All of the articles bellow are discussing Ship detection using satellite imagery with Machine learning models and AI – our purpose is to provide beneficial information in order to conduct research on the ship detection topic:

Article 1:

Name of Article: Ship detection on Sentinel-2 images with Mask R-CNN model

Author: Andrea C.

A time analysis of maritime traffic using PyTorch and open data

As part of a larger ML project, we decided to explore the possibility to assess maritime traffic using publicly available satellite images. In particular, our goal was to estimate a time series that is representative of the volume of the maritime traffic observed over time in a given region. In this article, we discuss the methodology and results obtained.

Note: all code discussed in the following is available on my personal GitHub at https://github.com/andrea-ci/s2-ship-detection.

Why satellite data

Photo by Matthijs van Heerikhuize on Unsplash

Satellite's gone up to the skies

Things like that drive me out of my mind

I watched it for a little while

I like to watch things on TV

Lou Reed, "Satellite of love"

In the last years, remote sensing has dramatically evolved and so have other fields such as computer vision and machine learning. In addition, more and more satellite imagery has become publicly available. Notable examples include Landsat and Sentinel-2 constellation imagery.

Satellites exhibit three fundamental advantages for the users:

they allow access to information that is often difficult to obtain by other means;

they provide a global geographic coverage;

they collect information periodically and with high temporal resolution (the so-called revisit period).

Moreover, the spatial resolution of images, i.e. the ability to differentiate two close objects, is constantly increasing, thus enabling more and more new applications. A famous example is given by market research firms [1] that recently have exploited satellite imagery to count cars in parking, in order to estimate retail demand. In principle, similar methods can be used also for applications with social and administrative purposes, such as measuring urban traffic or counting crowds at political rallies.

Sentinel-2 mission

In this experiment, we consider the ESA Sentinel-2 mission. Sentinel-2 satellites are equipped with a Multi-Spectral Instrument (MSI) that collects data using 13 bands at 3 different resolutions: 10, 20, and 60 meters. In order to have the best available resolution, we have worked with B02, B03, and B04 bands, which are located in the visible spectrum and provide a resolution of 10 meters.

Sentinel-2 bands with resolution of 10 meters—Source: https://sentinels.copernicus.eu

Sentinel-2 images can be accessed in different ways, including:

Copernicus Open Access Hub, a website from ESA which provides free and open access to Sentinel-1, Sentinel-2, Sentinel-3, and Sentinel-5P user products;

the Registry of Open Data, from Amazon AWS;

Sentinel HUB, a Cloud API for satellite imagery for which also a convenient Python library, sentinelhub, is available.

Modeling approach

Detecting ships on images is a hard task because by construction the number of positive samples (i.e. pixels belonging to a ship) is extremely small compared to the number of negative ones. Thus, instead of attempting a semantic classification task, I chose to go for an object detection approach.

Processing workflow—image by author

As a baseline, a pre-trained Mask R-CNN model has been considered. This model adds an extra branch to the Faster R-CNN model, which in turn is based on the

architecture of Resnet, introduced in “Deep Residual Learning for Image Recognition”.

Resnet stands for Residual Network as this network introduces the concept of residual learning. Residual learning is an approach that aims to improve the performance of deep convolutional neural networks for classification and recognition tasks. Generally speaking, deep networks learn features at a low, middle, and high level through their layers. Residual networks learn instead the residuals, i.e. the difference between features, by using shortcut connections between layers. This approach has proven to make easier training, getting better accuracy values. Resnet models come in 5 variants, containing 18, 34, 50, 101, and 152 layers respectively.

For the deep learning implementation, I have used PyTorch with the Mask R-CNN model provided by TorchVision. It comes with 50 layers and is pre-trained on COCO dataset.

Now it’s time to finetune the model.

Model finetuning: data and training

For fine-tuning the model, I have used competition data from Kaggle Airbus Ship Detection Challenge.

This dataset is composed of 192556 images, of which only 42556 contain at least one ship (22% of the total). Moreover, most of them (around 60%) contain exactly one ship. Thus, the dataset is highly unbalanced and the number of positive samples is quite limited.

To make things harder, ships contained in the images can differ significantly in size and they can be located in the open sea or at docks and marinas, e.g. adjacent to the land.

Dataset comes with a CSV file where ship masks are represented by run-length encoding. Since most of the images do not contain any ships, I remove them from the dataset. For the remaining ones, masks are generated so they can be used for the creation of the model targets.

Preparing training data—image by author

Masks are integer-based 2D arrays, meaning that for any pixel x:

x=0, if it does not represent a ship;

x=1, if it is part of the first ship;

x=2, if it is part of the second ship;

and so on.

Pixel numbering is needed to identify any eventual ship included in the image because the model returns objects (ships) and each object is characterized by a unique bounding box.

The structure of the model output is clearly visible here: it consists of a dictionary with the information related to the object(s) detected by the model.

After 10 training epochs, evaluation metrics exhibit acceptable values so that training can be terminated and a model is finally ready to be applied on Sentinel-2 images.

Inference on Sentinel-2 data

For the experiment, I have focused on a small area surrounding the port of the city of Olbia, in Sardinia (Italy). This port is one of the main access points to the island for ferries coming from the Italian peninsula, especially during summer vacations.

A period included between 2017 and 2020 has been considered and two images have been acquired, at regular time intervals, for each month. As a result, the model input is composed of a sequence of 96 raw RGB images. Unfortunately, some of them resulted malformed, either because of API issues during the download or because they didn't pass the quality checks of the Sentinel-2 processing chain so that they must be discarded.

Inference on a Sentinel-2 snapshot—image by author

The sequence of images is processed by the model and detection results are saved to a CSV file: for each image, acquisition date and number of detected ships detected are reported. Also, the total area occupied by ships is included, although it is not used for this analysis.

Detection results— image by author

Analysis of results

Using Pandas it is easy to extract and visualize the time series from the CSV file. The counts reported in the file are averaged on a monthly basis so to obtain a time series of vessels observed daily in the reference month.

The time series shows a strong seasonality. Most likely this behavior is due to a couple of different reasons:

activities of the port are mainly of touristic type and the flow of tourists is larger in the summer;

conditions of weather in summer favor the acquisition of cleaner images (i.e. with low cloudiness), thus allowing a larger number of ships to be correctly detected.

Both of these factors lead to maximum traffic volumes in the summer, which then gradually decrease until they reach minimum values during the winter.

According to a local newspaper [2], in the summer season of 2020 port of Olbia has recorded a reduction in traffic of about 15% with the respect to 2019, due to pandemic restrictions that affected the whole country.

Considering the aggregated sum of the counts obtained during the summer season (i.e. between April and September, both inclusive), data obtained are consistent with this statement as they indicate a traffic reduction of about 16.28% in 2020.

Image by author

Conclusions

Satellite images in the visible spectrum are highly sensitive to weather conditions and their usage must be carefully evaluated based on the specific project requirements (e.g. region of interest, revisit period, image resolution). However, their application to extract a proxy measure for maritime activities appears, in principle, to be possible.

Article 2:

Ship Detection in Sentinel 2 Multi-Spectral Images with Self-Supervised Learning

Automatic ship detection provides an essential function towards maritime domain aware- ness for security or economic monitoring purposes. This work presents an approach for training a deep learning ship detector in Sentinel-2 multi-spectral images with few labeled examples. We design a network architecture for detecting ships with a backbone that can be pre-trained separately. By using self supervised learning, an emerging unsupervised training

procedure, we learn good features on Sentinel-2 images, without requiring labeling, to initialize our network's backbone. The full net- work is then fine-tuned to learn to detect ships in challenging settings. We evaluated this approach versus pre-training on ImageNet and versus a classical image processing pipeline. We examined the impact of variations in the self-supervised learning step and we show that in the few-shot learning setting self-supervised pre-training achieves better results than ImageNet pre-training. When enough training data are available , our self - supervised approach is as good as ImageNet pre-training . We conclude that a better design of the self-supervised task and bigger non-annotated dataset sizes can lead to surpassing ImageNet pre-training performance without any annotation costs.

1. Introduction

Ship detection is an important challenge in economic intelligence and maritime secu- rity, with applications in detecting piracy or illegal fishing and monitoring logistic chains. For now, cooperative transponders systems, such as AIS, provide ship detection and identification for maritime surveillance. However, some ships may have non-functioning transponders; many times they are turned off on purpose to hide ship movements. Mar- itime patrols can help to identify suspect ships, but this requires many resources and their range is restricted. Therefore, using satellites, such as those from the European Space Agency Sentinel-2 mission, to detect ships in littoral regions is a promising solution thanks to their large swath and high revisit time.

high-resolution

Some commercial satellite constellations offer very high resolution images (VHR) (<1 m/pixel) with low revisit time (1-2 days). However, VHR images are usually limited to the R, G, B bands and image analysis on such high resolution images is computationally intensive. On the other hand, synthetic aperture radar (SAR) satellites can also be used, although their resolution is lower than VHR optical sources (e.g., Sentinel 1 has 5 m resolution), the analysis of their imagery is the main approach to ship detection since SAR images can be acquired irrespective of cloud cover and the day and night cycle. The downsides of SAR are low performance in rough sea conditions, but, most importantly, detection is only done on seas away from land and is not possible for moored ships in harbor or for ships smaller than 10 m [1]. Furthermore, SAR is vulnerable to jamming [2].

The Copernicus Sentinel-2 mission of the European Space Agency offers free multi- spectral images with a refresh rate of maximum 5 days and a resolution down to 10 m, as detailed in Table 1. Our work focuses on this data source for several reasons. First, multi-spectral information allows to better extract a ship fingerprint and distinguish it from land or man-made structures, as shown in [3,4]. Second, a multi-spectral optical learning based approach can perform detection in both high seas and harbor contexts, while also removing the requirement of storing a vector map of coastlines and performing cloud removal as a pre-processing step. Thus, it could be adapted to a real-time, on-board satellite setting and is not affected by jamming.

Although ship detection is a challenging task, ship identification in remote-sensing images is even more difficult [5]. A coarse identification could be made by ship type (container ship, fishing vessel, barge, cruiseliner, etc.) using supervised classification, with accuracy that should be closely related to image resolution. However, to establish ship identity uniquely, it does not seem feasible with the Sentinel-2 sensor to extract features fit for this purpose, such as measurement of ships to meter precision, extracting exact contours, or detecting salient unique traits of different ships. Our work focuses on detection but the approach is generic and could be extended to other sensors with better resolution, eventually allowing identification.

Recent remote sensing approaches based on machine learning require large amounts of annotated data. Some efforts to collect and annotate data have been made for VHR images, for SAR and for Sentinel 2, but, for the latter, these works did not target ship detection in particular. For object detection using convolutional neural networks (CNN), an interesting way to overcome the lack of data is to use transfer learning. This is achieved either by using CNNs pretrained on large labeled data sets gathered in a sufficiently "close" domain (such

as digital photographs), or by pretraining a neural network on the satellite image domain. The latter can be done through an unsupervised pipeline using self-supervised learning (SSL) [6], a contrastive learning paradigm that extracts useful patterns, learns invariances and disentangles causal factors in the training data. Features learned this way are better adapted for transfer learning of few-shot object detectors. We propose to use this paradigm to create a ship detector with few data.

For VHR images, a large amount of literature exists, with the number of works follow- ing the increasing number of sensors and the quantity of publicly available data [7,8]. Many of these approaches focused on detecting ships with classical image processing pipelines: image processing using spectral indices or histograms (e.g., sea-land segmentation, cloud removal), ship candidate extraction (e.g., threshold, anomaly detection, saliency), and, then, rule-based ship identification or classification using statistical methods. Virtually all of these works focus on VHR images with R,G,B, and PAN bands, occasionally with the addition of NIR, with resolution less than 5 m. Deep learning was applied to images with under lm resolution by using object detection convolutional neural networks (CNN): R-CNNs [9,10], YOLO [11,12], U-Net

For SAR imagery, [1] reviews four operational ship detectors that work on multiple sensors. All of the approaches use classical processing chains and start by filtering out land pixels. This filter is either based on shapefiles or on land/water segmentation masks generated from the SAR image. However, in both cases, a large margin is taken around the coastlines, eliminating any ships that are moored in ports. Deep learning was also applied to SAR ship detection, with notable results detailed in [15].

In multi-spectral images, the most notable work is [4] which uses SVMs to identify water, cloud, and land pixels and then builds a CNN to fuse multiple spectral channels. This fusion network predicts whether objects in the water are ships. Other approaches, such as [3], rely on hand made rules on size and spectral values to distinguish between ships, clouds, islands, and icebergs. The only Sentinel 2 ship dataset publicly available is [16] but it only includes small size image chips and weak annotations for precise localization,

1. e., a single point for each ship, obtained by geo-referencing AIS GPS coordinates to pixel coordinates in the chips.

Although large datasets exist for VHR images, for Sentinel-2 none are available with pixel level annotations while usually thousands of examples are needed to train deep learning object detectors. Few-shot learning based approaches can bring interesting perspectives for remote sensing in general and in our setting in particular. Few-shot learning consists in training a neural network with few labeled samples, most often thanks to quality feature extractors upon which transfer learning is performed. One recent method for unsupervised learning of features extractors that enable few-shot learning is contrastive self-supervised learning [6,17]. Contrastive SSL relies on a "pretext training task", defined by the practitioner, that helps the network to learn invariances and latent patterns in the data [18-20].

Several strategies exist for choosing the pretext task: context prediction [21], jigsaw puzzle, or simply by considering various augmented views. The latter is used by [22,23] for remote sensing applications like land use classification and change detection.

Contributions

In this work, we make two contributions:

(1) A deep learning pipeline for ship detection with few training examples. We take advantage of self-supervised learning to learn features on large non-annotated datasets of Sentinel 2 images and we learn a ship detector using few-shot transfer learning;

(2) A novel Sentinel 2 ship detection dataset, with 16 images of harbours with a total of 1053 ship annotations at the pixel level.

2. Materials and Methods

Our approach is based on a U-Net architecture with a ResNet-50 backbone to produce binary ship/no-ship

segmentation masks of the input image. U-Net has been used exten- sively in remote sensing applications, traditionally with a simple downsampling path of consecutive convolution blocks with no downward skip connections.

2.1. *U-Net Architecture*

Although the "vanilla" version of U-Net is usually trained from scratch, in this work we modify it to use a different backbone, ResNet-50, that can be easily pre-trained separately using a contrastive objective and then plugged into the U-Net architecture. Figure 1 describes graphically this architecture.

The network takes as input a 64 x 64 pixel patch with 6 channels corresponding to the B2 (B), B3 (G), B4 (R), B8 (NIR), and Bll and B12 (SWIR) spectral channels. The downsampling path reduces the width and height through strided convolution layers while increasing the numbers of channels. The last layer of the ResNet50 backbone has 2048 channels. A "bridge" is added between this layer and the first UPconv block of the upsampling branch of the U-Net.

Baselines

1. ImageNet transfer learning: Instead of pre-training the backbone with SSL, this baseline uses a ResNet-50 encoder pretrained on ImageNet as implemented by the torchvision package. Since these encoders are trained on RGB images, we copy the weights of the first 3 channels of the first layer in order to initialize the channels corresponding to spectral bands B8, Bll, and B12. Both the TL and FT ship detector training approach can be applied to this baseline;
2. Random initialization: Instead of using a trained backbone network, we initialize the ResNet-50 encoder randomly following the standard Kaiming initialization. Only the fine-tuning (FT) detector training mode is applied when initializing the weights randomly;
3. BL-NDWI—Water segmentation baseline with NDWI: We develop a simple baseline which is based on classical image processing techniques. We use the NDWI spectral index *NDWI* = ~~gol+BO8~~ [aR]d [we] threshold its value to segment water and non- water pixels. The threshold for the NDWI segmentation is chosen to obtain the best performance on the whole dataset, which may lead to suboptimal choices for some images.
   Next, we eliminate land pixels using the water/land segmentation (CO) (Section 2.5.1) map, giving a ship proposal map. We consider non-water pixels in what are normally water regions to potentially be ships. We extract connected components and we eliminate those that have a width and height greater than 50 pixels (500 m) since no ships larger than this size exist. These are due to islands or sandbanks not correctly mapped in OpenStreetMap layers or thin water banks where the coastline annotation in OpenStreetMap is imprecise.
   Finally, we do several filtering passes on the resulting proposal map: morphological opening and we apply watershed segmentation on the resulting map to identify individual ships.

1. Discussion

Globally, the results allow us to conclude that deep learning techniques achieve promising results. In all cases (close to shore and open-sea) recall is high, more than 75%. We obtained less than 0.14 false alerts per square kilometer in the open sea and close to the sea shore, the false alarm rate is around 1 ship/km^2. Although the BL-NDWI baseline could be improved by finding more optimal NDWI thresholds for each image, the performance difference with respect to deep learning approaches seems hard to make up for.

Networks trained with SSL achieve better results compared with ImageNet pretrained ones. In the few-shot setting, SSL pre-training is usually better and more stable. When sufficient examples are available SSL pretraining is as good as ImageNet or training from scratch. We notice also that performance increases with the size of the pre-training datasets. Since these are not annotated it is easy to

build such datasets. The ones chosen here have no relation to the ship detection problem at hand, thus no significant effort is needed to select the images in these datasets.

The pretext task needs to be chosen according to the downstream task in order to learn the needed invariances. The region based pretext task looks promising probably because it helps the encoder to better cluster together similar elements, such as water, agriculture crops, or residential areas, while a simple pretext task data-augmentation only focuses on color or noise invariances. The benefit of such pretext task can be seen in our case as it lowers the number of false positives over land and near the shore.

In terms of computational complexity, the difference between all deep learning meth- ods lies in the way we pre-train the weights. Compared to the supervised pipelines that can be used to train a ResNet-50 on ImageNet dataset, the self-supervised pipeline has a similar complexity but requires much larger batch size. This is particularly problematic for multi-spectral images, and a GPU with at least 8 Go of memory is necessary. The pretext task training is time-consuming: it took us nearly two days to train it on SEN12MS dataset using a multi-GPU machine (4 GPU with a total of 64 Go of memory). Training on a single desktop GPU with at least 8 Go of memory is feasible, but lasts longer.

Conclusions

We presented a method to train a ship detector in Sentinel-2 images using self- supervised learning. Our method plugs in a SSL-trained backbone in a U-NET architecture. It achieved better or similar results to standard deep-learning approaches and significantly better results than a spectral index based method. The choice of pretext task in the SSL stage is a major source of performance improvements.

Further studies should focus on the design of a more effective pretext task. Our work shows that there is room for improvement although the direction towards this goal remains unclear. Instead of hand-designed pretext tasks, learning a better pretext task could be a fruitful avenue of research. However, the computational cost of pre-training is high, so it would be necessary to first reduce this cost or to approximate the pre-training stage performance with a light-weight proxy model.

The SSL pipeline can be applied to networks where no ImageNet pretraining is available, such as custom architectures specific to remote sensing. Thus, an interesting research goal would be training, through SSL, a feature extractor designed for remote- sensing applications, that improves upon learning from scratch or ImageNet pretraining by a large margin.

For ship detection, our image-based approach is still two orders of magnitude away from the false alert performance of methods applied on SAR images [1]. To improve image-based ship detection, a better network backbone could be studied, more adapted to small objects. Furthermore, it would be better to include cloud filtering and land/water classification explicitly in the network. Although adding more data is always a good idea, the few-shot setting is more challenging and can bring about more methodological improvements in deep learning for remote sensing.