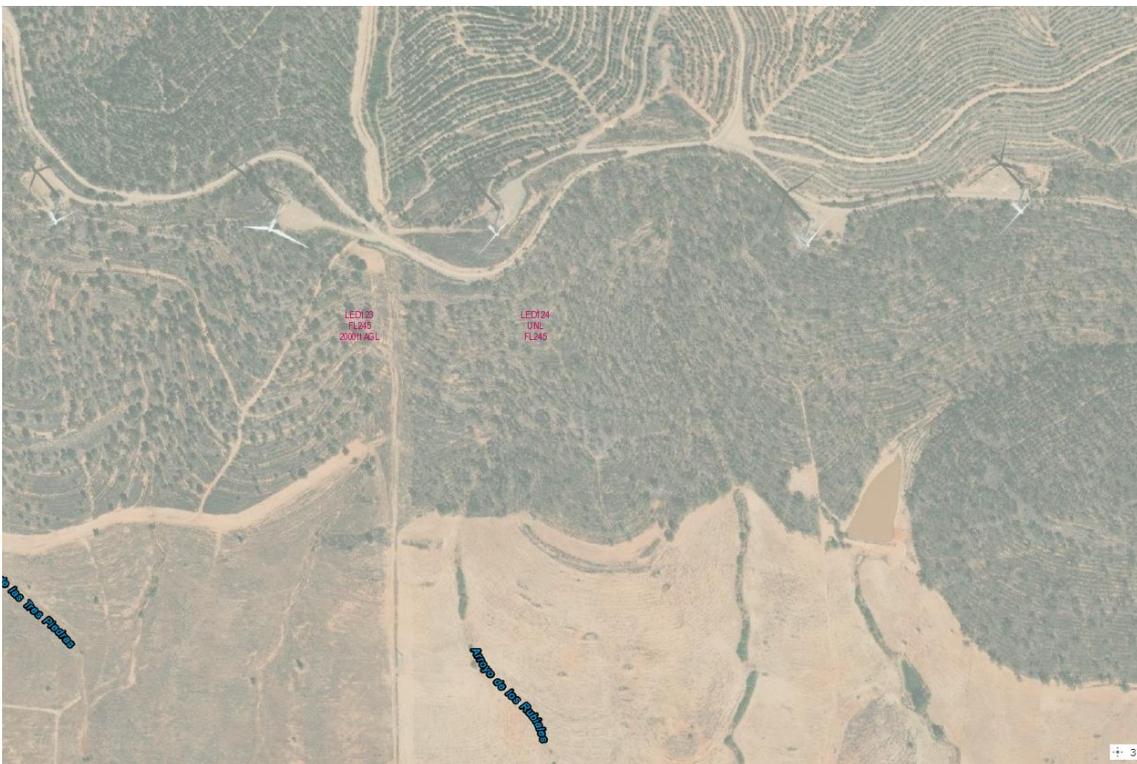


Figura 5: Zona con aerogeneradores sin representar en las cartas de navegación aérea. Fuente: <https://insignia.enaire.es/>

## 1. OBJETIVOS

El objetivo principal que se pretende alcanzar en este trabajo es el de localizar los aerogeneradores existentes en el territorio nacional. Esta localización se llevará a cabo gracias a la creación de un modelo de entrenamiento basado en el aprendizaje profundo que sea capaz de detectar objetos en función de las etiquetas que se hayan establecido y el objeto a perseguir, que en este caso se trata aerogeneradores.

Como objetivos secundarios, pero no menos importantes, tenemos: familiarizarnos con el uso de la aplicación ArcGIS Pro y en especial con la herramienta Deep Learning, entender el funcionamiento interno de esta última, ver qué opciones ofrece y en qué se diferencian. Conocer la normativa en la que se va a basar este trabajo, consultando los decretos de AESA (Agencia Estatal de Seguridad Aérea) y la normativa OACI (Organización de Aviación Civil Internacional).

Como objetivo final tenemos el de establecer la aplicación de los resultados obtenidos. Cómo pueden ser más eficientes y a qué áreas u organismos les puede resultar más útil la localización de estos objetos.

## 2. ESTADO DEL ARTE

Se ha dividido el estado del arte de este trabajo en tres partes. La primera de ellas trata sobre los avances que se han ido originando acerca del método de entrenamiento Deep Learning, como fue su origen y en qué campos se empezó a utilizar. La segunda parte explica los conceptos generales de este método y qué tipos de clasificación se pueden realizar. Por último, se expone un pequeño resumen de una aplicación específica de la detección de elementos por el método del aprendizaje profundo.

### 2.1. ORIGEN DEL DEEP LEARNING:

Este concepto tuvo su origen en los años 40 y 50 del siglo XX. Alan Turing, matemático e informático, entre otras profesiones, crea el "Test de Turing". Este test establecía un método para detectar si una máquina podía ser "inteligente". En la última etapa de los años 50, se establecieron las primeras redes neuronales artificiales, se puede decir que en esta época aparece por primera vez el concepto de Inteligencia Artificial. [\[1\]](#)

Durante los siguientes 20 años, este tipo de ideas sufrieron un corte de financiación e investigación. No obstante, se producen dos grandes avances en este campo. En el 1967, se describe el algoritmo de "Nearest Neighbor" considerándose con esto, el origen del reconocimiento por patrones. En 1979 un grupo de estudiantes de la Universidad de Stanford, logran que un robot sea capaz de desplazarse por una habitación reconociendo obstáculos y evitándolos.

En la siguiente década, la de los 80, de nuevo se produce un empujón en este campo gracias al avance de la tecnología. Se introduce por primera vez el concepto de Learning, en concreto "Explanation Based Learning". Este método consistía en el análisis de datos de entrenamiento, de tal manera que seleccionaba los importantes y desechara el resto.

Nuevamente, en la década de los 90 y durante los primeros años del nuevo siglo, ocurrió un nuevo "bloqueo" en cuanto a la investigación en el campo de la inteligencia artificial. La reputación de estos estudios tardó en recuperarse. Sin embargo se pueden destacar dos momentos claves en este periodo.

1990s: de comienza a estudiar el Machine Learning desde un punto de vista ligado al dato y no al conocimiento, que era como se venía haciendo en los últimos años. Es por esto, que los profesionales del campo, comienzan a generar programas que son capaces de analizar multitud de datos y además extraer conclusiones y resultados.

1997: por primera vez, un ordenador vence al campeón del mundo de ajedrez Gary Kasparov gracias a un método de entrenamiento basado en el Machine Learning.

En los últimos 20 años, concretamente desde el año 2006, se ha producido un gran avance tecnológico en todos los ámbitos. Los ordenadores son más potentes y las capacidades de almacenamiento de datos se han ampliado exponencialmente. Todo esto ha facilitado que el campo del Machine Learning se haya vuelto a estudiar. Las grandes empresas muestran un gran interés por esta forma de trabajar y procesar los datos, por esta razón, están invirtiendo gran capital para que se pueda avanzar y seguir innovando.

Como datos claves de estos casi 20 años tenemos:

2006: se establece finalmente el término "Deep Learning" gracias a Geoffrey Hinton, un informático británico.

2012: Jeff Dean (Google) y Andrew Ng (Universidad de Stanford) se unen para crear y liderar en proyecto GoogleBrain. El objetivo de éste es desarrollar una Red Neuronal Profunda que detecte patrones en imágenes y videos.

2012: de nuevo Geoffrey Hinton entra en esta lista temporal para liderar un grupo que resulta ser ganador del concurso de Visión por Computador utilizando una Red Neuronal Profunda.

2014: en este año Facebook se lanza a la inversión de este campo y crea DeepFace. Esta herramienta es un algoritmo basado en RNPs (Red Neuronal Profunda) y es capaz de reconocer a personas.

En la última década, empresas y marcas como Google, Amazon, Microsoft entre otras, invierten tiempo y dinero en la investigación en el campo de la inteligencia artificial y específicamente en el área del aprendizaje profundo, creando o comprando aplicaciones como DeepMind, Machine Learning de Amazon, Distributed Machine Learning Toolkit...

En la figura 6 se muestra un resumen de la línea temporal del Machine Learning.

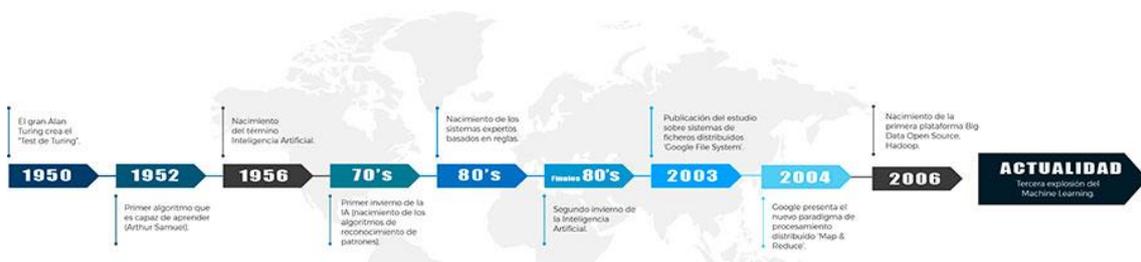


Figura 6: Línea temporal Deep Learning. Fuente: <https://www.futurespace.es/machine-learning-los-origenes-y-la-evolucion/>

En la figura 7 podemos ver la relación que existe entre la Inteligencia Artificial y el Machine Learning.

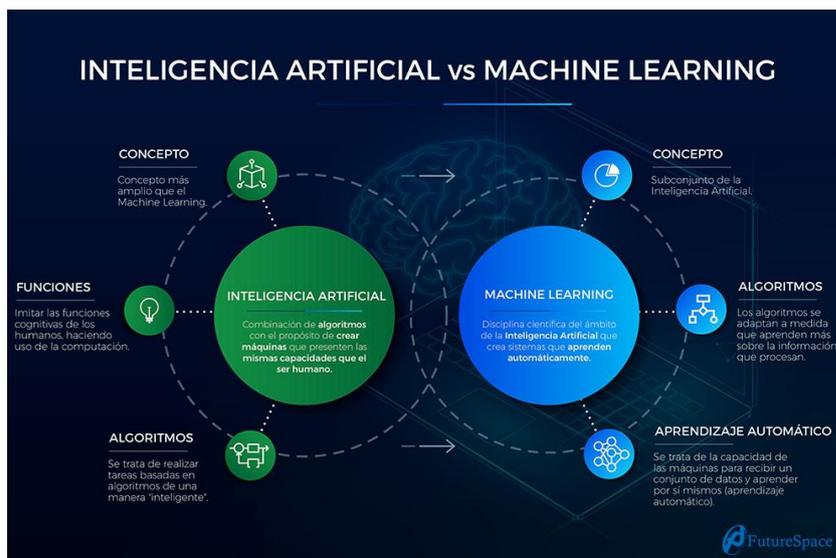


Figura 7: Esquema IA y ML. Fuente: <https://www.futurespace.es/machine-learning-los-origenes-y-la-evolucion/>

A lo largo de estos años de evolución del Deep Learning, los campos en los que se ha utilizado este tipo de Inteligencia Artificial son de lo más variado. A continuación, vamos a explicar alguno de ellos.

Traducción automática: la traducción automática de textos existe desde hace tiempo, aplicaciones como Google Translate o DeepL Translate llevan años funcionando, sin embargo, gracias a este avance de redes neuronales y al aprendizaje profundo se están logrando resultados más precisos.

Asistencia sanitaria: en el mundo de la medicina se está empezando a utilizar el Deep Learning como ayuda para dar diagnósticos tempranos, detectar enfermedades mortales, como pueden ser los tumores, mitigando así los riesgos de los pacientes y reduciendo el tiempo de diagnóstico y el coste. Un ejemplo concreto del Deep Learning en el campo de la medicina, es la detección de la metástasis en el cáncer de tiroides [2].

Detección del fraude: son muchos los delitos y fraudes que se cometen a diario a través de transacciones bancarias. El Deep Learning es capaz de identificar patrones en este tipo de movimientos e identificar comportamientos anómalos o sospechosos. [3]

Vehículos autónomos: cuando se ve una película del futuro, siempre incluyen coches que conducen solos. Lo que antes parecía imposible, ya se está consiguiendo poco a poco. Ya hemos visto que existen automóviles que aparcan solos, y otros que detectan obstáculos y son capaces de frenar antes de que lo haga el conductor. No obstante, existe una preocupación de los desarrolladores de estas máquinas, y es que, en la conducción influyen múltiples variables y escenarios desconocidos, por lo que lograr una conducción segura al 100% puede llevar algo más de tiempo. Lo que sí es cierto, es que, gracias a las cámaras, los sensores de movimiento, los mapas geográficos, etc, las máquinas son capaces de lograr una navegación libre de tráfico, identificar rutas más cortas, desviarte de carreteras de pago, y todo ello a tiempo real para que el usuario pueda reaccionar a las adversidades de la mejor forma posible. [4]

Reconocimiento visual: el Deep Learning sirve también para reconocer patrones detectados en imágenes [5], por ejemplo, el sistema de desbloqueo de los móviles funciona con este algoritmo.

En este proyecto se va a utilizar el aprendizaje profundo (Deep Learning) para detectar una serie de imágenes establecidas.

## 2.2. CONCEPTOS GENERALES Y APLICACIÓN DEL DEEP LEARNING

Como se ha explicado en el apartado anterior, el aprendizaje profundo es una evolución de la Inteligencia Artificial y posteriormente del Machine Learning. Utiliza varias capas de algoritmos en forma de redes neuronales. El modelo de aprendizaje tiene que entrenarse con diferentes imágenes del objeto a detectar, estas imágenes pasan por un proceso de diferentes capas y son analizadas, destacando los identificadores esenciales e importantes para poder lograr una clasificación óptima.

Para entender un poco más este tema, se muestra una imagen (figura 8) que resume el concepto que hemos explicado.

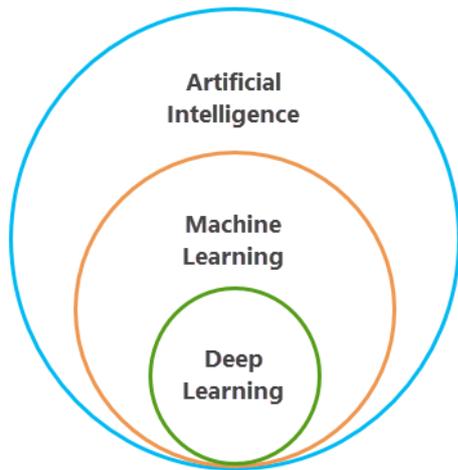


Figura 8: Composición Deep Learning. <https://pro.arcgis.com/es/pro-app/latest/help/analysis/deep-learning/what-is-deep-learning-.htm>

Todo este proceso funciona con diferentes algoritmos que vamos a nombrar y explicar brevemente.

Algoritmos de Regresión ([figura 9](#)): estos algoritmos están basados en establecer un método estadístico para relacionar un número de características con una variable continua. Para poder llevar esto a cabo, se recurre a la regresión lineal, que establece una recta para detectar la tendencia de un conjunto de datos. Los datos que se “alejen” de esa recta se podrán considerar erróneos y desecharlos. También es común que se utilicen algoritmos de Regresión Lógica.

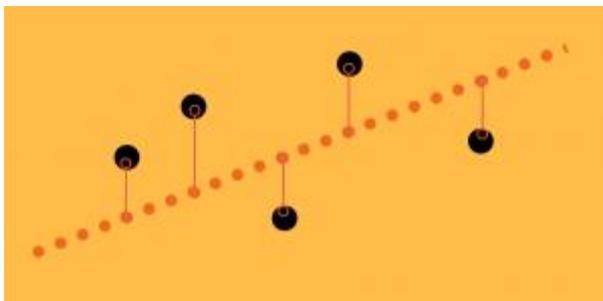


Figura 9: Algoritmo de Regresión. <https://www.aprendemachinelarning.com/principales-algoritmos-usados-en-machine-learning/>

Algoritmos basados en Instancias ([figura 10](#)): estos algoritmos se basan en muestras de entrenamiento que son requeridos por el modelo de aprendizaje. Necesitan partir de una base para poder realizar las comparativas necesarias y la detección de los objetos de cada trabajo. Se los conoce también por ser algoritmos que trabajan basándose en la memoria del programa, creando un modelo de entrenamiento, agregando nuevos datos y comparando las semejanzas con las muestras previamente señaladas. Los algoritmos basados en Instancias más usados son el K-Nearest Neighbor (KNN) y el Self-Organizing MAP.

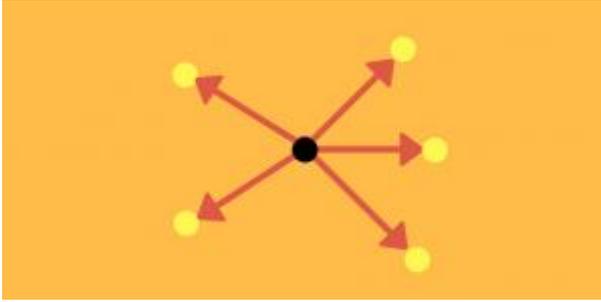


Figura 10: Algoritmo basado en Instancias. <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/>

Algoritmos de Árbol de Decisión (figura 11): se basan en valores reales de los atributos de la imagen o los datos.

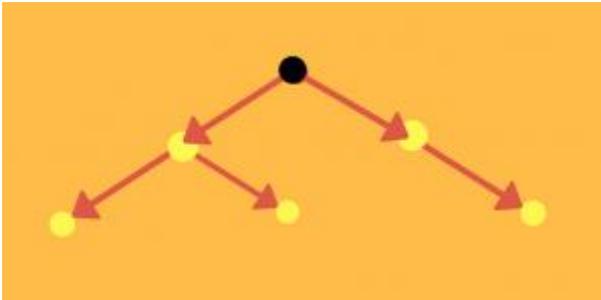


Figura 11: Algoritmo de Árbol de Decisión. <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/>

Algoritmos Bayesianos (figura 12): utilizan el Teorema de Bayes de probabilidad para buscar la hipótesis más fiable a partir de un conjunto de datos.

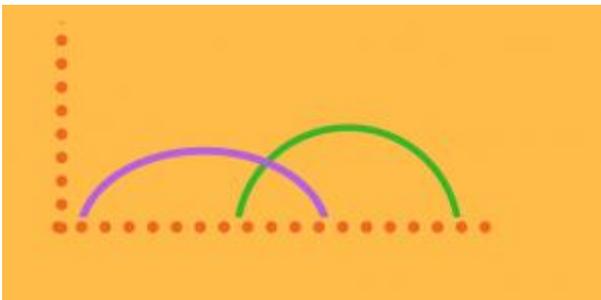


Figura 12: Algoritmo Bayesiano. <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/>

Algoritmos de Clustering (figura 13): están basados en la agrupación de datos ya existentes de los cuales no se conocen sus características comunes. Este algoritmo crea puntos en común “puntos centrales” y jerarquías para que se puedan crear grupos según sus diferencias o similitudes.

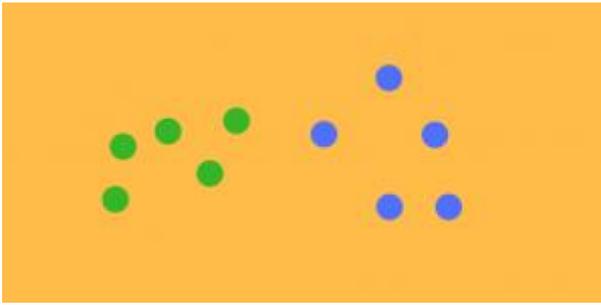


Figura 13: Algoritmo de Clustering. <https://www.aprendemachinelarning.com/principales-algoritmos-usados-en-machine-learning/>

Algoritmos de Redes Neuronales (figura 14): estos algoritmos están basados en otros que se han nombrado anteriormente, que son los de Clasificación y los de Regresión. La ventaja de este algoritmo con respecto a los nombrados, es que está inspirado en las funciones biológicas del ser humano, es decir, tratan de imitar el funcionamiento de las redes neuronales del cerebro humano. Para poder llevar a cabo la detección de patrones con este algoritmo, se necesita mucha capacidad de procesamiento y almacenamiento de datos, es por eso que, durante el siglo pasado, este tipo de tecnología no se pudo estudiar con más exactitud y es ahora cuando está despegando.

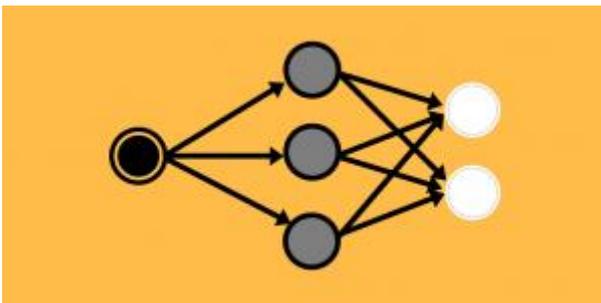


Figura 14: Algoritmo Redes Neuronales. <https://www.aprendemachinelarning.com/principales-algoritmos-usados-en-machine-learning/>

Algoritmo de Reducción de Dimensión (figura 15): estos algoritmos permiten simplificar o dividir el modelo que se quiere estudiar, de esta manera se simplifican los datos de una manera no supervisada, se crea el algoritmo y posteriormente se procede a la búsqueda supervisada utilizando ese algoritmo generado.

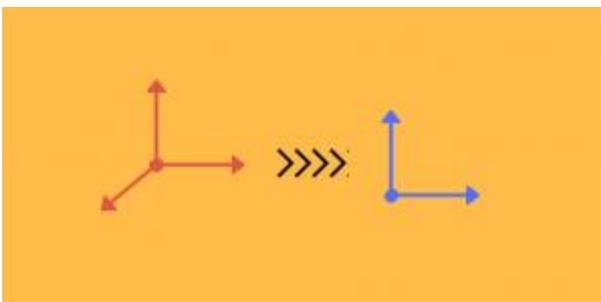


Figura 15: Algoritmo de Reducción de Dimensión. <https://www.aprendemachinelarning.com/principales-algoritmos-usados-en-machine-learning/>

Procesamiento del Lenguaje Natural “Natural Language Processing “(NLP): este algoritmo es un conjunto del Machine Learning, la Lingüística y el DataScience. El fin de éste es llegar a comprender el lenguaje humano. Con este tipo de algoritmos se consiguen aplicaciones de

traducción automática entre otras lenguas, asistentes de voz como Siri, y un sinnúmero de utilidades que están surgiendo en estos últimos tiempos.

**Algoritmos de Aprendizaje Profundo (Deep Learning):** este algoritmo es un método de Machine Learning que permite a la computadora realizar entrenamientos diversos para poder estimar una predicción y dar como resultado un conjunto de datos. Se necesitarán siempre 3 tipos de datos de Capas de Neuronas, que son:

- Las capas de entrada: son las imágenes “madre” en las cuales el algoritmo y el programa realizará el proceso de búsqueda.
- Las capas ocultas: son las imágenes en las que está basado el modelo de aprendizaje, cuantas más imágenes se hayan incluido para entrenar, más exacto o más referencias de búsqueda tendrá el algoritmo.
- La capa de salida: serán los datos que devuelve el algoritmo tras realizar la búsqueda completa.

### Referencia [6].

En este proyecto se va a llevar a cabo la búsqueda de aerogeneradores con este último algoritmo, el de aprendizaje profundo.

### 2.3. CLASIFICACION DE IMÁGENES DE SATÉLITE CON DEEP LEARNING

“-.Mark Pritt y Gary Chern “Satellite Image Classification with Deep Learning”, 2017.” [7]

En este trabajo se plantea resolver los problemas de detección de objetos o zonas geográficas de gran extensión. Estos problemas se ocasionan debido a la escasez de analistas, al gran tamaño de las zonas de búsqueda y a que las herramientas convencionales de detección y clasificación no devuelven el resultado lo suficientemente preciso.

Con el Deep Learning, se consigue una comprensión de las imágenes mediante unas redes neuronales convencionales. En este proyecto se describe un sistema de aprendizaje profundo para realizar la clasificación de objetos e instalaciones.

Las similitudes que encontramos con el presente proyecto son:

- La utilización del Deep Learning para la detección de objetos. Esta herramienta está implementada en Python y utiliza bibliotecas de aprendizaje profundo Keras y TensorFlow. En nuestro caso particular, la herramienta también está implementada en Python y utiliza las mismas librerías o bibliotecas. (figura 16)

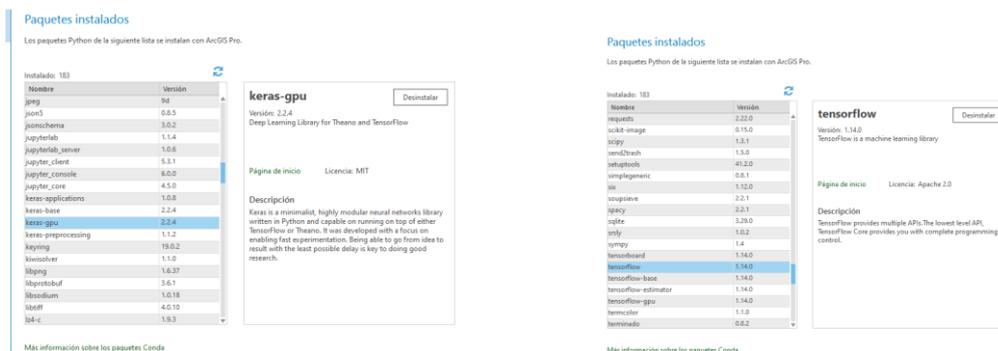


Figura 16: Librerías instaladas en ArcGIS Pro. Elaboración propia.

- El método de detección utilizado de aprendizaje está basado en un conjunto de CNNs con redes neuronales de procesamiento. (apartado 6.2.1. de este documento).
- La solución final de detección de objetos, edificaciones o terrenos. (Figura 17)



Figura 17: algunas soluciones del proyecto. Fuente: <https://ieeexplore.ieee.org/abstract/document/8457969/authors#authors>

Podemos observar también que uno de los ejemplos de soluciones obtenidas (figura 18), son aerogeneradores. En este caso, indica que la precisión de localización ha sido del 99%. Puede ser un dato interesante para comparar con el que se obtenga de este trabajo.



Figura 18: detección de aerogeneradores. Fuente: <https://ieeexplore.ieee.org/abstract/document/8457969/authors#authors>

### 3. MATERIAL EMPLEADO

#### HARDWARE

Para llevar a cabo este proyecto, como material se ha utilizado un ordenador marca OMEN by HP Laptop.

Tiene un procesador Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz y una RAM instalada de 16.0 GB (15.9 GB usable). Además, contiene dos tarjetas gráficas instaladas, la Intel UHD Graphics 630 y la NVIDIA GeForce GTX 1050. Ambas tarjetas facilitan a la máquina a la hora de llevar a cabo los procesamientos de imágenes. No obstante, como veremos más adelante, al tratarse en algunos casos de imágenes de gran tamaño, el proceso de búsqueda puede acarrear un tiempo elevado.

#### SOFTWARE

Se va a utilizar el Software de ArcGIS Pro. Este programa es una aplicación de escritorio creada por ESRI y que está destinada a trabajar con SIG (Sistemas de Información Geográficos). Contiene una gran cantidad de funciones destinadas a la visualización de datos, a realizar análisis, transformaciones, entre muchas otras. En este caso en particular vamos a utilizar la herramienta Imagine Analyst, que es la que contiene las funciones del Deep Learning.

ArcGIS Pro no es un software libre, se necesita licencia para poder trabajar con él. Gracias a la licencia de estudiante obtenida en este Máster se ha podido trabajar sin problema.

### 4. NORMATIVAS AESA Y OACI

#### 4.1. AESA

AESA es la Agencia Estatal de Seguridad Aérea, este organismo pertenece a la secretaria de estado de Transportes, Movilidad y Agencia Urbana. Esta agencia está destinada a verificar que se cumplan las normas de aviación civil en el conjunto de la aviación española.

Podemos destacar algunas de sus responsabilidades más importantes que son, la supervisión, inspección y ordenación del transporte aéreo, la navegación y la seguridad aérea. El objetivo que tienen marcado es el de aportar una seguridad, calidad y sostenibilidad en el campo de la aviación civil.

La AESA contiene multitud de normativas, decretos y leyes, en este caso vamos a estudiar y adentrarnos en la que habla específicamente de la normativa de los aerogeneradores. Este documento se encuentra disponible en la plataforma del Ministerio [8] y es de acceso público. Su nombre es "Señalamiento e Iluminación de Turbinas y Parques Eólicos".

Tras leer el documento, se va a exponer en este trabajo el apartado que explica la señalización de los Aerogeneradores

*"Serán susceptibles de señalamiento y/o iluminación aquellos parques eólicos que, por su condición, determine que es necesario en los siguientes casos:*

- *Parques eólicos que se encuentren dentro de las zonas afectadas por Servidumbres Aeronáuticas (Aeródromos, Radioeléctricas, y de Operación), independientemente de la altura de los elementos del mismo (aerogeneradores, antenas meteorológicas, grúas necesarias para su construcción).*

- *Parques eólicos cuyos elementos (incluidos los necesarios para la construcción) tengan una altura superior a 100 metros en todo el territorio nacional, sobre planicies o partes prominentes del terreno o nivel del mar dentro de aguas jurisdiccionales.*

Así mismo, la normativa especifica cuando se considera obstáculo un objeto:

*Todos los objetos fijos (ya sean temporales o permanentes) y móviles, o partes de los mismo que:*

- *Se encuentren en un área destinada al movimiento en superficie de aeronaves*
- *Sobresalgan de una superficie definida destinada a proteger a las aeronaves en vuelo*
- *Queden fuera de dichas superficies definidas y se haya considerado como un peligro para la navegación aérea.*

*En el apartado 5.1 de la normativa también se especifica la obligación de la comunicación a AESA a la hora de instalar nuevos aerogeneradores:*

- *Aerogeneradores que se encuentren dentro de las zonas afectadas por Servidumbres Aeronáuticas, independientemente de la altura del aerogenerador.*
- *Aerogeneradores fuera de las zonas afectadas por Servidumbres Aeronáuticas y cuya altura sea superior a los 100m”*

Este texto entrecomillado se ha obtenido de la Normativa AESA. [8]

#### 4.2. OACI

La OACI es un organismo especializado de las naciones unidas y sus siglas significan (Organización de Aviación Civil Internacional). Este organismo no dicta reglamentos mundiales, pero su objetivo es favorecer la cooperación en el transporte aéreo entre diferentes países.

Por este motivo, la OACI tiene publicados una serie de normas y recomendaciones que los países pueden adoptar como leyes en sus territorios. Así la navegación internacional puede ser similar en cada territorio. Por esta razón la norma explicada anteriormente de la AESA está basada en las recomendaciones de la OACI. [9]

Una vez explicadas y entendidas ambas normativas, entendemos entonces que los parques eólicos tienen la obligación de pedir permiso y avisar para una nueva instalación de aerogeneradores y que estos deben estar debidamente señalados. Esto seguramente se cumpla, sin embargo, sigue existiendo la problemática de no saber dónde se encuentran esos elementos, de no disponer de una base de datos completa y actualizada de cada obstáculo que se pueda encontrar un dispositivo aéreo.

Tras la lectura de las normativas, no se ha encontrado un decreto, norma o ley en la que se establezca que los aerogeneradores deben estar geolocalizados y que esa información se deba distribuir entre los diferentes organismos encargados de la aviación, como pueden ser AENA, las Fuerzas Armadas, empresas dedicadas a vuelos con drones...

Lo que sí existe es comunicación entre los citados organismos y los diferentes parques eólicos, es decir, AENA, por ejemplo, solicita la información a un parque eólico y éste le suministra una base de datos con coordenadas de los aerogeneradores y con información de ellos.

Sin embargo, esto puede resultar poco efectivo, ya que solo en la península ibérica hay aproximadamente unos 1200 parques eólicos. El simple hecho de preguntar uno por uno y de generar una única base de datos para todos, ya conllevaría demasiado tiempo.

Por esta razón surge la idea de localizar estos elementos a través de herramientas diseñadas para detectar patrones establecidos.

Gracias al visor INSIGNIA, hemos descargado la capa existente de obstáculos en formato .csv. Esta capa contiene los aerogeneradores localizados y sus coordenadas geodésicas. Se ha cargado en el software ArcGIS Pro y se ha insertado el servicio WMS de la ortofoto del PNOA de máxima actualidad para poder tener una primera visualización de la información de la que vamos a partir.

En la mayoría de las visualizaciones se aprecia que los aerogeneradores están correctamente situados, sin embargo, hay zonas en las que algunos de ellos no están registrados o viceversa, existen zonas sin aerogeneradores y que están señalizados.

Por ejemplo, en la [figura 19](#), los puntos violetas corresponden a la capa puntual del archivo mencionado anteriormente, que hace referencia a la localización de los aerogeneradores y se puede apreciar que los elementos están correctamente registrados.

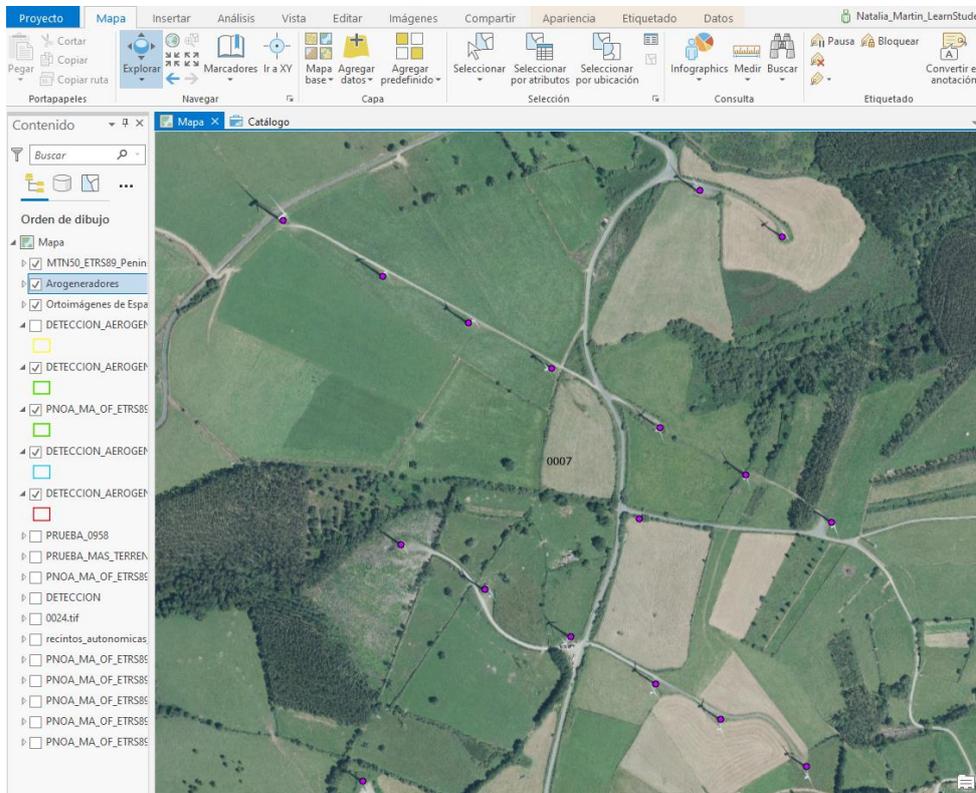


Figura 19: ejemplo localización aerogeneradores. Elaboración propia

En la [imagen 20](#) podemos ver que, en la parte inferior de la imagen, hay elementos que faltan por georreferenciar.

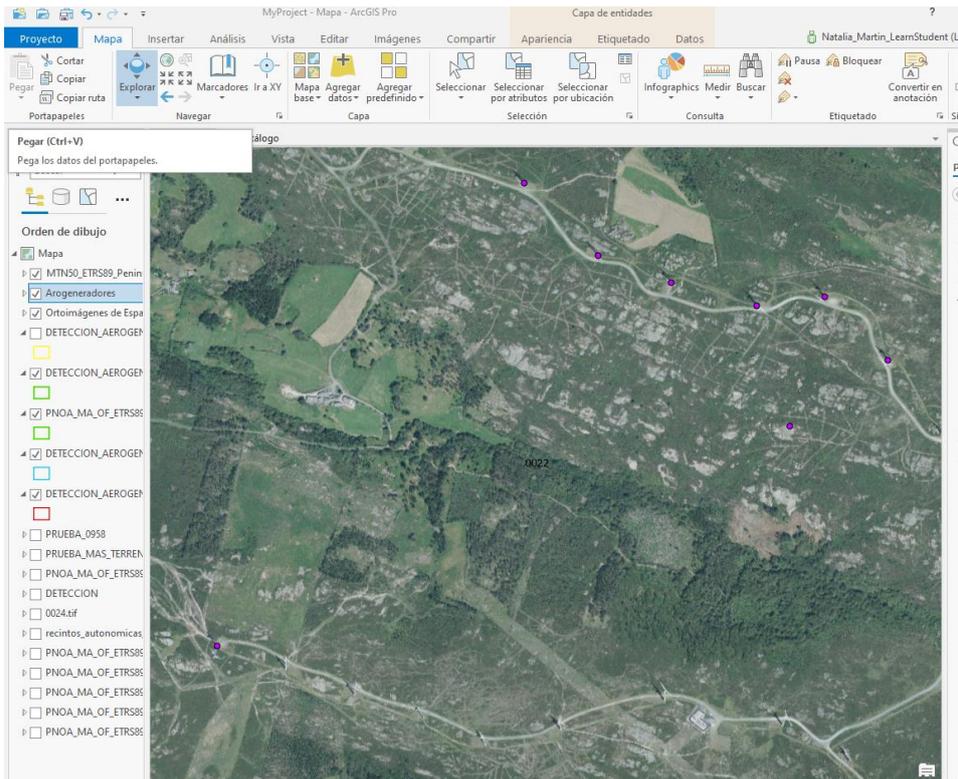


Figura 20: ejemplo aerogeneradores sin registrar. Elaboración propia

Y, por último, en el siguiente ejemplo que se muestra (figura 21) vemos el caso contrario. Se aprecia claramente que el lugar donde están los puntos violetas, eran antiguas instalaciones eólicas, pero que habrán sufrido una baja o desmantelamiento y esos aerogeneradores ya no existen.

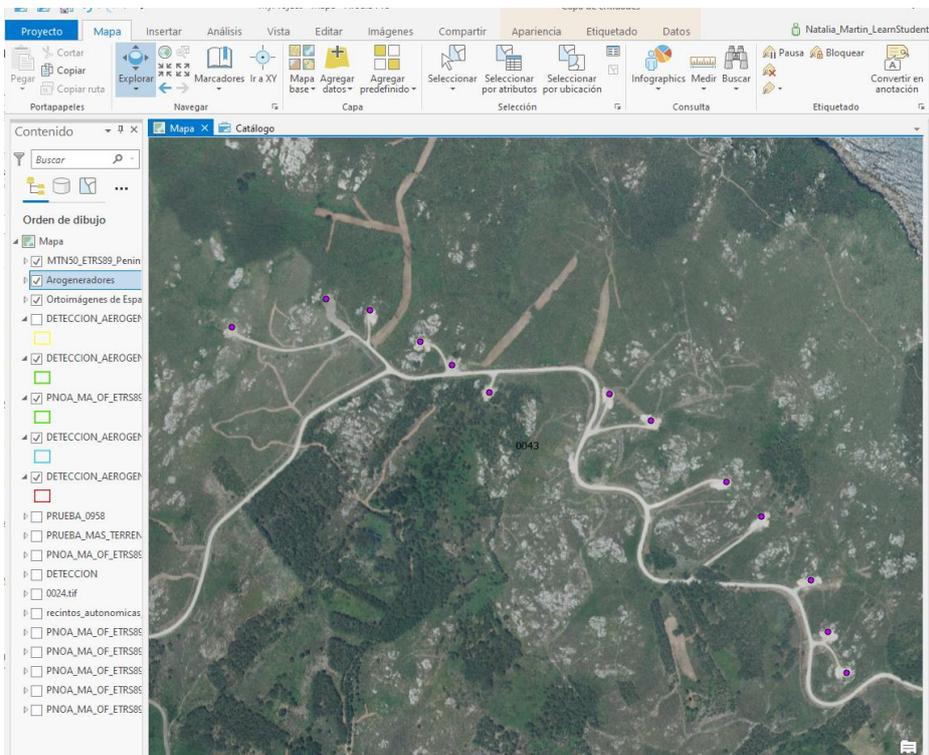


Figura 21: ejemplo espacio sin aerogeneradores. Elaboración propia

En este proyecto nos centraremos en el caso de la [imagen 20](#), es decir, localizar aquellos elementos que no se encuentren registrados en la capa de Aerogeneradores proporcionada.

## 5. METODOLOGÍA

### 5.1. METODOLOGÍA GENERAL

Antes de mostrar la metodología llevada a cabo con el programa utilizado en este proyecto (ArcGIS Pro), se va a proceder a explicar la metodología del Deep Learning de manera general, de esta forma, se tendrá una noción básica de este método de trabajo que puede ser útil para diversos Softwares que utilizan el Deep Learning. Algunos de estos programas son CNTK (Kit de herramientas cognitivas de Microsoft), TensorFlow, PyTorch, API de IA de vídeo de Cloudinary. También existen programas SIG que contienen esta herramienta, entre ellos destacamos, ERDAS IMAGINE, QGIS, ArcGIS, ArcGIS Pro.

Los pasos a seguir en la utilización de Deep Learning son los mismo independientemente del Software que se utilice, si es cierto que pueden existir algunas diferencias, pero en líneas generales funcionan de la siguiente manera:

#### 5.1.1. CREACIÓN DE UNA RED NEURONAL

Este tipo de redes neuronales establecen una arquitectura de conexiones en las que el modelo va aprendiendo automáticamente [10]. Antes de la existencia del Machine Learning, estas interconexiones las programaba el usuario, es decir, establecía unas reglas de relaciones y conexiones con las que el programa trabajaba. Sin embargo, con las creaciones de estas redes neuronales, el modelo incluso puede descubrir relaciones de variables que ni el humano conoce.

En la [figura 22](#) se muestra un ejemplo del esquema de una Red Neuronal.

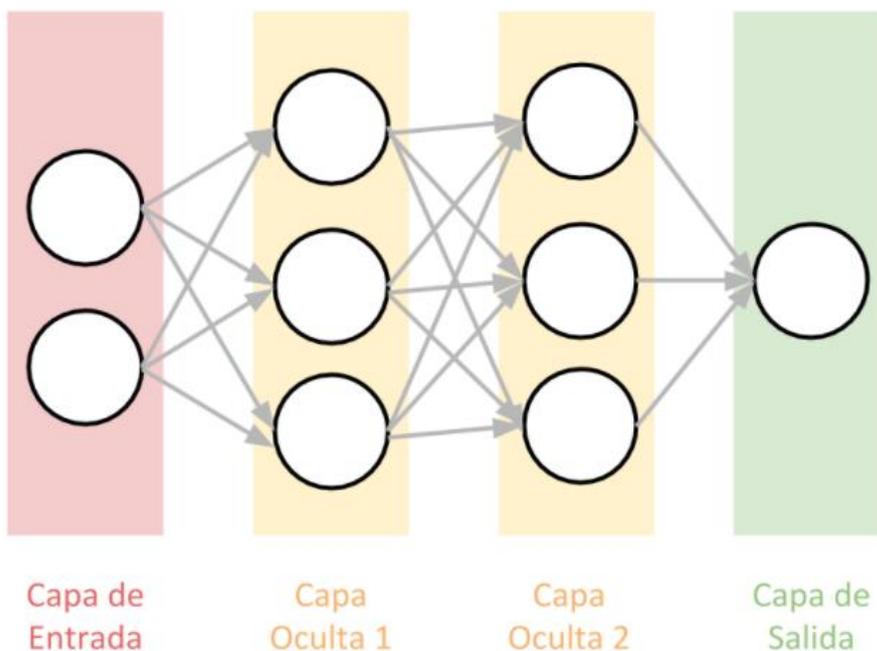


Figura 22: Esquema Red Neuronal: Fuente: <https://www.aprendemachinellearning.com/aprendizaje-profundo-una-guia-rapida/>

Como se puede observar en la [figura 22](#), las redes neuronales están compuestas siempre por tres estilos de capas diferentes. Estas capas son:

- Capa de entrada: es la capa que capta los datos de entrada y los transmite a la primera serie de capa oculta.
- Capa oculta: pueden ser más de una. Son las que realizan los cálculos matemáticos con las capas que se han introducido de entrada.
- Capa de salida: es el resultado que se devuelve en función de los cálculos realizados.

### 5.1.2. ENTRENAMIENTO DE LA RED NEURONAL

Esta parte probablemente sea la de mayor dificultad de todo el proceso del Deep Learning. Para que este entrenamiento sea efectivo se necesitan dos factores importantes. Uno de ellos es disponer de una gran variedad de datos, valores o imágenes que compongan la capa de entrada, y el otro es disponer de un cálculo computacional potente.

Cuando ya se disponga del conjunto de datos de muestra, el siguiente paso a seguir es el de realizar un proceso iterativo. En este proceso se crea una función que comparará el nivel de “bueno/malo” del resultado con respecto al dato real. Esta comparativa se realiza con la función “coste”. El caso perfecto sería que nuestro coste tuviera valor cero, es decir, que no exista error porque el valor resultado de la predicción es igual que el de la capa de entrada o resultado. Este valor cero será difícil alcanzarlo, pero a medida que el modelo vaya entrenando, irá ajustando los pesos de las conexiones de la red neuronal y así mejorara las predicciones.

### 5.1.3. DETECCIÓN OBJETOS

Con la red neuronal entrenada, el último paso es probar su funcionamiento. Se establece la imagen o datos de referencia para la detección de objetos y se introduce el modelo de entrenamiento de la red neuronal. Tras el proceso, el programa devolverá los resultados obtenidos y habrá que valorarlos y estudiarlos para ver si los resultados son los esperados.

Este proceso que se ha explicado, sería la regla general que siguen los Deep Learning y en función del programa que se utilice, existirán patrones o pasos que puedan cambiar en pequeños detalles.

Debido al campo al que pertenece el máster estudiado, lo adecuado es utilizar un programa SIG para la detección de estos objetos, ya que después, los datos obtenidos se pueden modificar, almacenar y exportar en diferentes formatos en función de lo que el usuario desee.

Antes de proceder con el objetivo del proyecto, se ha llevado a cabo un estudio y comprensión de la herramienta Image Analyst y en concreto Deep Learning.

## 5.2. METODOLOGÍA CON ARCGIS PRO

Para que esta herramienta funcione correctamente, hay que llevar a cabo tres pasos. Que son:

- Utilización de la herramienta de clasificación de imágenes para la creación de patrones.
- Creación del modelo de entrenamiento con los patrones definidos anteriormente.
- Detección de objetos utilizando el modelo generado.

### 5.2.1. UTILIZACIÓN DE LA HERRAMIENTA DE CLASIFICACIÓN DE IMÁGENES.

Esta opción no se encuentra dentro del Deep Learning, si no en la pestaña de imágenes del programa. (Figura 23)

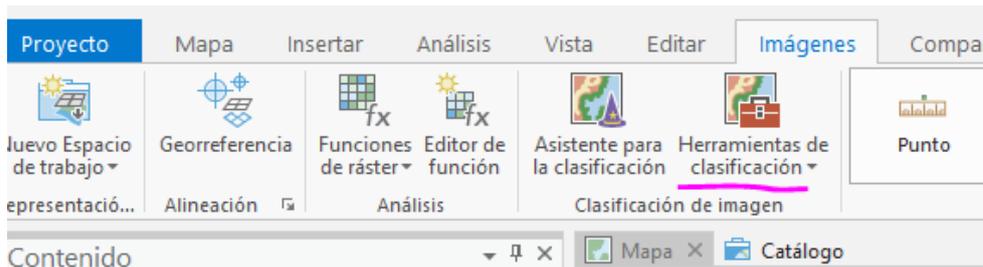


Figura 23: herramienta de clasificación. Elaboración propia.

Dentro de esta herramienta encontramos la opción de etiquetar objetos de aprendizaje profundo, que es la que utilizaremos para generar los patrones determinados del objeto a localizar.

En la [imagen 24](#) se muestra una parte de la Península Ibérica con la ortofoto cargada como servicio WMS [14], la capa añadida de las hojas de España del MTN50 (también descargada del IGN) [15] y la capa cargada de los aerogeneradores que se ha descargado del visor INSIGNIA [16]. De esta manera podemos ayudarnos a la hora de elegir las hojas adecuadas para realizar el estudio.



Figura 24: Zona de España con la capa de aerogeneradores geolocalizados. Elaboración propia.

El funcionamiento del etiquetado de objetos para aprendizaje profundo es sencillo. En primer lugar, se necesita una imagen de alguna zona que contenga los elementos que se quieren clasificar, en nuestro caso, para comenzar con el estudio de la herramienta, hemos descargado

del IGN la imagen satelital correspondiente a la hoja 0001 del MTN50 del PNOA de máxima actualidad debido a que sabemos que en esa zona existen los elementos a tratar.

Pulsamos sobre la opción de etiqueta de objetos para aprendizaje profundo y nos aparece la siguiente ventana.

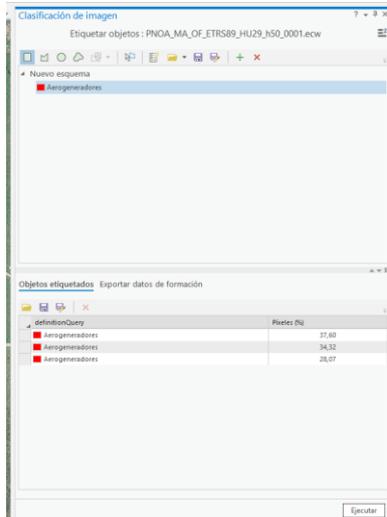


Figura 25: Ventana de clasificación de imagen. Elaboración propia.

En esta ventana creamos un nuevo esquema llamado Aerogeneradores. Una vez creado, generamos tantas etiquetas como consideremos necesarias. Como estamos en fase experimental de la herramienta, vamos a generar únicamente tres para ver el funcionamiento general y cómo se comporta. Más adelante, para lograr el objetivo marcado en este trabajo, el entrenamiento requerirá de más imágenes. Podemos ver un ejemplo de la clasificación de etiquetas (figuras, 26,27 y 28).



Figura 28: Ejemplo Clasificación



Figura 26: Ejemplo Clasificación



Figura 27: Ejemplo Clasificación

El siguiente paso que hay que realizar es el de exportación de las etiquetas. Aquí ya tenemos diferentes opciones que vamos a ir explicando paso por paso.

Tenemos la siguiente ventana de exportación de datos (figura 29):

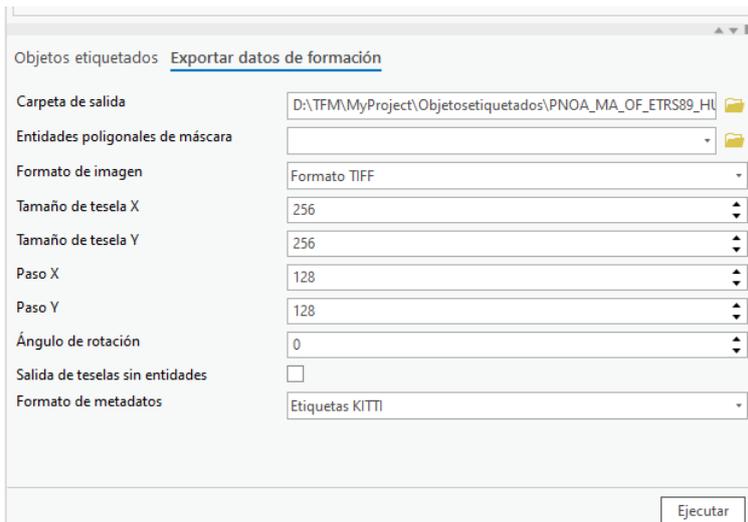


Figura 29: Exportación de datos de formación. Elaboración propia.

Introducimos la carpeta en la que queremos guardar las imágenes de entrenamiento.

Elegimos el formato de guardado, en el que tenemos las siguientes opciones:

- MRF (metarráster): este formato está diseñado para el almacenamiento en la nube. Divide el ráster en tres archivos separados.
- PNG: este formato es más conocido. Al guardar la información en este tipo de archivos, la información se mantiene intacta y no se pierde la calidad de la imagen.
- JPEG: son archivos de imagen comprimidos. Este tipo de formato pierde información en su almacenamiento y no sería muy recomendable para las imágenes de un mapa.
- TIFF: este tipo de formato está diseñado especialmente para la importación de edición de imágenes y además son compatibles con datos de ráster GIS.

Dada la naturaleza de este trabajo, el formato elegido de exportación de etiquetas para lograr los objetivos propuestos será en TIFF.

Los siguientes 4 apartados de la [imagen 29](#), son datos opcionales que se pueden modificar en función de lo que se pretenda obtener. Vamos a explicar cada uno de ellos.

- Tamaño de tesela X e Y: corresponden a los tamaños de los chips de imagen en las dimensiones X e Y. Es decir, es el número de píxeles que se van a almacenar en cada imagen, por lo tanto, se puede decir que esta opción es el tamaño de la imagen.
- Paso X e Y: este valor determina la distancia a la que se van a desplazar la dirección X e Y respectivamente. Se utiliza para la superposición de imágenes. Si este valor coincide con el establecido en el tamaño de la tesela, la superposición será nula. Si se determina la mitad del valor de tesela, esta superposición será del 50%.

Como estamos en fase experimental de la herramienta, se van a dejar estas opciones con los valores predeterminados que aparecen en la imagen.

El siguiente atributo que se puede establecer es el del ángulo de rotación. Si en este apartado se introduce un valor angular, lo que hará al exportarse será generar más chips de imágenes con el ángulo de rotación establecido. En este caso dejaremos valor 0, tal y como viene predeterminado.

La salida de teselas sin entidades significa que existe la opción de exportar los chips de imagen que no contengan muestras de entrenamiento. En este caso dejaremos esta selección sin marcar, como viene por defecto y analizaremos los resultados obtenidos.

Por último, hay que seleccionar el formato de los metadatos, aquí también tenemos diversas opciones que vamos a detallar [11].

- Etiquetas PASCAL Visual Object Classes: estos metadatos siguen el formato del conjunto de datos PASCAL VOC (Pattern Analysis, Statistical Modeling and Computational Learning Visual Object Classes). Este tipo de datos constan de dos componentes, en primer lugar, un conjunto de imágenes y en segundo lugar una clasificación. En este tipo de formatos se almacenan los datos del nombre de imagen, el valor de cada clase y los delimitadores de ellos que consta. Se almacenan en archivos XML, lo que hace fácil su lectura.
- Teselas clasificadas: en este tipo de formato se extrae una imagen por cada etiquetado, es decir, lo extrapola y cada imagen contiene información complementaria.
- Teselas etiquetadas: a diferencia de la anterior, si se escoge esta opción, entonces cada etiqueta de salida se exportará con una clase definida. Con esta selección, se podrían establecer los parámetros Blacken Around Feature [t3] y Crop Mode [t4].
- Etiquetas KITTI: los archivos generados con este etiquetado son archivos de texto sin formato.
- Máscara RCNN [12]: Para entender mejor como se crean las máscaras RCNN, vamos a explicar en qué consiste el algoritmo RCNN y el Faster RCNN por separado.

RCNN (figura 30): las características obtenidas con RCNN permiten mejorar la representación de los objetos durante su detección, sin embargo, este método requiere un alto coste computacional, ya que la extracción de características debe hacerse de forma individual para cada región en la que se quiera localizar objetos.

- o Ingresar una imagen a la red neuronal.
- o Extraer las regiones de una imagen que contenga el objeto a localizar.
- o Utilizar el transfer learning [t1] para la creación de características de los objetos.
- o Clasificación de las propuestas en función de las regiones establecidas.

Faster-R-CNN (figura 31): es similar al RCNN pero con la mejora en cuanto a la detección de objetos, ya que permite que se puedan detectar a tiempo real:

- o Ingresar una imagen y sus cuadros de delimitación adecuados.
- o Extraer mapa de características.
- o Aplicación de la agrupación RoI y obtener el vector de características.
- o Usar el conjunto de las dos capas completamente conectadas para obtener las predicciones de la etiqueta de clase, y las ubicaciones del cuadro limitador de cada objeto.

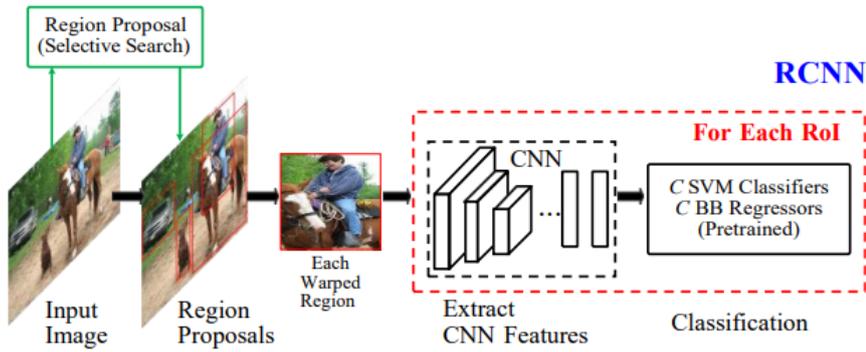


Figura 30: Funcionamiento RCNN. Fuente: <http://repositorio.unsa.edu.pe/bitstream/handle/20.500.12773/12247/ISnichw.pdf?sequence=1&isAllowed=y>

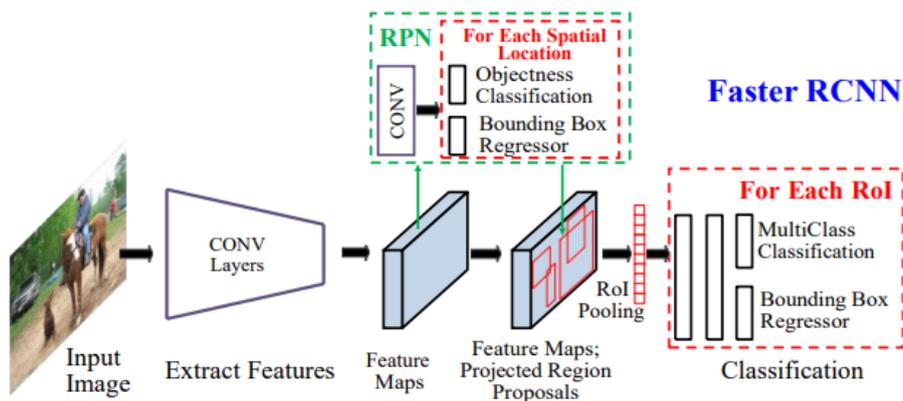


Figura 31: Funcionamiento Faster RCNN. Fuente: <http://repositorio.unsa.edu.pe/bitstream/handle/20.500.12773/12247/ISnichw.pdf?sequence=1&isAllowed=y>

Máscara R-CNN [13] (figura 32): con este método de detección, la localización es más precisa. Esto es gracias a que el algoritmo realiza una segmentación de los objetos que se encuentran en las imágenes y predice la localización exacta de cada pixel de la imagen que corresponde a los elementos de búsqueda. La desventaja en cuanto a los métodos explicados anteriormente es que el tiempo que necesita emplear para la detección de objetos es mayor. No obstante, este método es el adecuado para usar en este proyecto.

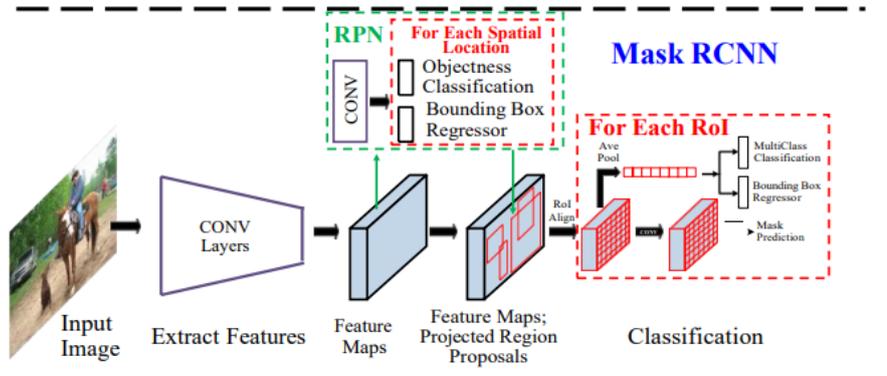
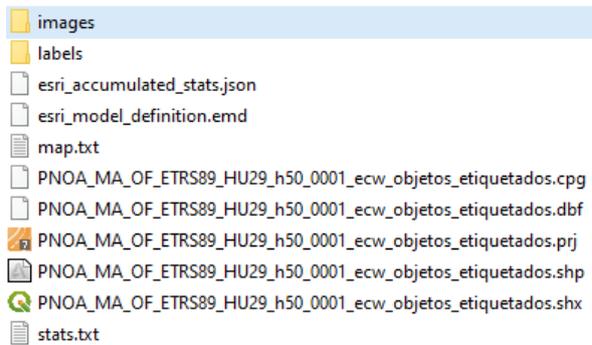


Figura 32: Funcionamiento Mask RCNN

Ya hemos explicado todos los formatos de metadatos disponibles que existen en la herramienta de exportación de datos de formación. Para el presente trabajo utilizaremos las máscaras RCNN.

Una vez establecidos todos los parámetros de exportación, se ejecuta la herramienta y en la carpeta que se había seleccionado para la exportación aparecen los siguientes archivos y carpetas ([figura 33](#)).



*Figura 33: Exportación datos de formación. Elaboración propia.*

En la carpeta “images” se encuentran las imágenes que la herramienta ha generado a partir de las características seleccionadas. En concreto, en esta ocasión, de las tres imágenes de etiquetado de las que partía ([figuras 26, 27 y 28](#)), la herramienta ha generado 29 imágenes en formato .tif. La razón por la que se ha generado este número de imágenes es por lo que se ha explicado anteriormente en el apartado de los pasos X e Y que determinada la distancia de desplazamiento en estos ejes. Este tipo de desplazamientos es una de las opciones que ofrecen las redes neuronales. Estas redes necesitan de gran cantidad de datos de entrenamiento para lograr una eficiencia, precisión y rendimiento mejores. Por esta razón, se genera un aumento de la imagen “Image Augmentation”, creando imágenes complementarias de entrenamiento mediante funciones como desplazamientos y rotaciones entre otras. [[17](#)]

En la carpeta “labels” se encuentran las etiquetas de estas imágenes creadas.

En el archivo “esri\_accumulated\_stats” se encuentran las estadísticas y características establecidas. En el [apartado 1](#) de anexos encuentran los contenidos de este documento.

El archivo “esri\_model\_definition” ([apartado 2 de anexos](#)) contiene una serie de características que van a ser de utilidad para el siguiente paso del proceso que será el de la creación de un modelo de entrenamiento. Podemos destacar algunas de estas características como son el tamaño de las imágenes, la extracción de bandas, el modelo de metada establecido...etc.

Por último, destacamos también que se ha creado una capa con las ubicaciones de los cuadros delimitadores de los objetos. Si cargamos esa capa en el programa, tiene este aspecto ([figura 34](#)):



Figura 34: Zoom a un elemento perteneciente a la capa de las etiquetas creadas. Elaboración propia.

### 5.2.2. CREACIÓN DEL MODELO DE APRENDIZAJE PROFUNDO.

En este apartado comenzamos a utilizar la herramienta que se encuentra en la pestaña de análisis y que contiene la llamada “Image Analyst Tools”. Dentro de esta herramienta encontramos las siguientes opciones:

- Classify Objects Using Deep Learning
- Classify Pixels Using Deep Learning
- Detect Objects Using Deep Learning
- Non Maximun Suppression
- Train Deep Learning Model

Para lograr el objetivo de este proyecto, la herramienta que tendremos que usar será la de detectar objetos usando el aprendizaje profundo, pero antes, necesitamos crear el modelo de aprendizaje y eso se hace con la última herramienta “Train Deep Learning Model”

Esta herramienta generara un modelo de entrenamiento con las imágenes aportadas en el proceso anterior.

De nuevo en este paso vamos a detenernos y explicar en la medida de lo posible los parámetros que existen, las variedades que contemplan y sus características.

Tenemos esta ventana ([figura 35](#)):

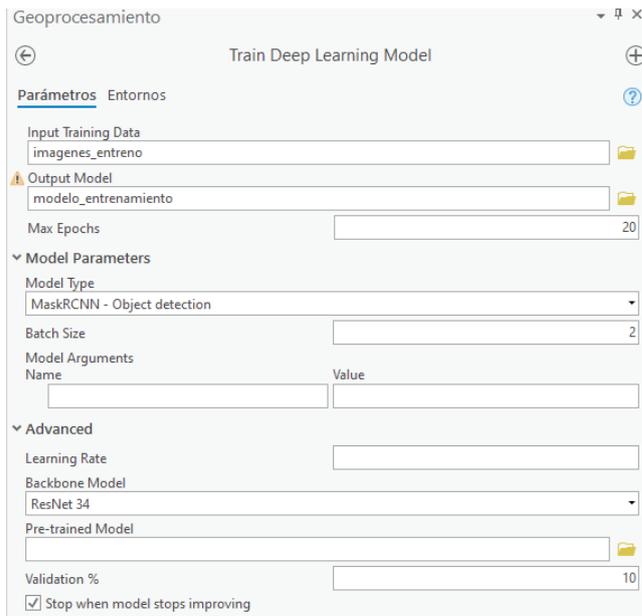


Figura 35: Creación Modelo de entrenamiento profundo. Elaboración propia.

En Input Training Data hay que introducir la dirección donde hemos almacenado los datos de formación. Hay que seleccionar la carpeta entera, no solo las imágenes de entrenamiento si no el conjunto de todos los archivos, ya que esta herramienta necesita los datos de etiquetado, las imágenes y los valores que se encuentran establecidos en los archivos mencionados de la [figura 33](#) de este documento.

En Output Model hay que seleccionar la carpeta de salida en la que se va a almacenar en modelo de entrenamiento.

Max Epochs hace referencia al número máximo de épocas por las que el modelo se va a entrenar. El valor por defecto es 20, esto significa que el conjunto de datos pasará hacia delante y hacia atrás a través de la red neuronal 20 veces.

Model Parametres:

- Model Type: este valor va asociado a las carecteristicas que se hayan introducido en el apartado anterior, como hemos elegido MaskRCNN, al introducir el Input Training Data, automáticamente el programa reconoce que ese es el formato que debe establecer para crear el modelo de entrenamiento y no deja cambiarlo por otro.
- Batch Size: el valor que aparece por defecto es 2. Este número corresponde al número de muestras de entrenamiento que se procesaran para el entrenamiento de una sola vez. Este número puede aumentar en función de la GPU, a mayor potencia de GPU, mayor valor se podrá establecer. En este caso se deja el valor que viene por defecto.
- Model Arguments: este tipo de parámetros son opcionales y está dirigido para gente experimentada y que sabe más acerca del tema del Deep Learning. En este caso vamos a dejar los parámetros tal y como vienen por defecto, es decir, sin rellenar.

Advanced:

- Backbone Model: es un parámetro que especifica la red neuronal preconfigurada y que se utilizará como arquitectura para el entrenamiento del nuevo modelo. Se va a dejar el valor que viene por defecto que es "ResNet 34" [t2], se trata de una red residual entrenada en el conjunto de datos ImageNET que contiene más de un millón de

imágenes y tiene 34 capas de profundidad. Esta variante de 34 capas aplica una serie de convoluciones en un bloque de muchas capas consecutivas. Además, ResNet-34 introduce unas conexiones entre las capas, es decir, cada capa se apoya de la anterior (sumar la entrada a la salida de cada una). Gracias a estas conexiones, la red neuronal puede decidir qué capas usar, de tal manera que el usuario le dará múltiples capas y la propia red neuronal decide cuales utiliza.

- Validation: se refiere al porcentaje de muestras de entrenamiento que se utilizarán para validar el modelo. El valor por defecto es 10.

Con los parámetros establecidos, ejecutamos la herramienta y en la carpeta parecen los siguientes archivos ([figura 36](#)):

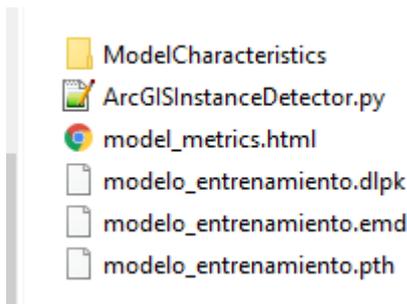


Figura 36: Carpeta de salida del modelo. Elaboración propia.

A su vez, en la carpeta que se creó al principio para almacenar los datos de formación, se ha creado otra carpeta llamada “models”, que está enlazada al modelo de entrenamiento generado.

En el apartado 3 de anexos se incluyen los contenidos de los archivos de esta carpeta.

Ya tenemos el modelo de entrenamiento creado, el último paso es detectar objetos en una imagen en concreto utilizando este modelo.

### 5.2.3. DETECCIÓN DE OBJETOS UTILIZANDO DEEP LEARNING

Ventana de Parámetros ([figura 37](#)):

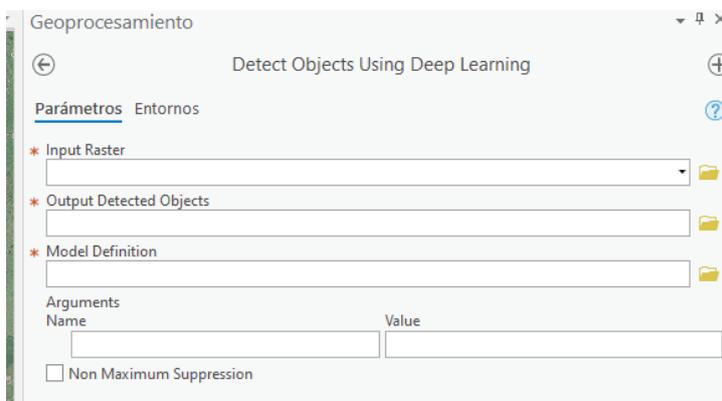


Figura 37: Parámetros de Detect Objects- Elaboración propia.

- Input Raster: se selecciona la imagen en la que se quiere que se detecten los objetos. En este caso será la “PNOA\_MA\_OF\_ETRS89\_HU29\_h50\_0001”

- Output Detected Objects: se selecciona la carpeta de salida en la que se va a almacenar la capa de los objetos detectados.
- Model Definition: se añade el modelo de entrenamiento creado en el apartado anterior.
- Arguments: estos argumentos determinan el umbral de confianza, los parámetros de aprendizaje profundo, de refinamiento. El valor de estos es opcional, se dejan por defecto tal y como aparecen en la [imagen 38](#):

Arguments Name	Value
padding	56
batch_size	1
threshold	0.9
return_bboxes	False

Non Maximum Suppression

Confidence Score Field  
Confidence

Class Value Field  
Class

Max Overlap Ratio

Figura 38: Argumentos para detección de objetos. Elaboración propia.

Se marca la opción de “Non Maximun Suppression”. Esto significa que se van a identificar los objetos duplicados y a eliminar.

Ventana de Entornos ([figura 39](#)):

Geoprocesamiento # x

Detect Objects Using Deep Learning +

Parámetros Entornos ?

▼ Análisis de ráster

Tamaño de celda  
Cantidad máxima de entradas 📁

▼ Tipo de procesador

Tipo de procesador ▼

Id. de GPU

▼ Coordenadas de salida

Sistema de coordenadas de salida 🌐

Transformaciones geográficas ▼

▼ Extensión de procesamiento

Extensión

▼ Procesamiento paralelo

Factor de procesamiento en paralelo

Figura 39: Entornos del Detect Objects. Elaboración propia.

Esta ventana es especialmente importante, ya que se va a establecer la forma de búsqueda de los objetos, el área de búsqueda y el tipo de procesador (gráficos o del equipo).

- Análisis de ráster: se va a seleccionar la opción de “cantidad máxima de entradas”. Esto significa que se va a utilizar el tamaño de celda más grande de todos los datos de entrada que existan.
- Tipo de procesador: como se explicó en las características del material empleado para este proyecto, disponemos de una GPU para el tratamiento de gráficos, por lo que la opción a elegir será GPU
- Coordenadas de Salida: en este caso no es necesario establecer un sistema de coordenadas de salida, ya que la capa que se va a generar es de tipo superficial y no puntual, por lo que no se necesitarán coordenadas ni transformaciones de coordenadas.
- Extensión de procesamiento: esta opción puede ser de las más importantes, ya que aquí se determina el área de búsqueda de la imagen seleccionada. Podemos elegir que busque en toda la imagen seleccionada o en la extensión actual que se esté visualizando. Vamos a elegir una pequeña extensión de terreno que contenga aerogeneradores para comprobar el funcionamiento de la herramienta.

Con todos los parámetros establecidos, ejecutamos la herramienta y obtenemos una capa que cargamos al programa y este es el resultado:



*Figura 40: Resultados obtenidos en la fase de experimentación. Elaboración Propia.*

Se puede observar que, en esta primera prueba de la herramienta ([figura 40](#)), los resultados han sido óptimos y se han localizado los aerogeneradores que se encontraban en ese espacio de la hoja.

## 6. RESULTADOS

Para lograr unos resultados óptimos y de confianza, lo que se ha realizado en primer lugar es una recopilación de aerogeneradores de diferentes partes del territorio nacional. De esta manera se tendrá mayor variedad y el modelo de entrenamiento tendrá más etiquetas de imagen en las que apoyarse y comparar. A mayor número de imágenes recopiladas para crear el modelo, mayor será la precisión a la hora de encontrar los objetos deseados.

Por lo tanto, en el paso de la clasificación de imágenes, se han seleccionado 100 aerogeneradores a lo largo de toda la Península Ibérica para tener mayor variedad, es decir, sombras con diferente proyección, suelos variados y posiciones variadas.

A continuación, se muestran algunos ejemplos de esta clasificación de imágenes ([figura 41](#)).



Figura 41: conjunto de imágenes de muestra. Elaboración propia.

Con estas 100 imágenes se ha creado el modelo de entrenamiento de aprendizaje profundo siguiendo los mismos pasos que en el apartado de la metodología.

Tras crear este modelo de aprendizaje, se genera una imagen que muestra la información acerca de la relación existente entre la validación y el entrenamiento.

En la figura 42 se puede ver esta gráfica mencionada. Al principio de ella, el entrenamiento tiene mayor valor que la validación de los datos, esto significa que el valor de entrenamiento de los datos está por encima de su validación, es decir, que el entrenamiento que se ha llevado a cabo ha sido correcto para la detección de patrones. Sin embargo, al final de la gráfica, la línea de validación de los datos supera ligeramente a la de entrenamiento, esto significa que los datos de entrenamiento establecidos no son lo suficientemente representativos a la hora de proporcionar una validación correcta o, en otras palabras, que el modelo de entrenamiento puede resultar demasiado complicado para resolver el problema que se plantea. No obstante, se considera la gráfica correcta para la detección de objetos en este proyecto ya que, los valores no son muy distantes entre sí

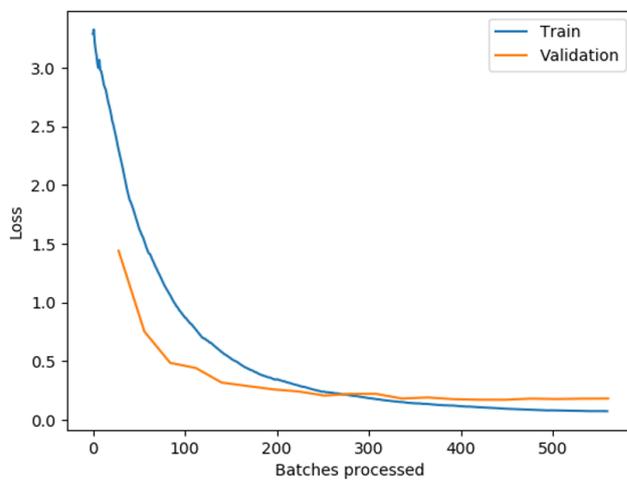


Figura 42: Model Characteristics. Elaboración propia.

Para que esta gráfica mejorase en cuanto a entrenamiento y validación de datos, se necesitaría crear un modelo de entrenamiento compuesto por más imágenes de muestra. Recordemos que se han establecido 100 imágenes para realizar el proyecto, sin embargo, la limitación del hardware empleado, impide que estos valores se puedan mejorar.

La figura 43 muestra una gráfica que cumpliría mejor los requisitos de validación y entrenamiento.

Se observa que ambos valores siguen prácticamente la misma línea, esta gráfica se acercaría más a un modelo de entrenamiento idóneo para la clasificación de imágenes.

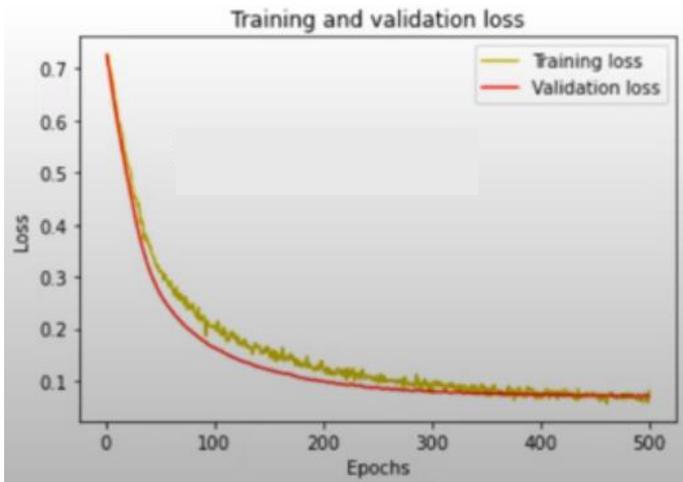


Figura 43: Ejemplo Model Characteristics. Fuente: <https://archive.ics.uci.edu/ml/index.php>

Con el modelo creado se han realizado diversas pruebas en zonas diferentes del territorio nacional, pero esta vez abarcando más terreno que en ejemplo que se llevó a cabo en la metodología.

Estos fueron los primeros resultados:

Parte de la hoja PNOA\_MA\_OF\_ETRS89\_HU29\_h50\_0001 (figura 44):

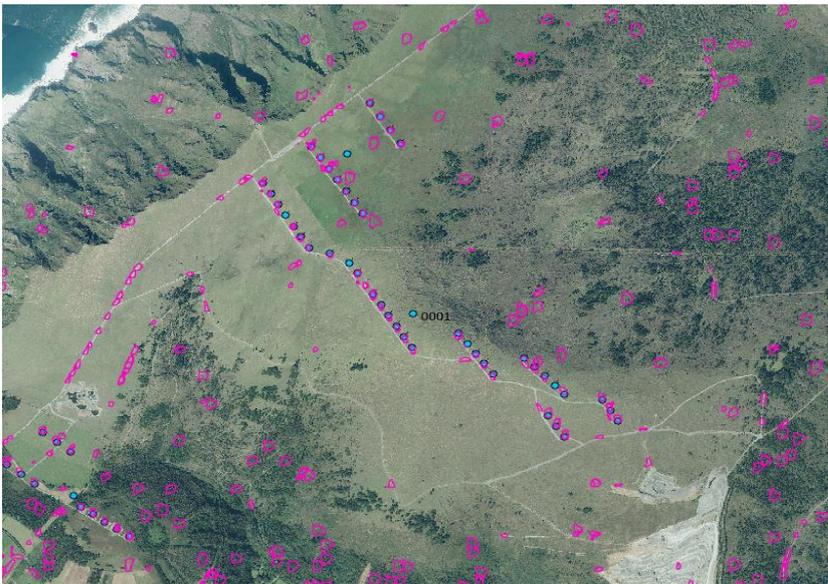


Figura 44: Resultados iniciales hoja 0001 con el modelo de 100 imágenes. Elaboración propia.

Vemos que los aerogeneradores los ha localizado (puntos azules correspondientes a la capa de información de la que partimos), sin embargo, ha interpretado más zonas del terreno como si fueran tal objeto y corresponden a caminos, piedras, o incluso a terrenos similares a los existentes bajo los aerogeneradores.

Parte de la hoja PNOA\_MA\_OF\_ETRS89\_HU29\_h50\_0024 (figura 45):



Figura 45: Primeros resultados hoja 0024 con el modelo de 100 imágenes. Elaboración propia

De nuevo, en esta ocasión la herramienta ha sido capaz de identificar los objetos requeridos, pero ha vuelto a destacar algunas zonas del terreno que son caminos o zonas blancas y que por lo tanto la composición tanto del color como de la forma pueden ser similares a las aspas de los aerogeneradores y el programa los interpreta como tal.

Parte de la hoja PNOA\_MA\_OF\_ETRS89\_HU29\_h50\_0980 (figura 46):



Figura 46: Primeros resultados hoja 0980 con el modelo de 100 imágenes. Elaboración propia

Volvemos a encontrarnos con la problemática de los dos casos anteriores. No obstante, en esta ocasión podemos decir que es la primera vez que el programa nos ha localizado

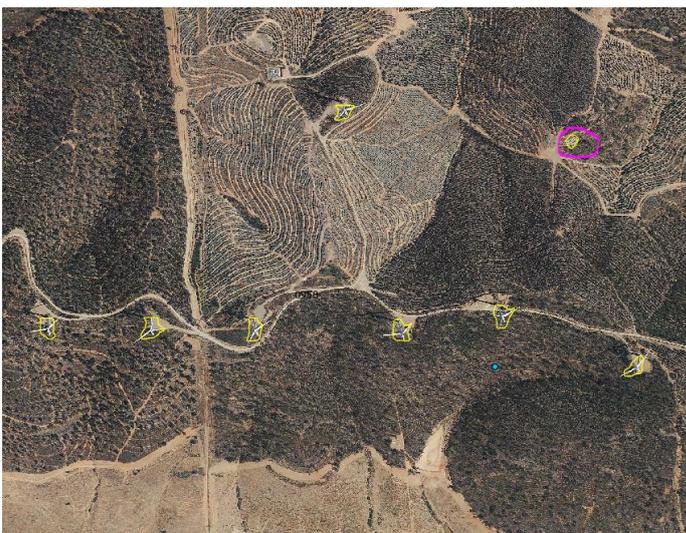
aerogeneradores de los que no se “conocía” su existencia. Tenemos la capa de la localización de los ejemplos (parte inferior de la [figura 46](#) se observa un punto). Pero el resto de aerogeneradores no estaban registrados.

Parte de la hoja PNOA\_MA\_OF\_ETRS89\_HU29\_h50\_0958 ([figura 47](#)):



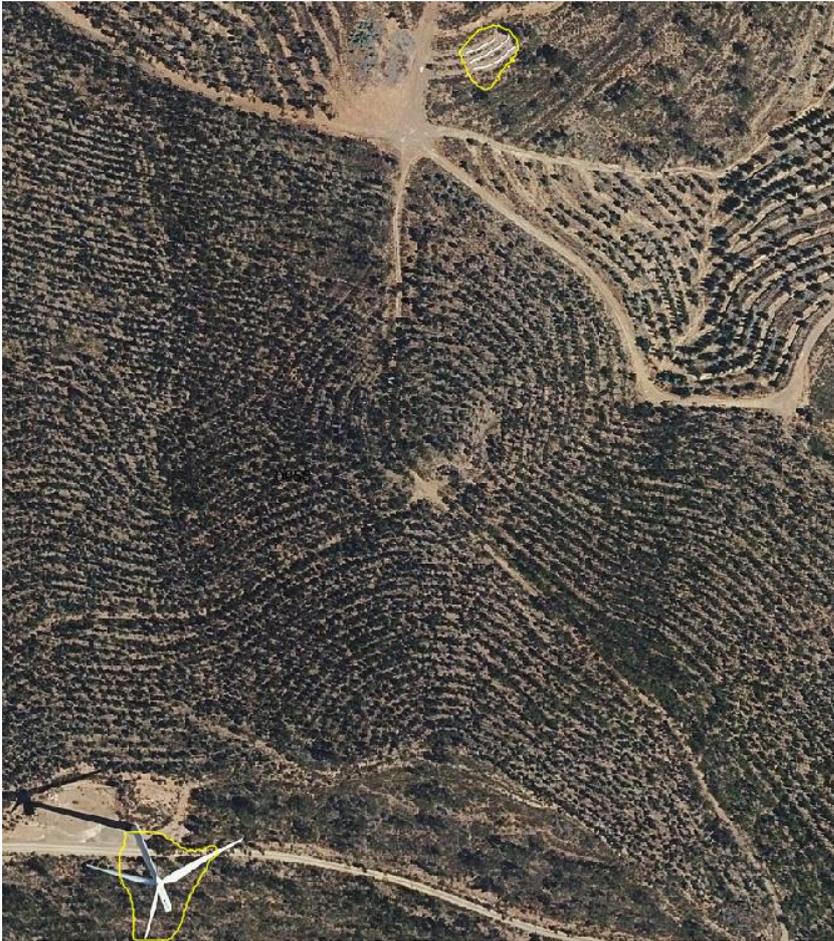
*Figura 47: Primeros resultados hoja 0958 con el modelo de 100 imágenes. Elaboración propia*

Como viene ocurriendo en los ejemplos anteriores, en esta ocasión también se han localizado aerogeneradores y otros elementos que no pertenecen a estos objetos. Vamos a mostrar dos imágenes a escala menor ([figuras 48](#) y [49](#)) para que se aprecie mejor la búsqueda que ha realizado la herramienta.



*Figura 48: Imagen 47 ampliada. Elaboración propia.*

Se puede apreciar que en la parte superior de la [figura 49](#), la herramienta ha interpretado que esa parte del terreno es un aerogenerador. Esto se debe a la similitud del color y forma.



*Figura 49: Ejemplo de zona detectada con un falso positivo. Elaboración propia.*

Haciendo zoom a esa zona, vemos que se puede tratar de una zona de invernaderos o de algún tipo de instalación en línea que se asemeja a las aspas de un aerogenerador.

#### Clasificación de los resultados

Este tipo de resultados obtenidos se puede clasificar estadísticamente en función de: verdadero positivo, falso positivo, verdadero negativo y falso negativo. Vamos a detallar cuándo se aplica cada término.

- Verdadero positivo: la referencia es verdadera y la predicción también, es decir, existe aerogenerador en la imagen de búsqueda y el modelo lo ha detectado.
- Falso positivo: la referencia es falsa y la predicción es verdadera, es decir, no existe aerogenerador en la imagen de búsqueda, pero el modelo lo ha detectado.
- Verdadero negativo: la referencia es falsa y la predicción ha sido también falsa. Es decir, el objeto deseado no se encuentra en la imagen de búsqueda y el modelo no lo ha detectado.
- Falsos negativos: la referencia es verdadera y la predicción falsa, es decir, existe el objeto en la imagen, pero el modelo no lo ha localizado.

Solución llevada a cabo tras observar los primeros resultados obtenidos de zonas de mayor extensión:

En primer lugar, se ha llevado a cabo una investigación de las etiquetas generadas tras utilizar la generación de imágenes. Se han visualizado las imágenes de la carpeta "labels" mencionada en el apartado de la metodología de este trabajo.

En este conjunto de imágenes (figura 50), se muestran algunas de estas etiquetas para comprender la problemática existente.



Figura 50: ejemplo de zonas desplazadas. Elaboración propia.

Tras analizar estas etiquetas, se ha llegado a la conclusión de que algún parámetro introducido a la hora de la clasificación de imágenes, no estaba correcto del todo o no hacía cumplir nuestro objetivo, ya que, en las figuras mostradas anteriormente, vemos que no se encuentran aerogeneradores, si no los terrenos adyacentes a estos objetos.

Si recordamos los parámetros explicados anteriormente, teníamos que:

Paso X e Y (figura 51): este valor determina la distancia a la que se van a desplazar la dirección X e Y respectivamente. Se utiliza para la superposición de imágenes. Si este valor coincide con el establecido en el tamaño de la tesela, la superposición será nula. Si se determina la mitad del valor de tesela, esta superposición será del 50%.

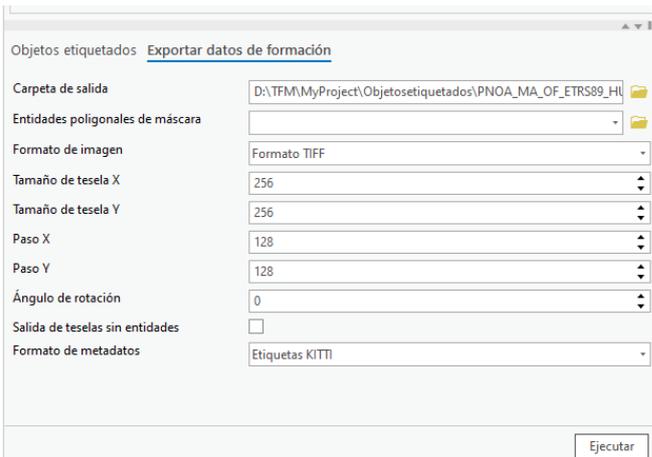


Figura 51: exportación de datos de formación. Elaboración propia.

Estos valores se habían establecido como la mitad del valor de la tesela, por lo que se ha producido una superposición del 50% en las etiquetas generadas.

Lo que se ha hecho para mejorar este etiquetado, es elevar el valor de parámetros Paso X y Paso Y así tener menos desplazamiento en las direcciones X e Y.

Se ha generado por lo tanto un nuevo modelo de entrenamiento de aprendizaje profundo, pero con los datos depurados, obteniendo en este casi etiquetas más concretas. En la [imagen 52](#) podemos observar algunas de estas etiquetas.



Figura 52: etiquetas para el modelo de entrenamiento. Elaboración propia.

Aunque sigue habiendo zonas bastante desplazadas del objeto principal, en esta ocasión en todas las etiquetas generadas el patrón que debe ser objeto del estudio (aerogenerador) se encuentra en todas las imágenes.

Con el nuevo modelo generado, se ha vuelto a usar la herramienta para la detección de objetos mediante Deep Learning.

Hasta ahora, todos los procesos de búsqueda que se han ido realizando en este proyecto han sido sobre zonas de hojas, en ninguna ocasión se ha intentado realizar búsquedas de extensiones muy grandes.

En esta ocasión, se ha intentado realizar la búsqueda en una hoja completa. La hoja PNOA\_MA\_OF\_ETRS89\_HU29\_h50\_0001 tiene un tamaño de casi 3 Gigabits, debido a este gran tamaño de imágenes y a las características del ordenador utilizado los procesos de búsqueda se han extendido hasta dieciocho e incluso veintitrés horas y en la mayoría de las ocasiones, se originaba un error y ArcGIS Pro dejaba de funcionar.

En la [figura 53](#) se muestra un ejemplo de la búsqueda completa de la Hoja 0001 y el tiempo que llevó en generarla.

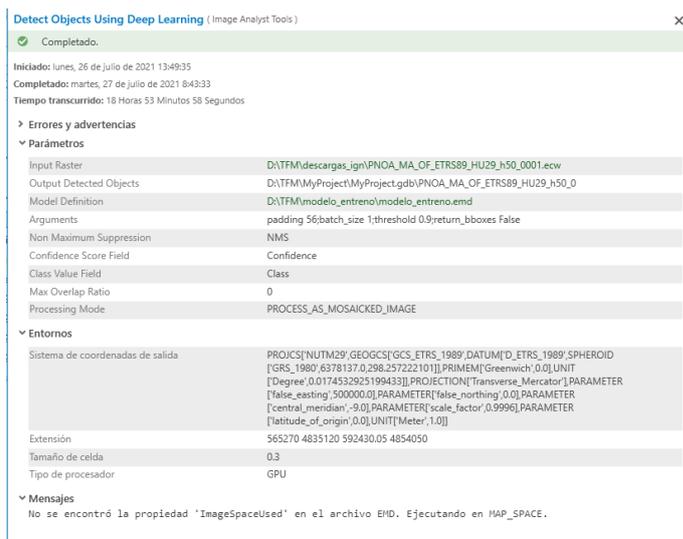


Figura 53: informe de la detección de objetos por aprendizaje profundo. Elaboración propia.

Para resolver este pequeño inconveniente y que la búsqueda sea óptima y eficiente, en Google Maps se ha localizado el parque eólico más grande de España que está en El Andévalo, Huelva ([figura 54](#)). La razón por la que se ha decidido realizar la búsqueda en esta zona es porque, al tratarse del mayor complejo eólico de España, encontraremos bastantes objetos de este tipo y, además, en el archivo descargado de INSIGNIA con las localizaciones de los aerogeneradores, este parque no está registrado en esta base de datos.

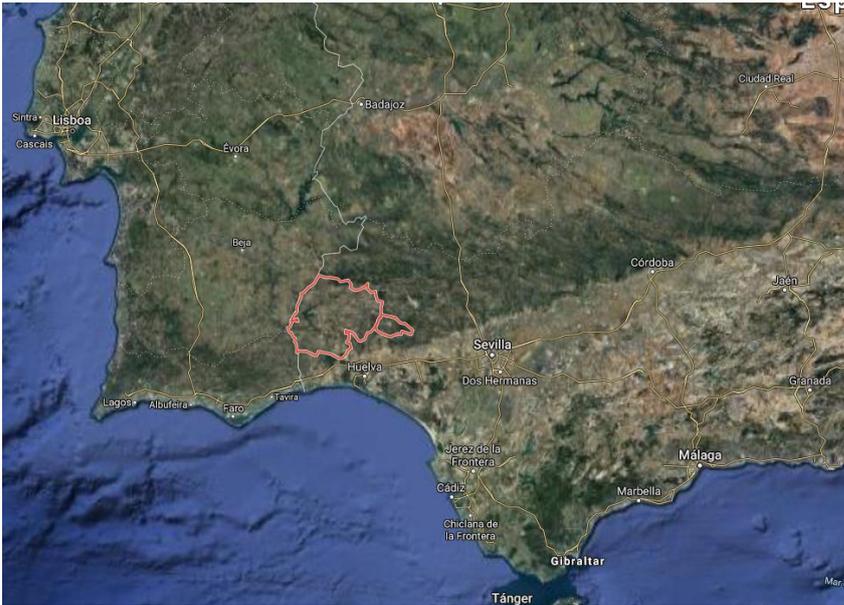


Figura 54: localización del parque eólico más grande de España. Elaboración propia.

Una vez que se ha localizado este complejo, con la ayuda de la ortofoto del PNOA y la descarga de las cuadrículas del MTN50, hemos podido localizar las hojas en las que se encuentra este parque eólico. Se trata de las hojas 0958 y 0980. (figura 55)

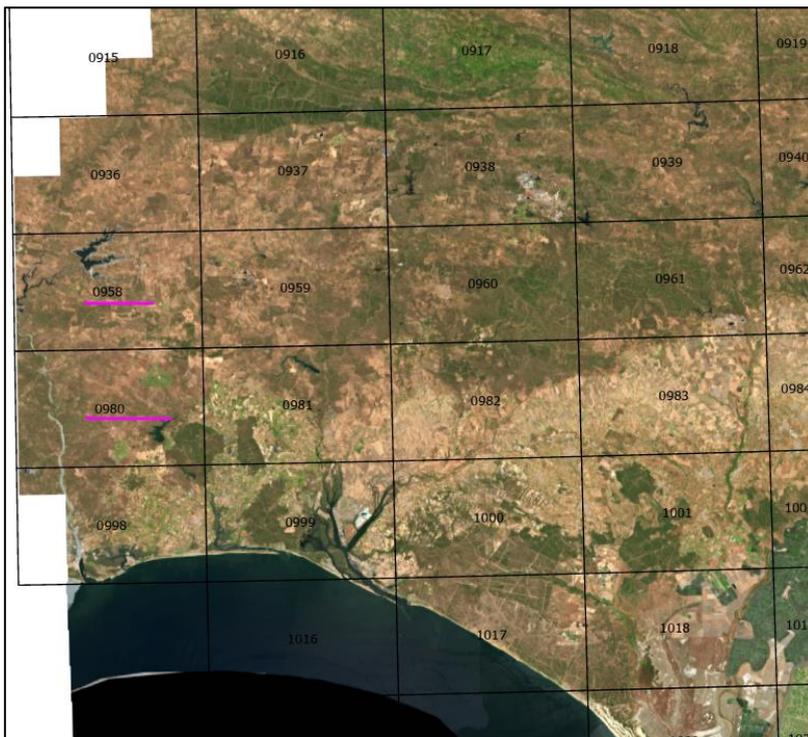


Figura 55: localización del parque eólico en las hojas del MTN50. Elaboración propia.

Como se ha mencionado anteriormente, no se va a realizar la búsqueda en la hoja completa. Lo que se ha decidido llevar a cabo es una división de cada hoja por cuadrantes (figuras 56 y 57). La división de estas zonas ha sido aproximada, no se busca que las 4 zonas sean iguales, sino que, con la ayuda de la geolocalización del parque eólico, se ha estimado dónde pueden estar los aerogeneradores.

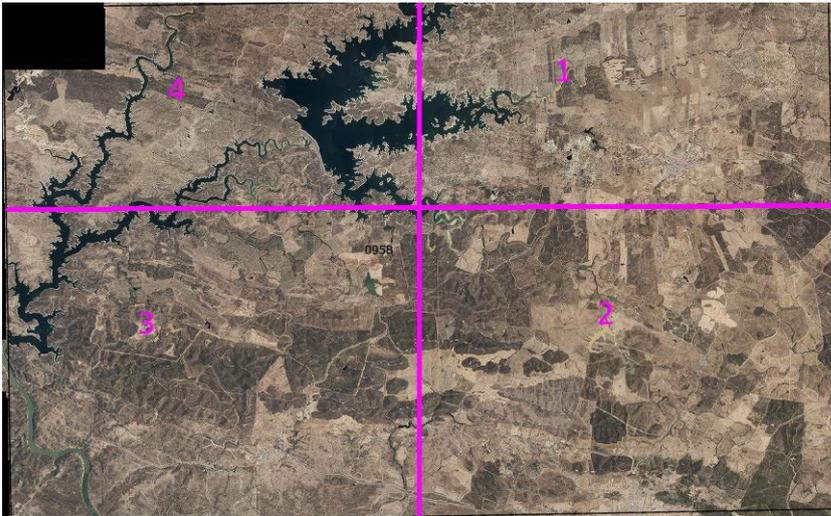


Figura 56: división de la hoja 0958 por cuadrantes. Elaboración propia.

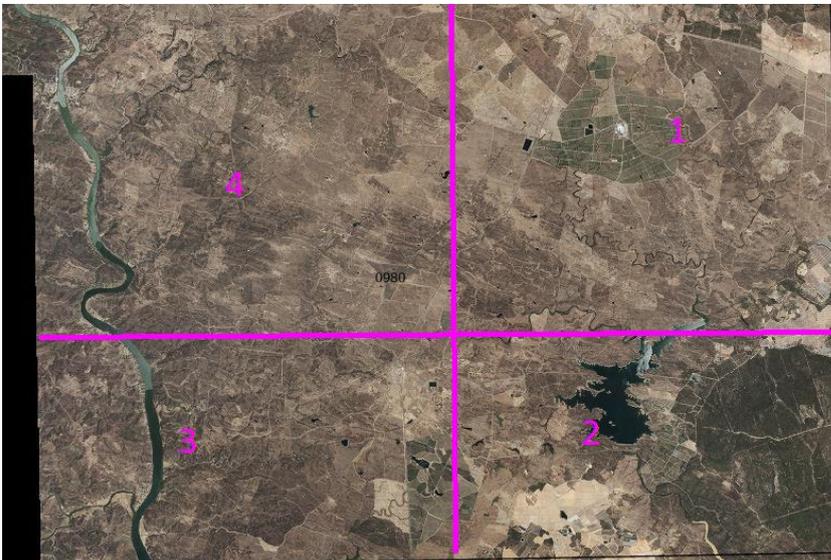


Figura 57: división de la hoja 0980 por cuadrantes. Elaboración propia.

Tras explorar el terreno de estas dos hojas y con ayuda de la localización de los parques eólicos, se ha llegado a la conclusión de que la búsqueda de aerogeneradores se va a realizar en los cuadrantes 3 y 2 de la hoja 0958 y en el cuadrante 3 de la hoja 0980.

#### Búsqueda en hoja PNOA\_MA\_OF\_ETRS89\_HU29\_h50\_0958:

##### Cuadrante 3

La capa generada contiene 38 elementos. Se han comprobado estos elementos y de los 38, 5 de ellos pertenecen a zonas del terreno o a otro elemento que no se corresponde con un aerogenerador. Esto supone un porcentaje de acierto del 86.84%. sin embargo, al hacer zoom de las zonas localizadas, se ha observado que algunos aerogeneradores no se han detectado con la herramienta.

Por ejemplo, en la [imagen 58](#) podemos ver que 3 elementos de este tipo no se han localizado.



Figura 58: elementos sin localizar. Elaboración propia.

Vamos a realizar de nuevo la búsqueda para ver si al tratarse de una zona de menos extensión, el programa es capaz de detectarlos.

En esta nueva detección de objetos ([imagen 59](#)), la herramienta ha vuelto a localizar los mismos elementos y los tres que anteriormente no había detectado, sigue sin hacerlo.

Esto puede deberse a que el número de imágenes y etiquetas sea escaso y la herramienta necesite más de 100 muestras para lograr un correcto funcionamiento.

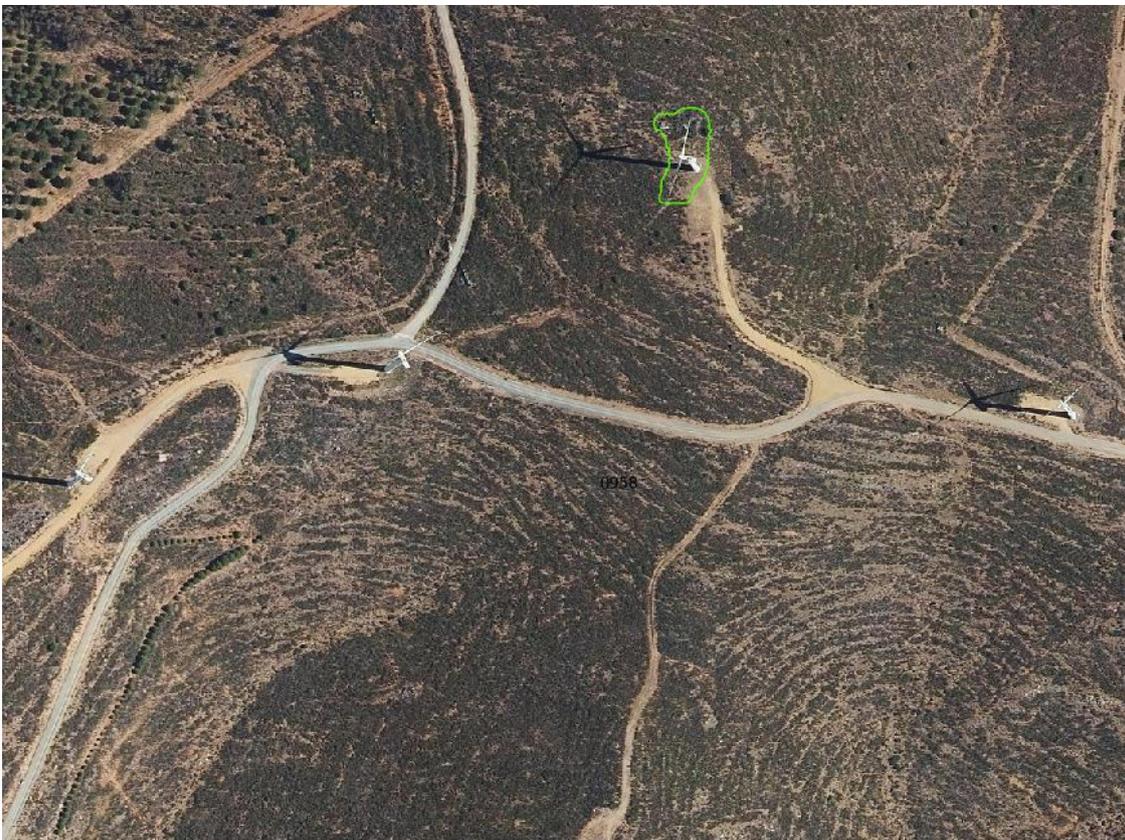


Figura 59: elementos din localizar tras realizar un segundo intento. Elaboración propia.

No obstante, si sumamos estos tres aerogeneradores no detectados a la suma del total de la capa de los localizados tendemos 41 elementos de los cuales 8 han sido los encontrados erróneamente o los no localizados, lo que da un acierto de la herramienta del 80,49%.

## Cuadrante 2

Tras la detección de objetos de este cuadrante, la capa resultante contiene 119 elementos. De nuevo se han comprobado estos elementos y el resultado es que 16 de esos objetos, no se corresponden a aerogeneradores. En la [figura 60](#), se muestra un ejemplo de algunos objetos que la herramienta ha interpretado según los parámetros establecidos de las etiquetas generadas.



*Figura 60: Ejemplo de falsos positivos en la detección de objetos. Elaboración propia.*

Se puede ver que en esa zona ([figura 60](#)) existe una edificación junto a los aerogeneradores, probablemente se trate de la central de transformación eléctrica o similar. La herramienta de detección ha interpretado que el tejado de ese edificio puede corresponder a las aspas de un aerogenerador.

No obstante, si hacemos el cálculo tenemos que, de 119, 16 han sido detecciones fallidas, lo que da un 86,55% de acierto en la búsqueda.

Por otro lado, se ha realizado la búsqueda manual de la zona para ver si ha ocurrido lo mismo que con el cuadrante 3 y no se han detectado elementos de viento sin identificar. En esta ocasión parece que la herramienta los ha detectado en su totalidad.

Búsqueda en hoja PNOA\_MA\_OF\_ETRS89\_HU29\_h50\_0980:

### Cuadrante 3

Se ha realizado la misma comprobación que en los casos anteriores y el resultado obtenido ha sido una detección de 52 elementos y de esos 9 han sido falsos positivos y 1 falso negativo, lo que conlleva a un 80,76% de acierto en la detección de aerogeneradores. (figura 61)



Figura 61: resultados de búsqueda en el cuadrante 3 de la hoja 0980. Elaboración propia.

En total, juntando las tres búsquedas tenemos:

	VERDADERO POSITIVO	FALSO POSITIVO	VERDADERO NEGATIVO	FALSO NEGATIVO
HOJA 0958				
C3	33	5	0	3
C2	119	16	0	0
HOJA 0980				
C3	52	9	0	1
Total	204	30	0	4
Total elementos	238	Porcentaje de acierto	85,71%	

Tabla 1: Resultados obtenidos. Elaboración propia.

En cuanto a resultados devueltos por la herramienta, ArcGIS Pro genera la capa de salida con la detección de los objetos y un informe del proceso de datos. En este informe se indica la imagen sobre la que se ha realizado la búsqueda, la ubicación de la capa de salida, el modelo de

entrenamiento que se ha utilizado y los parámetros que se han establecido en la detección de objetos. (Figura 63).

Por otro lado, la capa generada contiene cuatro atributos, el identificador, el tipo de capa, la clase o nombre del objeto y un campo llamado Confidence. En este último campo, se puede ver el porcentaje de confianza de la detección del objeto (figura 62).

FID	Shape	Class	Confidence
0	Polígono	AEROGENERADORES	98.794377
1	Polígono	AEROGENERADORES	99.48895
2	Polígono	AEROGENERADORES	99.803585
3	Polígono	AEROGENERADORES	99.650311

Figura 62: Atributos de la capa generada. Elaboración propia

Sin embargo, este programa no devuelve un análisis estadístico de datos ni de los cálculos o iteraciones que se han llevado a cabo durante el proceso de detección.

Iniciado: miércoles, 4 de agosto de 2021 22:28:08  
 Completado: miércoles, 4 de agosto de 2021 23:53:11  
 Tiempo transcurrido: 1 Hora 25 Minutos 3 Segundos

> Errores y advertencias

▼ Parámetros

Input Raster	PNOA_MA_OF_ETRS89_HU29_h50_0980.ecw
Output Detected Objects	D:\TFM\PRUEBA_100_IMAGENES\HOJA_0980\DETECCION \DETECCION_AEROGENERADORES.shp
Model Definition	D:\TFM\PRUEBA_100_IMAGENES\MODELO_ENTRENAMIENTO \MODELO_ENTRENAMIENTO.emd
Arguments	padding 56;batch_size 1;threshold 0.9;return_bboxes False
Non Maximum Suppression	NMS
Confidence Score Field	Confidence
Class Value Field	Class
Max Overlap Ratio	0
Processing Mode	PROCESS_AS_MOSAICKED_IMAGE

▼ Entornos

Extensión	640119.294854171 4132742.25662239 646530.075067154 4137367.615018
Tipo de procesador	GPU

> Mensajes

Figura 63: informe del proceso. Elaboración propia.

Se ha comprobado que la carpeta generada para la capa de salida no contenga algún informe de este tipo y efectivamente, en esta carpeta únicamente están los archivos complementarios a la capa generada. (Figura 64).

D:) > TFM > PRUEBA\_100\_IMAGENES > DETECCION

Nombre	Fecha de modificac
DETECCION.CPG	03/08/2021 10:46
DETECCION.dbf	03/08/2021 10:46
DETECCION.prj	03/08/2021 10:46
DETECCION.shp	03/08/2021 10:46
DETECCION.shx	03/08/2021 10:46

Figura 64: contenido de la carpeta generada. Elaboración propia.

### Aplicación de los resultados

Los resultados obtenidos se podrían aplicar de diferentes maneras.

En primer lugar, se podrían agrupar todas las capas generadas en una sola y así tener todos los datos recopilados en un solo documento o capa.

Otra aplicación sería generar una capa puntual partiendo de la capa superficial que se ha creado al realizar la detección. Recordemos que la capa creada tras la utilización de la herramienta, engloba el objeto localizado, pero no genera coordenadas de él. Por lo tanto, esta sería una posible aplicación y modificación de los resultados iniciales.

En el supuesto caso de que esto lo realizara algún organismo oficial y obtuviera estos resultados, se podría poner en contacto con el parque eólico en concreto y demostrar los resultados obtenidos para así pedir información acerca de todos los elementos que contenga, como pueden ser sus coordenadas y las alturas de los elementos.

## 7. CONCLUSIONES

### Conclusión en cuanto al objetivo principal del proyecto

En líneas generales, podemos decir que el objetivo del proyecto se ha cumplido. Se ha conseguido realizar la localización de los aerogeneradores.

No obstante, hemos encontrado algunos obstáculos o inconvenientes que se podrían mejorar a la hora de querer conseguir una mayor precisión de la herramienta o abarcar espacios de mayor extensión.

En primer lugar, las características del hardware empleado no han estado a la altura de lo que la herramienta exige para poder lograr un rendimiento mayor. Eso ha limitado el procedimiento de algunos puntos de la herramienta.

La elección de las 100 imágenes ha resultado ser escasa para la creación del modelo, a mayor número de patrones de muestra, mayor será el rendimiento del Deep Learning, con 300 o 400 imágenes la detección habría conseguido una mayor precisión en la búsqueda de aerogeneradores, sin embargo, con la limitación del hardware que comentamos, esta selección no ha sido posible llevarla a cabo. No obstante, se puede valorar como positivo el porcentaje de

acierto alcanzado, que ha sido del 85,71% ([tabla 1](#)). Con esta mejora del modelo y del hardware, se podría llegar a alcanzar el 99% mencionado en el estado del arte. ([figura 18](#))

Tras las primeras pruebas realizadas para ver y entender el funcionamiento de la herramienta, se originó un problema a la hora de la detección de objetos; caminos similares al color y forma de los aerogeneradores, techos de edificios con líneas blancas, invernaderos y demás objetos similares, también fueron detectados e interpretados como aerogeneradores. Este problema se consiguió reducir al cambiar el parámetro de desplazamiento para que en la creación de etiquetas no se desplazara y cogiera el terreno de alrededor del objeto en cuestión. Con este cambio, se pudo ver que la búsqueda mejoró en el porcentaje de acierto.

Por otro lado, la idea inicial del proyecto era realizar las búsquedas en las ortofotos del PNOA que están divididas por hojas del MTN50. Esas hojas han resultado tener un tamaño excesivo, la mayoría están en los 8GB de peso, esto ha repercutido a la hora de detectar los objetos deseados. Se realizaron varios intentos de búsqueda en las hojas completas y ArcGIS Pro en la mayoría de los casos se quedaba sin funcionar y en los casos que funcionó, tardó un tiempo muy elevado, este tiempo oscilaba entre las 18 y 25 horas de procesamiento.

La solución a este problema ocasionado ha sido la de dividir las hojas por cuadrantes, siendo efectiva la búsqueda en cuanto a tiempo y resultados. Sin embargo, ha seguido teniendo lugar el “error” comentado anteriormente, eso se solucionaría incrementando el número de imágenes a la hora de crear el modelo de entrenamiento.

Sin embargo, este problema es posible que siempre aparezca, aunque sea en menor porcentaje, pero que no se consiga reducir a cero. El Deep Learning está “despegando” y sin duda aún tiene mucho margen de mejora en cuanto a efectividad. No obstante, aquí entra la importancia del “ojo humano” para la interpretación y el análisis de los resultados, es decir, aún con el programa mejorado y el modelo entrenado con múltiples etiquetas, se puede dar el caso de que siga interpretando un camino como aspas del molino de viento, ya que la gama de color y la forma siempre van a ser similares.

La tecnología nos ayuda a mejorar el trabajo en cuanto a tiempo y eficacia, pero teniendo siempre en cuenta que la interpretación de los resultados tiene que ser supervisada, analizada y mejorada por una persona encargada de ello.

#### Conclusión en cuanto a objetivos secundarios.

Uno de los objetivos secundarios que teníamos era el de familiarizarnos con el Programa ArcGIS Pro y el funcionamiento de la herramienta Deep Learning. En esta ocasión, considero que los objetivos también se han conseguido. Se han logrado comprender los parámetros de las herramientas y en qué afectan a la hora de la detección de objetos. Sí es cierto que existen parámetros para nivel avanzado, pero al tratarse del primer encuentro con esta metodología, se han explicado y utilizado los parámetros básicos para que el modelo y la detección funcionaran lo mejor posible.

En cuanto a conocer las normativas AESA y OACI, se puede decir que se ha conseguido plasmar las normas básicas de navegación y obstáculos, que era lo que interesaba para el proyecto. Estas normativas son muy extensas, pero se ha conseguido una buena síntesis para que los lectores del proyecto entiendan en qué se basa la detección de los aerogeneradores y porqué es importante que estén geolocalizados.

Por último, la aplicación o el destino de los resultados obtenidos, desde mi punto de vista, pueden ser de gran importancia para organismos o empresas dedicadas a este sector. En primer lugar, el visor INSIGNIA, que debería tener representados todos los aerogeneradores, hemos visto que carece de algunos de éstos, por lo que se podría informar de este proyecto, o de la herramienta para que consiguiera tener actualizado su visor. Por otro lado, veo importante también la comunicación con los distintos parques eólicos para que conozcan su situación y la actualicen. Por último, el Ejército del Aire, el Ministerio de Defensa, Aena, podrían beneficiarse también de estos datos obtenidos y así no depender de otros organismos para tener actualizados sus datos de navegación.

### Conclusión del Deep Learning

Como se ha comentado en el estado del arte del presente trabajo, la inteligencia Artificial y todos los procesos que derivan de ésta tuvieron sus impedimentos en el pasado debido a la precariedad de las máquinas de aquel entonces. Sin embargo, en la actualidad, del Deep Learning está en constante avance y mejora, se puede decir que está en su primer periodo de vida y que aún queda mucho por avanzar, mejorar y desarrollar, sin embargo, ya se está viendo que puede ser una forma de trabajo más que efectiva y que sin duda sustituirá en un futuro otras formas de trabajo que se van quedando más obsoletas.

## GLOSARIO DE TÉRMINOS

[t1]: Transfer Learning: Es un método de aprendizaje automático que basa su entrenamiento en tomar como punto de partida otros modelos entrenados previamente, librándose así de comenzar de cero en cada modelo.

[t2]: Resnet34: Artículo, “A Transfer Residual Neural Network Based on ResNet-34 for Detection of Wood Knot Defects” para entender mejor el funcionamiento de ResNet-34. Fuente: <https://www.mdpi.com/1999-4907/12/2/212>

[t2]: CV-Tricks. ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of Convolutional Networks – CV-Tricks.com. Disponible en: <https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/>.

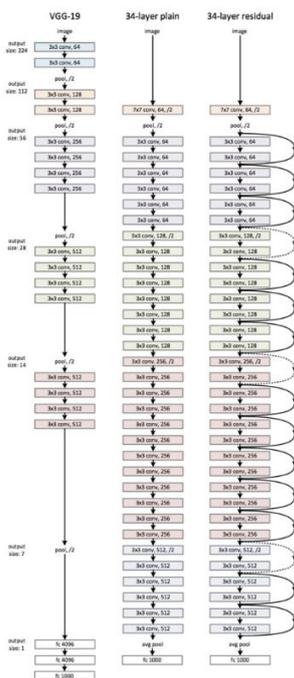


Figura 65: Arquitectura ResNet-34. Fuente: <https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/>.

[t3]: Blacken Around Feature: este parámetro se encarga de determinar si los píxeles pueden sufrir cambios, como oscurecerse alrededor del objeto que se quiere estudiar. Este tipo de formato únicamente funciona si se elige el etiquetado de teselas.

<https://pro.arcgis.com/es/pro-app/latest/help/analysis/image-analyst/label-objects-for-deep-learning.htm>

[t4]: Crop Mode: este parámetro se encarga de determinar si las teselas deben ser recortadas a la hora de exportarse para que todas tengan el mismo tamaño. Este formato únicamente funciona si se elige el etiquetado de teselas.

<https://pro.arcgis.com/es/pro-app/latest/help/analysis/image-analyst/label-objects-for-deep-learning.htm>

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Evolucion de la Inteligencia Artificial. <https://empresas.blogthinkbig.com/una-breve-historia-del-machine-learning/>
- [2] Application of deep learning to the diagnosis of cervical lymph node metastasis from thyroid cancer with CT: external validation and clinical utility for resident training. Fuente: <https://busqueda.bvsalud.org/portal/resource/pt/mdl-32065285?src=similardocs>
- [3] Machine Learning en la detección de fraudes de comercio electrónico aplicado a los servicios bancarios. Autor, Fredi Alvarez. Fuente: <https://dspace.palermo.edu/ojs/index.php/cyt/article/view/4310/6143>
- [4] Deep learning for self-driving cars: chances and challenges. Authors: Qing Rao, Jelena Frtunikj, Authors Info & Affiliations. Fuente: <https://dl.acm.org/doi/abs/10.1145/3194085.3194087>
- [5] Facial Recognition Using Deep Learning. Autor: Neelabh Shanker Singh. Fuente: [https://link.springer.com/chapter/10.1007/978-981-15-0372-6\\_30](https://link.springer.com/chapter/10.1007/978-981-15-0372-6_30)
- [6] Principales Algoritmos usados en Machine Learning. Fuente: <https://www.aprendemachinlearning.com/principales-algoritmos-usados-en-machine-learning/>
- [7] Satellite Image Classification with Deep Learning. Autores, Mark Pritt y Gary Chern. Fuente: <https://ieeexplore.ieee.org/abstract/document/8457969>
- [8] Guía de señalamiento e iluminación de turbinas y parques eólicos. Autor, AESA (Agencia Estatal de Seguridad Aérea). Fuente: <https://www.aeeolica.org/uploads/documents/3005-guia-de-senalamiento-e-iluminacion-de-turbinas-y-parques-eolicos-aesa.pdf>
- [9] Convenio sobre Aviación Civil Internacional. Autor, OACI (Organismo de Aviación Civil Internacional). Fuente: <https://www.anac.gov.ar/anac/web/uploads/normativa/anexos-oaci/anexo-14-vol-i.pdf>
- [10] Guía de aprendizaje profundo. Fuente: <https://www.aprendemachinlearning.com/aprendizaje-profundo-una-guia-rapida/>
- [11] Labels objects for deep learnin. Fuente: <https://pro.arcgis.com/es/pro-app/2.6/help/analysis/image-analyst/label-objects-for-deep-learning.htm>
- [12] Arquitectura de red. Análisis de mascara R-CNN. Fuente: <https://programmerclick.com/article/7448220979/>
- [13] Una Revisión Sistemática de Métodos para Localizar Automáticamente Objetos en Imágenes. Autor: Deisy Chaves. Fuente: <https://pdfs.semanticscholar.org/3346/3ed1f39185347ba0b46e7e6b1a8734e0b79d.pdf>
- [14] Servicio Wep Map de la Ortofoto del PNOA de España. Fuente: <http://www.ign.es/wms-inspire/pnoa-ma?request=GetCapabilities&service=WMS>
- [15] Servicio de descargas del CNIG. Fuente: <https://centrodedescargas.cnig.es/CentroDescargas/index.jsp>
- [16] Visor de navegación aérea del ministerio de Transportes, Movilidad y Agenda Urbana. Fuente: <https://insignia.enaire.es/>

[17] The Effectiveness of Data Augmentation in Image Classification using Deep Learning. Autor: Jason Wang y Luis Perez. Fuente: <https://arxiv.org/abs/1712.04621>

**REFERENCIAS DE IMÁGENES**

Figura 1: Artículo De Revista Mapping. La inspección en vuelo en el CECAF. Fuente: [http://revistamapping.com/wp-content/uploads/2018/03/Revista-MAPPING-185\\_A1.pdf](http://revistamapping.com/wp-content/uploads/2018/03/Revista-MAPPING-185_A1.pdf) .....8

Figura 2: Proteccioncivil.es <https://www.proteccioncivil.es/catalogo/carpeta02/carpeta24/vademecum17/vade01.htm> .....9

Figura 3: <https://insignia.enaire.es/> .....9

Figura 4: <https://insignia.enaire.es/> ..... 10

Figura 5: <https://insignia.enaire.es/> ..... 10

Figura 6: <https://www.futurespace.es/machine-learning-los-origenes-y-la-evolucion/> ..... 13

Figura 7: <https://www.futurespace.es/machine-learning-los-origenes-y-la-evolucion/> ..... 13

Figura 8: <https://pro.arcgis.com/es/pro-app/latest/help/analysis/deep-learning/what-is-deep-learning-.htm>..... 15

Figura 9: <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/> ..... 15

Figura 10: <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/> ..... 16

Figura 11: <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/> ..... 16

Figura 12: <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/> ..... 16

Figura 13: <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/> ..... 17

Figura 14: <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/> ..... 17

Figura 15: <https://www.aprendemachinelearning.com/principales-algoritmos-usados-en-machine-learning/> ..... 17

Figura 17: Fuente: <https://ieeexplore.ieee.org/abstract/document/8457969/authors#authors> ..... 19

Figura 18: <https://ieeexplore.ieee.org/abstract/document/8457969/authors#authors> ..... 19

Figura 22: <https://www.aprendemachinelearning.com/aprendizaje-profundo-una-guia-rapida/> ..... 24

Figura 30: <http://repositorio.unsa.edu.pe/bitstream/handle/20.500.12773/12247/ISnichw.pdf?sequence=1&isAllowed=y>..... 30

Figura 31: <http://repositorio.unsa.edu.pe/bitstream/handle/20.500.12773/12247/ISnichw.pdf?sequence=1&isAllowed=y>..... 30

Figura 32: <http://repositorio.unsa.edu.pe/bitstream/handle/20.500.12773/12247/ISnichw.pdf?sequence=1&isAllowed=y>..... 30

Figura 43: <https://archive.ics.uci.edu/ml/index.php> ..... 39

Figura 65: <https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/> ..... 55

## BIBLIOGRAFÍA

- Ahmed Ali, Mohammed Al-Saffar.  
**Review of deep convolution neural network in image classification.**  
Último acceso: 29/08/2021.  
Fuente: <https://ieeexplore.ieee.org/document/8253139/authors#authors>
- Alvarez Fredi.  
**Machine Learning en la detección de fraudes de comercio electrónico aplicado a los servicios bancarios.**  
Último acceso: 29/08/2021.  
Fuente: <https://dSPACE.palermo.edu/ojs/index.php/cyt/article/view/4310/6143>
- Atienza Arroyo, Adrián.  
**Detección e identificación automática de actrices y actores mediante el uso de algoritmos de Deep Learning.**  
Último acceso: 01/09/2021.  
Fuente: <https://uvadoc.uva.es/bitstream/handle/10324/38351/TFGG3622.pdf?sequence=1&isAllowed=y>
- Belal, Ahmed.  
**Image splicing detection using mask-RCNN.**  
Último acceso: 03/09/2021.  
Fuente: <https://link.springer.com/article/10.1007/s11760-020-01636-0>
- Campos-Taberner, M.  
**Deep learning para la clasificación de usos de suelo agrícola con Sentinel-2.**  
Último acceso: 01/09/2021.  
Fuente: <https://polipapers.upv.es/index.php/raet/article/view/13337/13212>
- **Cartas aeronáuticas.**  
Último acceso: 28/08/2021.  
Fuente:  
<https://www.icao.int/NACC/Documents/Meetings/2012/PBNGNSS2012Workshop/4-1%20%20PBN%20Aeronautical%20Charts%20--AIM--.pdf>
- Chaves, Deisy. Revista Iberoamericana.  
**Una Revisión Sistemática de Métodos para Localizar automáticamente Objetos en Imágenes.**  
Último acceso: 03/09/2021.  
Fuente:  
<https://pdfs.semanticscholar.org/3346/3ed1f39185347ba0b46e7e6b1a8734e0b79d.pdf>

- Choquehuayta, Wilder Nina.  
**Detección de embarcaciones utilizando deep learning e imágenes satelitales ópticas.**  
Último acceso: 02/09/2021.  
Fuente:  
<http://repositorio.unsa.edu.pe/bitstream/handle/20.500.12773/12247/ISnichw.pdf?sequence=1&isAllowed=y>
- **Convolutional neural networks are fantastic for visual recognition tasks.**  
Fuente: <https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/>
- **El transfer learning y las redes convolucionales.**  
Último acceso: 26/08/2021.  
Fuente: <https://www.viewnext.com/transfer-learning-y-redes-convolucionales/>
- Fernandez, Eduardo A.M.  
**Water Tanks and Swimming Pools Detection in Satellite Images: Exploiting Shallow and Deep-Based Strategies.**  
Último acceso: 01/09/2021.  
Fuente: <https://sol.sbc.org.br/index.php/wvc/article/view/13491/13339>
- Galarza Bravo, Michelle.  
**Detección de peatones en la noche usando faster r-cnn e imágenes infrarrojas.**  
Último acceso: 01/09/2021.  
Fuente: [http://scielo.senescyt.gob.ec/scielo.php?script=sci\\_arttext&pid=S1390-860X2018000200048](http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-860X2018000200048)
- Gonzalez Pacheco, Victor.  
**Una breve Historia del Machine Learning.**  
Último acceso: 27/08/2021.  
Fuente: <https://empresas.blogthinkbig.com/una-breve-historia-del-machine-learning/>
- K.N.R. Surya Vara Prasad.  
**A Downscaled Faster-RCNN Framework for Signal Detection and Time-Frequency Localization in Wideband RF Systems.**  
Último acceso: 29/08/2021.  
Fuente:  
[https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9075413&casa\\_token=IOwKLIs6ffMAAAAA:89JhPdp7jGMzgfDIJcFpYRHdvmtUxNeOrWt9iQDFZMStRMpWjRq63zLcuSBC85TeCNKYOS46g](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9075413&casa_token=IOwKLIs6ffMAAAAA:89JhPdp7jGMzgfDIJcFpYRHdvmtUxNeOrWt9iQDFZMStRMpWjRq63zLcuSBC85TeCNKYOS46g)
- Mask R-CNN. Fuente: <https://programmerclick.com/article/96221511727/>

- Teba Hermoso, Carlos.  
**Sistema Basado en Deep Learning para la identificación de Estilos de Cómicos.**  
Último acceso: 27/09/2021.  
Fuente: <http://tauja.ujaen.es/bitstream/10953.1/14199/3/TFG-CarlosTebaHermoso.pdf>
- Yang, Zhuo.  
**Investigation of Deep Learning for Real-Time Melt Pool Classification in Additive Manufacturing.**  
Último acceso: 02/09/2021.  
Fuente: <https://ieeexplore.ieee.org/abstract/document/8843291/authors#authors>

- ESRI:

<https://desktop.arcgis.com/es/arcmap/latest/tools/spatial-analyst-toolbox/export-training-data-for-deep-learning.htm>

<https://desktop.arcgis.com/es/arcmap/10.3/map/map-export-and-print/exporting-your-map.htm>

<https://doc.arcgis.com/en/imagery/workflows/standard-workflow/preparing-your-data/data-sources-and-formats.htm>

<https://pro.arcgis.com/es/pro-app/latest/help/analysis/deep-learning/what-is-deep-learning.htm>

<https://pro.arcgis.com/es/pro-app/latest/help/analysis/deep-learning/deep-learning-in-arcgis-pro.htm>

<https://blog.esri.es/entrada-de-blog/consejos-y-trucos-para-trabajar-con-el-servicio-sentinel-2-views-parte-iv/>

Instalador bibliotecas aprendizaje profundo: [https://github.com/Esri/deep-learning-frameworks/blob/master/README.md?rmedium=links\\_esri\\_com\\_b\\_d&rsource=https%3A%2F%2Flinks.esri.com%2Fdeep-learning-framework-install](https://github.com/Esri/deep-learning-frameworks/blob/master/README.md?rmedium=links_esri_com_b_d&rsource=https%3A%2F%2Flinks.esri.com%2Fdeep-learning-framework-install)

## ANEXOS

### 1. Contenido del documento generado “esri\_accumulated\_estats.json”

```
1 {
2   "Version" : 1,
3   "NumBands" : 3,
4   "TileSizeX" : 256,
5   "TileSizeY" : 256,
6   "NumClasses" : 1,
7   "NumTiles" : 25,
8   "OutputFeatures" : true,
9   "MetaDataMode" : "RCNN_Masks",
10  "Classes" : [
11    {
12      "ClassValue" : 1,
13      "ClassName" : "molinos"
14    }
15  ],
16  "FeatureStats" : {
17    "NumClasses" : 1,
18    "NumImagesTotal" : 25,
19    "NumFeaturesTotal" : 25,
20    "NumImagesPerClass" : [
21      25
22    ],
23    "NumFeaturesPerClass" : [
24      25
25    ],
26    "NumFeaturesPerImage" : {
27      "Min" : 1,
28      "Max" : 1,
29      "Mean" : 1,
30      "Sum" : 25,
31      "Count" : 25
32    },
33    "FeatureAreaPerClass" : [
34      {
35        "Min" : 4.5740634367240869,
36        "Max" : 329.44562041284917,
37        "Mean" : 140.83483403410688,
38        "Sum" : 3520.870850852672,
39        "Count" : 25
40      }
41    ]
42  }
43 }
```

### 2. Contenido del documento generado “esri\_model\_definition.emd”

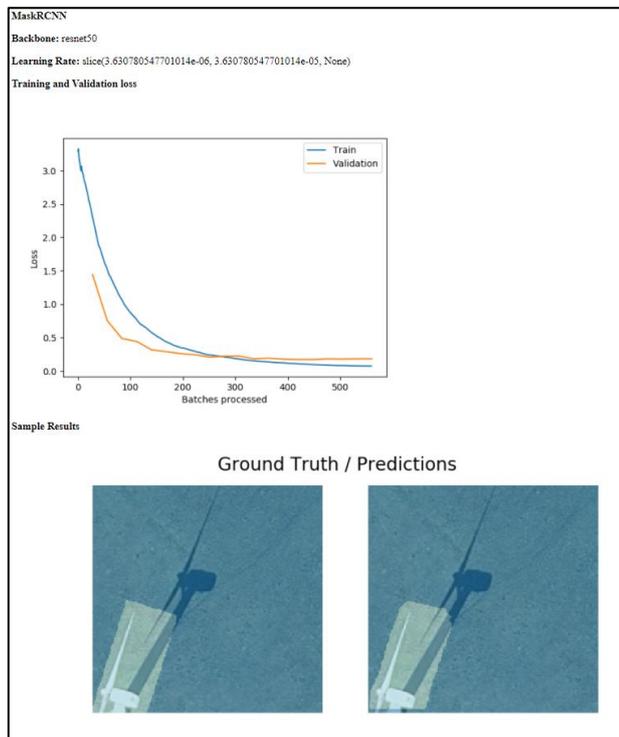
```
1 {
2   "Framework" : "e.g. TensorFlow or PyTorch",
3   "ModelConfiguration" : "e.g. UNet or FasterRCNN",
4   "ModelType" : "e.g. ObjectDetection or ImageClassification",
5   "ModelFile" : "e.g. ./trained.model",
6   "Description" : "e.g. which data model was trained on, when, by whom, mAP, etc",
7   "ImageHeight" : 256,
8   "ImageWidth" : 256,
9   "ExtractBands" : "e.g. [0, 1, 2]",
10  "DataRange" : "e.g. [0.1, 1.0] (optional)",
11  "ModelPadding" : "e.g. 64 (optional)",
12  "BatchSize" : "e.g. 8 (optional)",
13  "PerProcessGPUMemoryFraction" : "e.g. 0.8 (optional)",
14  "MetaDataMode" : "RCNN_Masks",
15  "ImageSpaceUsed" : "MAP_SPACE",
16  "Classes" : [
17    {
18      "Value" : 1,
19      "Name" : "molinos",
20      "Color" : [
21        255,
22        0,
23        0
24      ]
25    }
26  ],
27  "WellKnownBandNames (FYI, these band names can be used in ExtractBands)" : [
28    "Red",
29    "Green",
30    "Blue",
31    "Infrared",
32    "CoastalBlue",
33    "Yellow",
34    "RedEdge",
35    "ShortWaveInfrared",
36    "NearInfrared",
37    "NearInfrared_1",
38    "NearInfrared_2",
39    "MidInfrared",
40    "MidInfrared_1",
41    "MidInfrared_2",
42    "Thermal",
43    "Thermal_1",
44    "Thermal_2",
45    "Panchromatic",
46    "PseudoRed",
47    "PseudoGreen",
48    "PseudoBlue"
49  ]
50 }
```

### 3. Contenido de la carpeta models.

#### 3.1. ArcGISImstabcDetector.py: es un código generado por la herramienta.

```
1
2 import json
3 import sys, os, importlib
4 sys.path.append(os.path.dirname(__file__))
5
6 import numpy as np
7 import math
8 import arcpy
9
10 def get_centroid(polygon):
11     polygon = np.array(polygon)
12     return [polygon[:, 0].mean(), polygon[:, 1].mean()]
13
14 def check_centroid_in_center(centroid, start_x, start_y, chip_sz, padding):
15     return ((centroid[1] >= (start_y + padding)) and (centroid[1] <= (start_y + (chip_sz - padding)))) and
16
17 def find_i_j(centroid, n_rows, n_cols, chip_sz, padding, filter_detections):
18     for i in range(n_rows):
19         for j in range(n_cols):
20             start_x = i * chip_sz
21             start_y = j * chip_sz
22
23             if (centroid[1] > (start_y)) and (centroid[1] < (start_y + (chip_sz))) and (centroid[0] > (start_x)) and (centroid[0] < (start_x + (chip_sz))) and (centroid[1] >= (start_y + padding)) and (centroid[1] <= (start_y + (chip_sz - padding))) and (centroid[0] >= (start_x + padding)) and (centroid[0] <= (start_x + (chip_sz - padding))):
24                 in_center = check_centroid_in_center(centroid, start_x, start_y, chip_sz, padding)
25                 if filter_detections:
26                     if in_center:
27                         return i, j, in_center
28                 else:
29                     return i, j, in_center
30     return None
31
32 def get_available_device(max_memory=0.8):
33     """
34     select available device based on the memory utilization status of the device
35     :param max_memory: the maximum memory utilization ratio that is considered available
36     :return: GPU id that is available, -1 means no GPU is available/uses CPU, if GPUtil package is not installed, will
37     """
38     try:
39         import GPUtil
40     except ModuleNotFoundError:
41         return 0
42
43     GPUs = GPUtil.getGPUs()
44     freeMemory = 0
45     available=-1
46     for GPU in GPUs:
47         if GPU.memoryUtil > max_memory:
48             continue
49         if GPU.memoryFree >= freeMemory:
50             freeMemory = GPU.memoryFree
51             available = GPU.id
52
53
```

#### 3.2. Model\_metrics.html



3.3. ModelCharacteristics: contiene la misma información que el apartado 3.2. de anexos, pero en formato png.

