# WINDMILLS DETECTION USING DEEP LEARNING ON SENTINEL SATELLITE IMAGES

M. Tertre, T. Laurençot \*

Thales Six Gts France, Image Intelligence Service, Vélizy, France thierry.laurencot@thalesgroup.com

#### Commission III, WG III/1

KEY WORDS: Windmills, Object detection, Sentinel, Satellite image, Deep Learning.

### **ABSTRACT:**

Automatic detection of objects from Earth Observation images is still a challenge for researchers. This paper aims at extracting automatically windmills on mid-resolution images (10-meter resolution), based on Sentinel satellite products. Sentinel-2 optical images are obvious candidates for our study. At 10-meter resolution, a windmill is represented with only a few pixels. We also start to consider Synthetic Aperture Radar (SAR) Sentinel-1 images but no particular windmill radar response on GRD (Ground Range Multi-look Detected) products seemed to be relevant.

Considering the maturity of deep learning techniques for object detection in computer vision, we explore the use of deep neural networks for windmill detection on remote sensing images. For that purpose, we had to create the training data sets but we took advantage of the availability of many Sentinel images and of the use of automated labelling as the objects are georeferenced. The proposed approach relies on the U-Net framework, reformulating our problem of object detection in terms of semantic segmentation. We trained several neural networks on different data sets emanating from different countries. That enabled us to measure the performance of detection within a country but also across two countries (training on a country and predicting on another country). The results show the ability of detection of such small objects with respect to the resolution and we obtain various levels of performances depending on the trained and test data sets.

### 1. INTRODUCTION

Remote sensing enables the monitoring of the evolution of objects on Earth. In the last years, there has been various improvement in the satellite domain: increase of the spatial resolution, increase of the temporal revisit and wider range of spectral bands. The resulting large amount of data has led to the challenge of extracting automatically specific objects, in order to help monitor our environment for example. This paper focuses on the detection and localization of windmills. The interest in such objects is driven by aeronautical needs for navigation and by energy transition purpose for estimating renewable wind energy production.

Based on the characteristics that windmills are in small area in small quantity with respect to a remote sensing image, (Li et al., 2018) proposed a method using SVM (Support Vector Machines) and morphological attribute filters for the detection of windmills. More recently, (Mridula, Sharma, 2021) Learning techniques experimented Deep and used Convolutional Neural Network (CNN) to detect such objects on high-resolution optical images. Concerning offshore maritime targets detection (including windmills), (Bentes et al., 2018) tested different models of CNN on high resolution SAR images. Most of their Neural Networks outperformed their CFAR (Constant False Alarm Rate) target detector considered as a baseline. (Wang et al., 2019) described a new method to detect and mark azimuth ambiguities automatically in high-resolution SAR images, using Single Shot multibox Detector (SSD) and took into account windmills.

Adapted to that kind of problem of object detection, our approach relies on Deep Learning techniques but our main goal is to work with mid-resolution images as the ones provided by the Sentinel satellites. The choice of such images is that they cover globally a large part of Earth, allowing us to use the method we developed possibly everywhere on Earth. The Sentinel images are also freely available and provide a large amount of data, a feature that is of utmost importance when using Deep Learning techniques. (Kruitwagen et al., 2021) illustrated the use of Sentinel images for the detection of photovoltaic solar energy generating units worldwide.

Within the Sentinel family, the two constellations with a resolution that matches the scale for objects detection like windmills are the Sentinel-1 and Sentinel-2 constellations.

The Sentinel-1 comprises SAR imaging satellites. Among the various products available, the Full Resolution GRD product acquired in Stripmap Mode seems a good candidate with a resolution of about 9 meters and a pixel spacing of about 3.5 meters (Bourgibot et al., 2016). Unfortunately, there are no such type of image products on Europe. We next turned to Full Resolution GRD products acquired in Interferometric Wide Swath Mode, with a resolution of about 20 meters and a pixel spacing of 10 meters. However, it was very challenging to try to recognize SAR particular pattern response of windmills. We thus decided not to use such images (see Appendix for more details).

The Sentinel-2 comprises 13-band multispectral optical imaging satellites, with a best spatial resolution of 10 meter. Among all the spectral bands available, we selected the ones having a 10-meter resolution, thus ending with four bands in the visible (Blue, Green and Red) and Near-Infrared spectrum (see Table 1).

<sup>\*</sup> Corresponding author

Band	Wavelength	Resolution
B2	490 nm	10 m
B3	560 nm	10 m
B4	665 nm	10 m
B8	842 nm	10 m

Table 1. Sentinel-2 Bands chosen.

The Sentinel-2 products used in this study are the L2-A orthoimages displaying bottom-of-atmosphere reflectances. Figure 1 shows how windmills appear on Sentinel-2 images compared to how it appears on a very high-resolution image.



**Figure 1**. One windmill on a VHR image (Geoportail © IGN) at full resolution and six aligned windmills on a Sentinel-2 image at full resolution.

The study presented in this paper explores the possibility to detect and localize windmills on Sentinel-2 images using CNN. By creating data sets on different countries, we aim at analysing the behaviour of a neural network trained on a country data set and tested on another country data set. The next chapter describes the neural networks method we used for windmills detection. The third chapter deals with the creation of various data sets in order to train (and evaluate) CNN. The fourth chapter presents the results and a discussion on the results. The final chapter summarizes the outcome of our analysis and describes some perspectives for future research.

#### 2. METHOD

Among image deep learning techniques, we may distinguish the following four different classes of problems:

- Image Classification: the aim is to classify the main object category within an image.

- Object Detection: the aim is to identify the object category and locate the position using a bounding box for every known object within an image.

- Semantic Segmentation: the aim is to identify the object category of each pixel for every known object within an image. Labels are class-aware.

- Instance Segmentation: the aim is to identify each object instance of each pixel for every known object within an image. Labels are instance-aware.

The detection of windmills lies more within the Object Detection class. However, in previous work, we experimented the use of semantic segmentation for buildings detection. The results for extracting specific types of spatially isolated buildings (storage tanks for example) were quite promising. We thus perform this study with neural networks specifically designed for semantic segmentation.

#### 2.1 Neural Network for semantic segmentation

Semantic segmentation consists in determining, for each pixel, the type of objects it belongs to, with the aim to group them together in regions in order to create a partition of the image. In semantic segmentation, the most effective neural networks are the convolutional neural networks (CNN), completely built upon convolution layers. The output objective of such an algorithm is to obtain a mask of segmentation of the same size of the input image: a naïve approach is to apply a succession of layers of convolutions with an increasing number of filters while keeping the same dimensions of the input image. In practice, it is not possible because of its computational cost and an architecture of type auto-encoder is preferred. This one consists of two parts:

- the encoder : it is a classical CNN, used in most problem of understanding scenes. It is constituted by a succession of layers of convolution with an increasing number of filters and possibly layers of pooling for down-sampling, allowing to reduce the dimensions of the image while increasing the number of channels.

- the decoder : it is an inverse network to the encoder. Its purpose is to increase the dimensions of the image while decreasing the number of channels. Nevertheless, it is also a network of convolution constituted by a succession of layers of convolutions with a decreasing number of filters and possibly by layers of up-sampling.

A problem appears with such a simple architecture: by decreasing the dimension of the image in the encoder to increase the number of channels, we lose spatial information to gain semantic information. Now, when the resolution is reincreased in the decoder, the spatial information is only partially recovered and there is thus a lack of precision in the reconstruction of the mask of segmentation. The cue is to combine these different pieces of information by establishing connections between the layers at the bottom and the top levels.

It is with this in mind that has be designed the U-Net architecture (Ronneberger et al., 2015). It is a network initially developed to perform segmentation of cancer cells on images from microscopes. However, it quickly became a reference in any problem of segmentation thanks to its performances. The main idea is to copy the outcome of several layers of the encoder and to concatenate them to those of the same dimensions in the decoder. It enables the combination of the spatial and semantic information in the same tensor and leads to an architecture in the form of U illustrated on Figure 2.



Figure 2. U-Net architecture (Ronneberger et al., 2015).

### 2.2 U-Net implementation

Generally, the weights of a neural network are initialized randomly (according to very specific normal laws) before being updated during the training. In particular, the computational cost of adjusting correctly and appropriately all these weights can be very high. To remedy this problem, a common use in Deep Learning is the transfer learning, that is the re-use of the weights of a network pre-trained on one (or more) task (s) and on a particular data set, in a different problem. Typically, a network of classification is trained on ImageNet data: the subset called ImageNet Large Scale Visual Recognition Challenge (ILSVRC) contains more than one million of images annotated and is divided in about 1 000 object classes (Russakovsky et al., 2015). Such a network is generally constituted by a succession of layers of convolution and subsequently followed by some fully connected layers at the end to establish the prediction. It is thus possible to re-use these layers of convolutions (typically, the fully connected layers are not kept) by inserting them into our network, and to make a new training on our data set only. There are here several possibilities: for instance, training the complete network. Alternatively, one can train the non-pretrained layers and use several learning rate (to decrease for example the value of the learning rate for the re-used layers in order to slightly modify them).

The general idea is that the first layers have learned features of very low level, common to any image, as the detection of edges, lines or spots, while the last layers are more specific to the dedicated problem and would logically be modified in a more consequent way. This technique is very interesting because it improves generally the results of the current problem, as it roughly means increasing the size of the data set of training. Furthermore, it is very easy to insert a pre-trained encoder in a network of U-Net type, with the distinction made between encoder and decoder.

In this study, we replace the basic encoder of U-Net by various neural networks pre-trained on ImageNet: VGG16, VGG19 (Simonyan et al., 2015) and ResNet50 (He et al., 2016). Figure 3 shows the architecture of a VGG-16 neural network.



Figure 3. VGG-16 Architecture.

In our U-Net framework, we use the binary cross entropy loss function in the training process to generate the output map. The output is thus a probability belonging to [0,1]. For the purpose of detection, a threshold is then applied to the output map. Usually, the standard value used is 0.5.



Figure 4. U-Net probability map output (left), results with a threshold of 0.5 (middle), detection results overlaid on the original image (right).

Figure 4 shows the output map generated by the U-Net neural network (left), the results after using a threshold of 0.5 (middle) and the bounding box of the windmills classified as True positive (green), False negative (red) and False positive (yellow) (right).

In our U-Net framework, we introduce Batch Normalization and use the Adam optimizer. We also set up early stopping for the loss function on the validation dataset. We base our work on the Keras Tensorflow implementation.

## 2.3 Evaluation

In order to evaluate the results of our neural networks, we rely on the usual following deep learning metrics.

The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly). The precision measures the model's accuracy in classifying a sample as positive.

$$Precision = \frac{\text{True Positive}}{(\text{True Positive+False Positive})}$$
(1)

The recall is calculated as the ratio between the number of Positive samples correctly classified to the total number of Positive samples. The recall measures the model's ability to detect Positive samples. The higher the recall, the more positive samples detected.

$$Recall = \frac{True Positive}{(True Positive+False Negative)}$$
(2)

Usually, precision and recall scores are provided together and are not quoted individually. Still, if a single number is required to describe the performance of a model, the most convenient is the Dice coefficient also known as F1 Score, which is the harmonic mean of the precision and the recall.

$$F_1 \text{ score} = 2 \cdot \frac{\text{Precision . Recall}}{\text{Precision + Recall}}$$
(3)

We use these three metrics in our study.

### 3. DATA SETS

As for every deep learning approach, there is a need for a labelled dataset of remote sensing images containing windmills. Having found none existing one, we need to create our own data set: it is not a problem to get Sentinel-2 images but more challenging to get a valid list of georeferenced windmills. Moreover, we want to have data sets on different countries for out study. In the end, we build three data sets: one for France, one for Spain and the last one for Germany and the Netherlands.

### 3.1 Windmill ground truth localisation

Searching for the location of windmills for our first area of interest that is France, we find a first list of windmills on the open data site data.gouv.fr. Unfortunately, after beginning to work with it, it appears that it cannot be trusted and the quality of the data varies enormously. Searching for other sources, we finally decide to use the OpenStreetMap (OSM) data. The advantage of such data is that there are updated continuously by the community. Windmills in the OSM database are described as points or polygons. In order to have a homogeneous database, we transform the polygons into points by taking their centroids. As the reliability in the ground truth is very important in training to avoid creating bias, we make an analysis on a sample of data to check their freshness and their geographic accuracy. It appears that some windmills are prematurely spotted in the OSM data, being ongoing projects and even not yet built. Concerning geographic accuracy, one concern is about localisation, and we contributed a little to refine the localisation of some windmills; the other concern is the choice of the spatial coordinates taken to model a volumetric windmill object made of a mast, a turbine and pales. As shown in Figure 5 on a highresolution image, the points extracted are roughly representing the middle of the mast of the windmill at its bottom. Figure 6 shows the same points on a Sentinel-2 image.



Figure 5. OSM windmills point overlaid on a Geoportail image.



Figure 6. OSM windmills point overlaid on a Sentinel-2 image.

Using the Overpass Turbo API enables us to extract these specific windmill objects from the whole OSM database. We had to make some processing to reduce polygon-described windmill to their centroid and concatenate them with the point-described windmills. We end up with a list of about 8 000 windmills for France.

One of the advantages of using OSM data is that the tools implemented for creating the list of windmills for the data set over France can be used to obtain those of Spain and of Germany and the Netherlands. We retrieve about 20 000 windmills in Spain, 29 000 windmills in Germany and 2 500 in the Netherlands. Figure 7 displays the location of the windmills extracted from the OSM database on a cartographic basemap.





### 3.2 Creation of the windmill image data sets

The image data sets for the deep learning neural networks are created from Sentinel-2 optical images with 10-meter resolution. Having the list of position of windmills, we are able to download from the Copernicus DataHub Sentinel-2 images whose footprints intersect the position of the windmills. We use the DHuSget API to download the images. The size of each image is 10 980 by 10 980 pixels. As previously mentioned, we are interested in the 10-meter products of the Sentinel-2 images and work either with each band to make a three channel product or rely directly on the True Colour Image file product.

Using U-Net framework, we need to create a mask to classify the points between windmill and not windmill. As windmills are described as points, we consider arbitrary a region around that point with two different sizes: 4 by 4 pixels encompassing at least the mast, part of the pales and the turbine and 6 by 6 pixels encompassing the shadow if any.

We tested several strategies to create our data sets. The first simple one is to generate patches of images centred on a windmill to be sure to have one. Unfortunately, it gives bias in the learning step. Another strategy is to extract patches from a regular grid leading to many patches of image without windmills. We finally use the strategy depicted on Figure 8 to create the different image data sets.



Figure 8. Image data set creation process.

The process described above enables us to provide images containing at least one windmill but often containing more as several windmills are usually gathered together. As stated by the state-of-the-art in training for deep learning, we also add images with no windmill.

After having made a first small data set and verified that the process of creation was successful, we increase our data sets by exploiting Sentinel-2 images within one year long in order to get images in all seasons. For example, for France, we download about 500 images with no maritime areas and with less than 1% of cloud cover. Figure 9 shows some of the downloaded images. As easily seen, in the L2 products, the effective usable portion of the images has to be taken into account (and the black parts have to be eliminated).



Figure 9. Mosaic of a subset of the Sentinel-2 images downloaded.

We end with a data set of about 60 000 images for France, a data set of about 38 000 images for Spain and of about 43 000 for Germany and the Netherlands

# 4. RESULTS

In order to evaluate the performance of our neural networks, we adopt the following process to identify a true or a false detection: given a region segmented coming from the thresholding of the Neural Network output map, we compute its centroid and measure the distance of that centroid to the ground truth. If the distance is less than three pixels, the detection is labelled as true and false otherwise. This value can be considered as high but it takes into account that a windmill location is modelled as a point in the ground truth database

### 4.1 France Area of Interest

This is our first area of interest. We adopt an incremental approach and increase the data set more and more with a variety of dates of acquisition (about one year at the end) to achieve a high F1 score. We end up with a data set of about 60 000 images. As usual for such methods, we divide that data set into three distinct data sets to have 75% (thus about 50 000) images for training, 20% (thus about 8 000) images for validation and the last 5% (about 2 000) images for test. Using the VGG16 and trying two different sizes of boxes for delineating windmills (4 by 4 pixels and 6 by 6 pixels), we achieve the following results using a threshold of the output map of 0.5:

Size of box	Precision	Recall	F1 score
4 by 4	0.97	0.93	0.95
6 by 6	0.96	0.93	0.95

Table 2. France data set results.

We obtain good results with a F1 score of 0.95 and we can handle windmills on various background and across seasonality. In the test data set, as described earlier, there is usually at least one windmill. We also find little difference in the use of the different sizes of boxes delineating the windmills. Figure 10 shows results with true positives (green boxes), whereas Figure 11 shows results with true positives (green), false positives (yellow) and false negatives (red). The right image shows that our dataset still contains some inconstancy between the image and the OSM database (on temporal aspect for example).



Figure 10. Good results of detection.



Figure 11. Limited results of detection.

# 4.2 Spain Area of interest

Using the same method as for France, we train dedicated neural networks on a Spain data set. Similarly to France, we use the same ratio (75%-20%-5%) to divide the Spain data set of about 40 000 images, thus having 30 000 images for training, 8 000 images for validation and 2 000 images for test. Using the same parameters, we obtain a F1 score of 0.93 for box size of 4 by 4 and 0.90 for box size of 6 by 6, thus leading to similar conclusion as from France.

We also compute the performance of the two neural networks learned on the France data set on the Spain test data set. We get a F1 score of 0.62 for box size of 4 by 4 and 0.60 for box size of 6 by 6, thus leading to a significant decrease of the performance that can certainly be attributed to the difference in windmills environment in Spain compared to that in France.

# 4.3 Germany/The Netherlands Area of interest

The data set made on Germany and the Netherlands contains about the same number of images as the one on Spain. Again, we train dedicated neural networks on this data set. Using the same parameters, we get a F1 score of 0.85 for box size of 4 by 4 and 0.84 for box size of 6 by 6. The overall results are not as good as for the previous data set but are still independent of the choice of the box size delineating the windmill.

We also compute the performance of the two neural networks learned on the France data set on that test data set. We obtain a F1 score of 0.76 for box size of 4 by 4 and 0.76 for box size of 6 by 6, having still poorer results but a smaller difference compared to Spain as the environment where the windmills are established is more similar.

### 4.4 Fine tuning

According to the above discussion, the results depend on the countries and the training data sets used. We aim at experimenting fine tuning (using transfer learning): we re-train a neural network trained on the France data set with a subset of the training data set of Spain and compute again the performance on that re-trained neural network on the Spain test data set. The result improves significantly and reaches a good F1 score of 0.94. We repeat a similar test with the Germany/The Netherlands case and obtain similar results. Table 3 summarizes those results:

Training data set	Test data set	F1 score
Spain	Spain	0.93
France	Spain	0.62
France then Spain subset	Spain	0.94
Germany/The Netherlands	Germany - The	0.85
	Netherlands	
France	Germany - The	0.76
	Netherlands	
France then Germany -	Germany - The	0.84
The Netherlands subset	Netherlands	

Table 3. Transfer learning results.

#### 4.5 U-Net Architecture dependency

As presented in chapter 2, we can use various pre-trained neural networks in the U-Net architecture. The results are not very different from one to the other. To illustrate it, Figure 12 shows the evolution of the loss function of the training data set and the validation data set during training when using VGG19 and ResNET50 on the German/The Netherlands data set.



Figure 12. Loss function value (with logarithm scale) versus epoch

### 4.6 Sentinel-2 bands used

Among the available bands of Sentinel-2 images, one is the Near Infrared band (B08). We took advantage of this band and projected several ways to use it:

- integrate it in a deep learning approach on its own. However, windmills are poorly distinguishable in that band.

- make a three-channel combination with two of the three others visible bands. We started making such data sets but the first obtained results were disappointing.

- modify the neural network in order to take four channels as input. Based on the results on the second point, we did not go into that direction. Moreover, that would have removed the possibility to take pre-trained networks in the U-Net architecture.

We thus did not continue in trying to use that band in our deep learning approach.

#### 5. CONCLUSION

We described in this paper our approach for detecting windmills on Sentinel-2 images based on the use of a U-Net architecture. Although a windmill on a Sentinel-2 image represents only few pixels, the results show that we can achieved good performance for the detection of such objects. Having creating different data sets on different countries, it seems that windmills are not so similar from one data set to the other. However, using fine tuning, we show that re-training a network learned on a data set of country with a subset of a data set of another country can increase significantly the performance of detection. However, further investigations should be conducted on the content of the data sets (number of images, proportion of images with windmills and without, size of the subset for fine tuning, ...). This paper also describes a method to build a data set of Sentinel-2 images that could be used for other objects. We

Sentinel-2 images that could be used for other objects. We applied the same process for creating a data set for water towers in France. The performance of the neural network trained on that data set led to a F1 score of 0.63. Figure 13 shows an example of results.



Figure 13. Example of water towers detected in France.

### REFERENCES

Bentes, C., Velotto, D., Tings, B., 2018: Ship classification in TerraSAR-X images with convolutional neural networks. *IEEE Journal of Oceanic Engineering*, vol. 43, no. 1, 258-266. doi.org/10.1109/JOE.2017.2767106.

Bourbigot, M., Hussez, R., Piantanida, R., Hadjuch, G., Poullaouec, J., 2016: Sentinel-1 Product Definition. Ref.: S1-RS-MDA-52-7440.

He, K., Zhang, X., Ren, S., Sun, J., 2016: Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*. 770-778. doi.org/10.1109/CVPR.2016.90.

Kruitwagen, L., Story, K. T., Friedrich, J., Byers, L., Skillman, S., Hepburn, C., 2021: A global inventory of photovoltaic solar

energy generating units. *Nature*, 598, 604-610. doi.org:/10.1038/s41586-021-03957-7.

Li, H., Zhao J., Zhang Y. and Zhang Y., 2018: Recognition of windmills in remote sensing image by SVM and morphological attribute filters. *IEEE International Geoscience and Remote Sensing Symposium*, 6923-6926. doi.org/10.1109/IGARSS.2018.8519274.

Maayan, F-A., Ben-Cohen, A., Amer, R., Greenspan, H., 2018: Improving the segmentation of anatomical structures in chest radiographs using U-Net with an ImageNet pre-trained encoder. *Third International Workshop, RAMBO 2018, BIA 2018, TIA 2018.* Lecture Notes in Computer Science, vol. 11040. Springer, Cham. doi.org/10.1007/978-3-030-00946-5\_17.

Mridula, A., Sharma S., 2021: WMDeepConvNets Windmill detection using deep learning from satellite images. *Data Science and Intelligent Applications*. Lecture Notes on Data Engineering and Communications Technologies, vol. 52. Springer, Singapore. doi.org/10.1007/978-981-15-4474-3\_19.

Ronneberger, O., Fischer, P., Brox, T., 2015: U-Net: convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention*. Lecture Notes in Computer Science, vol. 9351. Springer, Cham. doi.org/10.1007/978-3-319-24574-4\_28.

Russakovsky, O., Deng, J., Su, H. et al., 2015: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision 115*, 211–252. doi.org/10.1007/s11263-015-0816-y.

Simonyan, K., Zisserman, A., 2015: Very deep convolutional networks for large-scale image recognition. *The 3rd International Conference on Learning Representations*. doi.org/10.48550/arXiv.1409.1556.

Wang, J., Chen, J., Wang, P., Men, Z., Pan, 2019: An algorithm for azimuth ambiguities detection on high resolution SAR images using SSD. *IEEE International Conference on Signal, Information and Data Processing*,1-5. doi.org/10.1109/ICSIDP47821.2019.9173356.

### APPENDIX

The following figures shows a part of a Full Resolution GRD Sentinel-1 product acquired in Interferometric Wide Swath Mode. It is the same area at the same date: the first image is in VV-polarisation mode and the second one is in VH-polarisation mode. The ground-truth places of windmills are represented with cyan circles. Knowing that there are some windmills, we can recognize some of them on the VH-polarised image (two white points localised near the circles) but none on the VVpolarised image. However, this kind of behaviour cannot be considered to be reliable, as it is sometimes possible to recognize windmills on VV-polarised image but not on VHpolarised image. Moreover, in some other cases, none of the polarised images shows a particular pattern of windmills. Without making a deeper study, that analysis led to give up trying to use such images for detecting windmills.



Figure 14. Ground truth spots of windmills on a VV-polarised IW GRD Sentinel-1 image



Figure 15. Ground truth spots of windmills on a VH-polarised IW GRD Sentinel-1 image