Oil and Gas Automatic Infrastructure Mapping: Leveraging High-Resolution Satellite Imagery through fine-tuning of object detection models

Jade Eva Guisiano, Éric Moulines, Thomas Lauvaux, Jérémie Sublime

Oil and Gas Automatic Infrastructure Mapping: Leveraging High-Resolution Satellite Imagery through fine-tuning of object detection models

Jade Eva Guisiano[1,2,3], Éric Moulines[1], Thomas Lauvaux[4], and Jérémie Sublime[2]

[1] École Polytechnique, Palaiseau, France
[2] ISEP School of Engineering, Paris, France
[3] United Nations Environment Program, Paris, France
[4] Université de Reims, Reims, France

Abstract. The oil and gas sector is the second largest anthropogenic emitter of methane, which is responsible for at least 25% of current global warming. To curb methane's contribution to climate change, emissions behavior from oil and gas infrastructure must be determined by an automated monitoring across the globe. This requires, as first step, an efficient solution to automatically detect and identify these infrastructures. In this extended study, we focus on automated identification of oil and gas infrastructure by using and comparing two types of advanced supervised object detection algorithms: Region-based Object Detector(YOLO and FASTER-RCNN) and Transformer-based Object Detector(DETR) with fine-tuning on our customized high-resolution satellite image database (Permian Basin U.S). The pre-training effect of each of these algorithms on detection results is studied and compared with non-pre-trained algorithms. The performed experiments demonstrate the general effectiveness of pre-trained YOLO v8 model with a Mean Average Precision over 90. The non-pre-trained model of this last one also over perform compare to FASTER-RCNN and DETR.

Keywords: Object detection · Remote sensing · Deep Learning · Computer vision · Oil and gas.

1 Introduction

Methane, an exceptionally potent greenhouse gas, has a much higher global warming potential than carbon dioxide, exacerbating the current climate crisis. Reducing methane emissions is an effective strategy to significantly slow the pace of global warming and its associated environmental impacts. The oil and gas industry (O&G) is a particular contributor to methane emissions, as it is the second largest anthropogenic source[1]. Methane is unintentionally released at various stages of the industry's supply chain. To effectively reduce these emissions in the O&G sector, a comprehensive understanding of the emissions profiles of individual operators, specific sites, and associated infrastructure is needed. This

knowledge would inform the formulation and refinement of regulatory measures and potential penalties to ensure they are appropriately tailored and thus optimally effective. The United Nations Environment Program has launched the

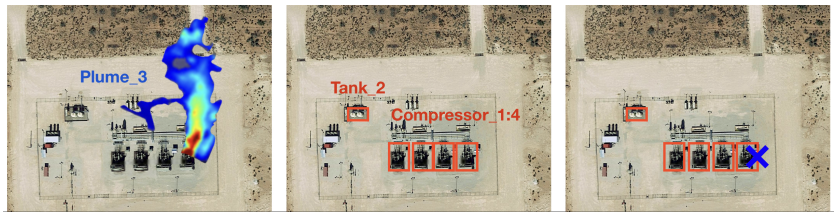

Fig. 1: Example of a detected methane plume associated with infrastructure at its source. In 3 automatic steps: detection of methane plume, detection of infrastructure, association of each plume and infrastructure. *Source: @Google Earth*

Methane Alert and Response System (MARS) program to link detected methane plumes to their specific sources. This is intended to provide operators with timely warning of discovered leaks. However, to enable near-continuous monitoring of the world's oil and gas resources, it is essential that this process be supported by an automated detection and attribution system. Certain methods such as OGNET[2] and METER-ML[3] use deep neural networks to identify specific types of oil and gas site. This study focuses on the topic of automated detection of oil and gas infrastructures a topic that has not been explored in depth in the existing literature. Sites in the oil and gas industry that contain wells, storage tanks, or compressor infrastructures are considered significant contributors to fugitive emissions and therefore form the targets we seek to automatically identify. Existing approaches to oil and gas infrastructure detection typically do not allow for the simultaneous detection of multiple infrastructures. With the goal of enabling the automatic detection of compressors, tanks, and well infrastructures simultaneously, this paper focuses on supervised object detection methods, specifically using and comparing the YOLO, FASTER-RCNN, and DETR algorithms. These algorithms, initially trained on the COCO database, are **fine-tuned** using the Oil and Gas (OG) database. The database OG, which was developed specifically for this study, contains aerial photographs with high spatial resolution (less than 1 meter). The images in the OG database are extracted from the Permian Basin, the most substantial oil and gas basin in the world, located in the states of New Mexico and Texas (US).

In the first part, the state of the art of object detection algorithms will be presented, along with their applications in the O&G sector. The OG database and its characteristics will then be detailed. Next, the YOLO and FASTER-RCNN algorithms and their parameters will be presented. Finally, the results section details the performance of each **pre-trained and non-pre-trained** algorithm.

2 State of the art

Object recognition algorithms, a subset of computer vision techniques, facilitate the automatic identification and location of multiple instances of a given class of objects in images or videos. These algorithms predominantly use either neural network-based methods or non-neural techniques. The non-neural strategies typically integrate SIFT [4] or HOG [5] (for feature extraction) with a classification algorithm such as Support Vector Machines (SVM). Despite their usefulness, recent studies suggest that neural-based object recognition methods generally outperform their non-neural counterparts[6].Neural approaches to object recognition can be divided into three categories depending on the degree of supervision in their learning process: supervised, semi-supervised, and self-supervised models[7]. In this paper, we will mainly focus on different supervised methods. Supervised object recognition models require an annotated image database for effective training. In this context, annotating an image involves identifying objects of interest by enclosing them in a bounding box and labeling them appropriately. During training, a supervised object recognition algorithm learns to locate and subsequently recognize the targeted objects. This complicated process can be executed over two primaries architectural frameworks[8]:

- **Two-stage detector:** is based on two main models, firstly Region Proposal Network (RPN) which is a fully convolutional network used to extract regions of objects, and secondly an extra model is used to classify and further refine the localization of each region proposal. RCNN[9] architecture is based on a selective search algorithm to propose regions of interest and then applies a CNN to each region to classify it as an object or background.
  As this method is particularly slow, the authors proposed Fast-RCNN[10], an optimized approach to RCNN by sharing computation across all regions proposed in an image. Finally, FASTER-RCNN[11], based on the architecture of Fast-RCNN, replaces the selective search algorithm with a RPN, which is trained to directly predict regions of interest. This latest version reduces computation time and improves the detection accuracy;
- **one stage detector:** Contrary to one stage detector, one stage detector don't need to integrate RPN to generate a region proposal, it can directly obtain the classification accuracy of the object and its coordinate position. These algorithms have the advantage of being faster than two-step algorithms. In this category we find YOLO[12] and its different versions[13], SSD [14] and RetinaNet [15]. Review studies compares the latter 3 methods, for example [16] for the pill identification task, showing that YOLO v3 offers the best performance in terms of execution time but the lowest accuracy. Another study [17] focuses on the comparison of SSD, RetinaNet, YOLO V4 and FASTER-RCNN for Tethered Balloon detection. It was show that YOLO v4 achieved the best trade-off between speed and accuracy with a precision of 90.3%. [18] also concludes that YOLO has better accuracy (increasing with version) via a broad comparison of RCNN and YOLO models and their variants;

– **Others:** There are also object detection methods based on approaches other than the one-two stage approaches detailed above. For example, DETR[19] is a transformer based detector with a 3 parts architecture constitute of a CNN, encoder-decoder transformer and a feed-forward network (FFN).

Object detection O&G applications Remote sensing object detection can be applied to a variety of problems, various studies [20] [21] [22] [23] summarizes object recognition algorithms applied to various remote sensing topics. For example, [24] summarizes the performance of FASTER-RCNN, SSD, and YOLO V3 algorithms for agricultural greenhouse detection based on high-resolution satellite imagery. [25] proposes automatic detection of earthquake-induced ground failure effects by using FASTER-RCNN. Others [26][27][28] focus on comparing one and two-stage object detection algorithms on satellite and aerial images. [29] uses DETR for object detection with enhanced multi-spectral feature extraction. In particular, object detection algorithms are also used for problems in the oil and gas sector. For example, in the work [30] [31] YOLO V4 is used to detect oil spills with Sentinel-1 SAR images. Some studies are also looking at oil and gas infrastructure detection:

– **Oil Tanks**: [32] proposes a recognition algorithm that harnesses deep environmental features, using the convolutional neural network (CNN) model and SVM classifier for oil tank recognition. Another study employs FASTER-RCNN for the same objective;
– **Oil Wells**: [33] introduces an enhanced version of YOLO v4 for detection using high-resolution images, similar to [34], where the authors utilize the faster R-CNN. [35] presents a database, dubbed Northeast Petroleum University-Oil Well Object Detection Version 1.0 (NEPU-OWOD V1.0), which includes the geographical locations of oil wells. This database was constructed via the application and comparison of nine object detection algorithms;
– **Pipelines**: In the context of pipelines, [36] utilizes a deep learning approach for object detection in underwater pipeline images, employing various YOLO configurations;
– **Oil & Gas Sites**: On a broader scale encompassing entire infrastructures, [37] employs high-resolution satellite images and YOLO V2 for automatic recognition of oil industry facilities, with a particular emphasis on well-sites.

In the field of object detection, a significant portion of existing methods are dedicated to the identification of specific infrastructures. While this focused approach proves beneficial in studies examining a single infrastructure, it may not be entirely sufficient when examining methane emissions in the oil and gas (O&G) sector. This is because such emissions can come from a variety of infrastructures. Recognising this multi-faceted challenge, this study broadens its scope to include three types of infrastructure that are essential to the O&G sector: Wells, Tanks, and Compressors. This more comprehensive approach provides a broader perspective and leads to a better understanding of the various sources of methane emissions in the sector. In addition, three different object detection algorithms are comparatively analysed in this study: YOLO v8, which follows

a single-stage detection paradigm; FASTER-RCNN, a two-stage method; and DETR, an encoder-decoder based detection approach. Each of these algorithms features a unique recognition strategy, providing a broad understanding of object recognition methods in the context of O&G infrastructures. This study also examines the impact of pre-training these models, in particular, how the pre-training phase influences detection outcomes and performance is investigated. The detailed findings from this study are presented and discussed in detail in this paper to further our understanding of the nuanced role that pre-training plays in object recognition.

3 An Oil and Gas infrastructure database

Algorithms employed in supervised object recognition necessitate a learning phase involving substantial interaction with a large repository of images. These images must be labelled with the target object in order to enable practical training. In the specific context of identifying wells, tanks, and compressors, this database must be replete with a variety of aerial photographs in which each of these objects or infrastructures is unambiguously identifiable. The procurement of such specialized labelled images, due to the lack of public availability, demanded the development of a dedicated database specifically designed for this purpose. In this study, we chose to extract high-resolution satellite

Fig. 2: Example of images and annotated objects from OG database : tank (red), compressor (purple) and well (blue)*source : @Google Earth*

images only from the Permian Basin region (over the states of New Mexico and Texas in the USA), which is the largest O&G basin in the world. 930 Google Earth images of sites with O&G infrastructures were extracted, with resolutions ranging from 15cm to 1m. Each of these images was then manually annotated by drawing bounding boxes around each well, compressor or tank present, as shown in Figure 2. Each of these boxes is associated with 1 of our 3 objects (label). In total, out of the 930 images, 1951 objects were annotated: Compressor 706 objects, Well 630 objects and Tank 615 objects All the images are in a 640x640 size format, each featuring between one and multiple instances of key infrastructure such as wells, tanks, and compressors. Another aspect worth mentioning is the special consideration given to wells in our database. Given the limited resolution of satellite imagery, it is often difficult

to discern the structural details of wells. Therefore, the recognizable shadows of wells that are present even at lower resolutions are included in the bounding boxes (as shown in the right column of Figure 2). This database is hosted on the open-source Roboflow platform and can be accessed via the following link: `https://universe.roboflow.com/thesis-ffaad/og-otgc5/dataset/6`. Following the requirements of a rigorous study design, we have divided our dataset into different subsets for training, validation, and testing. Of the total images, 80% (744 images) are used for training, 13% (or 120 images) for validation, and the remaining 7% (66 images) for testing.

4 Object Detection algorithms presentation

The structural organization of object detection algorithms is usually defined by three main components:

- **Backbone**: This refers to a deep learning architecture, usually a convolutional neural network (CNN), that is tasked with the essential function of feature extraction. Through this process, the backbone identifies and abstracts the salient features from the input data;
- **Neck**: Serving as an intermediary between the backbone and the head the neck performs a fusion of the features extracted from the different layers of the backbone model. This synthesized information forms the basis for the subsequent predictions performed by the head;
- **Head**: The head forms the final component of the object recognition model and is responsible for predicting the classes and bounding box regions. These predictions form the final output of the object recognition model. In particular, the head can produce a number of outputs, typically configured to detect objects of different sizes in an image.

Backbone Pre-training The majority of object recognition models, including but not limited to YOLO, FASTER-RCNN, and DETR, provide an option for a pre-trained version of the backbone. This pre-training generally helps to improve recognition performance. The pre-training of these algorithms is done using extensive databases of thousands of image categories, ranging from everyday objects such as airplanes and dogs to more specific objects such as apples and chairs. Prominent among these databases are ImageNet [38], which contains 200 classes and about half a million annotated objects, and COCO [39], which contains 80 classes and nearly 1 million annotated objects. In addition, the Pascal database VOC [40] includes about 20 classes with about 63,000 annotated objects. Most modern object recognition algorithms are pre-trained on the COCO dataset. A main advantage of pre-training backbones is the significant reduction in the custom dataset training phase. Pre-trained backbones that have already learned to recognize general features and patterns from large databases can transfer this knowledge to the object recognition task at hand. This not only minimizes training time, but also enables the use of smaller datasets. Pre-built models also play a critical role in mitigating the problem of over-fitting, which occurs when the

model over-learns from the training data, compromising its ability to generalize to new data. Considering these factors, the architecture of recognition algorithms can be classified into three different families: single-stage, two-stage, and other algorithms. In this study, a representative algorithm from each of these families is evaluated: YOLO v8, FASTER-RCNN, and DETR.

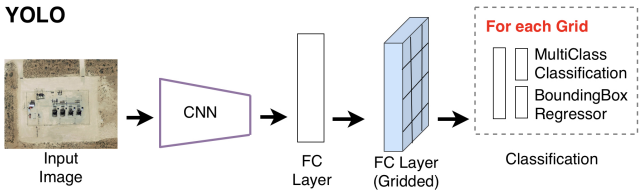

Fig. 3: **YOLO architecture:** Convolutional Neural Network (CNN), Fully Connected (FC) layer, Girdded FC layer

4.1 One stage object detector : YOLO

You Only Look Once (YOLO) [12] v8 is one of the most recent versions which outperforms previous versions in term of precision as illustrated in Figure4. YOLO v8 architecture is based on the ResNet-50 (CNN) backbone which has

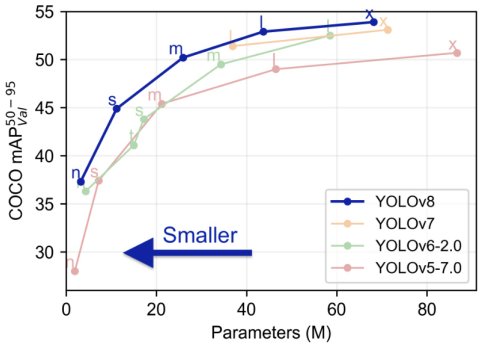

Fig. 4: YOLO Mean Average Precision (mAP) for COCO object detection by versions and models. *source :* `https://github.com/ultralytics/ultralytics`

been pre-trained on the ImageNet dataset. The ResNet-50 backbone is then fine-tuned on the COCO dataset to learn to detect objects in 80 different categories. YOLO V8 has a declination of 5 pre-trained models (n,s,m,l,x) trained on COCO 2017 dataset. These models vary according to the number of parameters

they hold directly influencing the level of precision, thus, the more parameters a model has, the better its accuracy (cf Figure 4). The 3 last pre-trained models (m,l,x) with the highest number of parameters were chosen and fine tuned with the OG database (image size 640x640), with 100 epochs, 16 batches, learning rate 0.001.

4.2 Two stage object detector : FASTER-RCNN

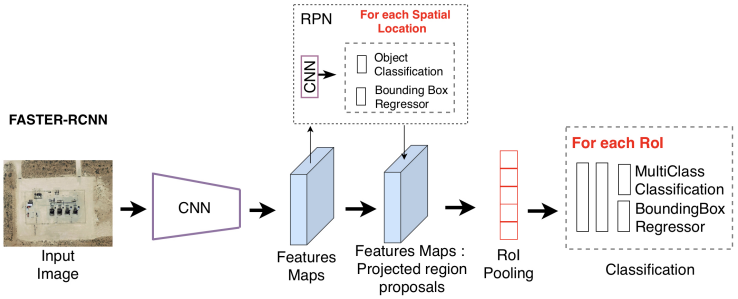

Fig. 5: **FASTER-RCNN architecture**

FASTER-RCNN [11] process in 2 main steps, first it uses use a Region Proposal Network (RPN) to generate regions of interests and secondly it send the region proposals down the pipeline for object classification and bounding-box regression.FASTER-RCNN architecture is based on 3 principal components : the backbone (CNN type varies according to chosen model), the RPN , and the ROI heads (classification and regression). FASTER RCNN provide 3 backbones architectures pre-trained on COCO 2017 base (train2017 and val2017) :

- **Feature Pyramid Network (FPN)**: Use a ResNet+FPN backbone with standard conv and FC heads for mask and box prediction;
- **C4**: Use a ResNet conv4 backbone with conv5 head which correspond to the original baseline in the Faster R-CNN paper;
- **Dilated-C5 (DC5)**: Use a ResNet conv5 backbone with dilations in conv5, and standard conv and FC heads for mask and box prediction, respectively.

We have fine-tuned 2 FPN model with ResNet50 and ResNet101, but also a DC5 model based on ResNet101. Epochs were fixed to 100, batches to 64 and learning rate to 0,001.

4.3 Encoder-decoder object detector : DETR

Unlike one-stage and two-stage detectors, DETR [19] is designed as a direct set prediction problem encompassing a unified architecture. DETR employs a backbone (with varying architecture contingent on the selected model), a transformer

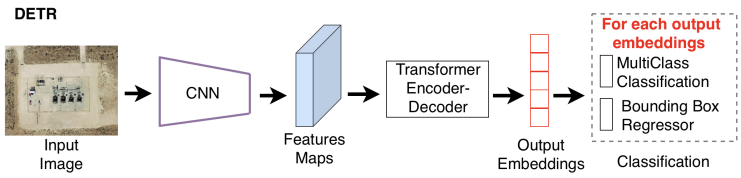

Fig. 6: **DETR architecture**

encoder-decoder architecture, and a bipartite matching between predicted and ground-truth objects. By uniting the backbone and transformer, DETR successfully simplifies the architecture by eliminating specific components to one and two-stage approaches such as anchor generation and non-maximum suppression (NMS). The following pre-trained backbone models are available, all of which have been pre-trained on the COCO 2017 database:

- **R50**: Incorporates a backbone that is based on an ImageNet pre-trained ResNet-50 model.
- **R101**: Deploys a backbone grounded in an ImageNet pre-trained ResNet-101 model.
- **R50-DC5**: Increases the feature resolution by employing dilation in the final stage of the backbone. The backbone in this model is based on ResNet-50.
- **R101-DC5**: Implements a similar process to R50-DC5 but relies on a backbone built on ResNet-101.

Pre-trained model R50, R101 and R101-DC5 were selected for test with epochs fixed to 100, batches to 2 and learning rate to 0,001.

4.4 Model evaluation

Average precision (AP) is a widely used metric for evaluating the efficiency of object recognition tasks.

The AP combines the precision and recognition curves into a single scalar quantity. The AP value ranges from 0 to 1 and tends toward 1 when both precision and recall are high, while it tends toward zero when either metric is low over a spectrum of confidence thresholds. AP is computed by calculating the difference between the current and subsequent recalls and multiplying that difference by the current precision:

$$AP = \sum_{k=0}^{k=n-1} [\text{Recalls}(k) - \text{Recalls}(k+1)] \times \text{Precisions}(k)$$

Where k is the number of object and n is the number of threshold. In addition, the mean average precision (mAP) is often used. It represents the average of AP calculated over all classes:

$$mAP = \frac{1}{n} \sum_{k=n}^{k=1} AP_k$$

where AP_k is the AP of the class k and n the number of classes.

5 Results

5.1 Algorithms and Models performance comparisons

For each algorithms (YOLO v8, FASTER-RCNN, and DETR), 3 models with different parameters and architectures were selected and compared. For each models, the output corresponds to the AP by class (Compressor, tank and well) and the mAP for the general model. The experiments were conducted with the use of a GPU NVIDIA GeForce RTX 3090 with 24 GO of memory. The experiments required the use of 3 distinct environments for the 3 algorithms with the following packages. YOLO V8 : ultralytics (Python 3.8 environment with PyTorch 1.8). FASTER-RCNN : Detectron2 with torch 1.5 and torchvision 0.6 and DETR with PyTorch 1.7 and torchvision 0.7.

Table 1: Pre-trained Algorithms Average Precision (AP) results in % on OG database. **The number of parameters is expressed in millions*

		Average Precision (AP)			
Model	**Parameters***	**Compressor**	**Tank**	**Well**	**Total**
YOLO V8					
8m	25.9	**99.5**	**98.8**	79.4	**92.6**
8l	43.7	**99.5**	88.1	**80.3**	89.3
8x	68.2	98.8	90.9	73.6	87.8
FASTER-RCNN					
R50-FPN	41.7	51.6	51.1	40.5	47.7
R101-FPN	60.6	**53.2**	**57.8**	35.4	**48.8**
R101-DC5	184.5	52.1	47.1	**42.9**	47.4
DETR					
R50	41	94.9	75.1	72.7	80.9
R101	60	**100**	**80.4**	**77.1**	**85.8**
R101-DC5	60	91.9	69.9	67.9	76.3

The table 1 show that the ensemble of mAP result (Total) of YOLO V8 model over-perform compare to those from FASTER-RCNN. DETR mAP results for each model are lower than those from YOLO V8 but are not very far. Surprisingly, YOLO v8 model (8m) with lower number of parameters performs better than the other with a higher number of parameters (which is the contrary for the COCO database as illustrated of Figure 4). It appears that for FASTER-RCNN and DETR models based on a simple Resnet101 architecture present a higher mAP compare to others and lower results with a DC5 architecture.

Compressor YOLO v8 and FASTER-RCNN present AP over 90% for compressor recognition, especially the FASTER-RCNN R101 model with has an AP of 100%. YOLO v8 8m and 8l models are also very close this last one with a 99,5% AP. On average and for compressor, YOLO v8's 3 models offer an AP of 99.3% , compared with 95.6% for DETR. On models average, YOLO v8 and FASTER-RCNN has better AP (with respectively 99,2% and 95,6%) than FASTER-RCNN (52,3%) for compressor recognition. FASTER-RCNN with lowest AP and mAP, has also on average the best results for compressor recognition.

Tank The best AP performance is obtained thanks to the YOLO v8 8m model with a score of 98,8%. On models average, YOLO has an AP of 92,6% against 75,1% for DETR and 52% for FASTER-RCNN. For tank recognition YOLO v8 largely over perform compare to others.

Well The best AP is still maintain by YOLO v8 8m model with a score of 92,6%. However, on models average for well recognition, YOLO v8 AP (89.9%) get closer to that of the DETR (81%). Indeed lose almost 10% of AP from compressor to well recognition. Concerning FASTER-RCNN, on models average, obtain its lower AP (39,6%).

5.2 Algorithms pre-training effect

In general, pre-trained models offer numerous advantages over non-pre-trained models, including the need for less data (fine-tuning) and an improvement in accuracy. The models selected for this study were trained using the COCO dataset, which consists of annotated everyday objects. However, the COCO dataset does not contain any objects from industry that could resemble the objects in the OG database. This discrepancy raises the question of the extent to which pre-trained models, originally trained on objects that are significantly different from our target objects, can still outperform the predictive accuracy of non-pre-trained models. To investigate this, the algorithms and models discussed previously were run without the weights from the pre-training phase.

According to the results delineated in Table 2, it is observed that the mean Average Precision (mAP) for the non-pre-trained YOLO v8 models is marginally lower than that for the pretrained models. Indeed, the average mAP of the YOLO v8 models is 89.4%, while that of the non-pre-trained models is 88.6%, indicating an overall decrease in mAP of 0.8%. While the difference may seem negligible, this result substantiates the assertion that pre-training YOLO v8 contributes to enhanced performance.

In compliance with the results delineated in Table 2, we observe that the mean average precision (mAP) for the non-pre-trained YOLO v8 models is marginally lower than that of the pre-trained models. In fact, the average mAP of the YOLO v8 models is 89.4%, while that of the non-pre-trained models is 88.6%, which represents an overall decrease in mAP of 0.8%. Even though the difference seems negligible, this result supports the claim that YOLO v8 pre-training contributes to improved performance.

Table 2: Non pre-trained algorithms Average Precision (AP) results in % on OG database. The empty spaces translate the non-convergence of the models and then the absence of results.

		Average Precision (AP)			
Model	**Parameters***	**Compressor**	**Tank**	**Well**	**Total**
YOLO V8					
8m	25.9	97.2	**90.6**	**81.4**	**89.7**
8l	43.7	97.7	90.4	78.0	88.7
8x	68.2	**98.1**	87.2	77.2	87.5
FASTER-RCNN					
R50-FPN	41.7	25.4	2.3	7.2	11.6
R101-FPN	60.6	—	—	—	—
R101-DC5	184.5	—	—	—	—
DETR					
R50	41	—	—	—	—
R101	60	—	—	—	—
R101-DC5	60	—	—	—	—

In terms of average precision (AP) by class, the results mirror those of the pre-trained YOLO v8 models; AP remains higher for the compressor class and lower for the well class. An interesting observation is the comparative analysis between the non-pre-trained YOLO v8 model and the pre-trained FASTER-RCNN and DETR models. The non-pre-trained YOLO v8 model outperforms all pre-trained FASTER-RCNN and DETR models in terms of mAP. This remarkable result demonstrates the superior efficacy of YOLO v8.

As regards of FASTER-RCNN, the pre-trained R50-FPN model shows significantly low AP and mAP. For the other models of FASTER-RCNN and all DETR models, convergence proved difficult even after increasing the number of iterations 20-fold and decreasing the learning rate by a factor of 1000. Non-pretrained models are notorious for their difficulty in achieving convergence, especially when dealing with smaller databases. The OG database is comparatively small, which may explain the observed lack of convergence, especially when compared to the larger COCO database.

5.3 Applications

To facilitate visual inspection of the previous results, the pre-trained model with the highest average precision (mAP) was selected for each algorithm and tested against the Oil and Gas (OG) database test data. Figure 7 illustrates the recognition performance of each algorithm model when applied to four different images from the test data.

Fig. 7: Visual object detection results from pre-trained YOLO V8, FASTER-RCNN and DETR on 4 test images from OG database (images sources @Google Earth)

Case A: This scenario showcases a view of three wells that are closely spaced. YOLO v8 is able to detect and correctly discriminate each well. In contrast, DETR detects the presence of wells but combines the first two into a single object. FASTER-RCNN, which has significantly low average precision (AP) for the well class (as shown in Table 1), is not able to detect a well in this particular scenario.

Case B: This case represents an unusual circumstance where the appearance of compressors is underrepresented in the Permian Basin and consequently in the learning database OG (the most common representation is shown in Case B). YOLO v8 can only detect one of the five compressors and one of the three tank units. DETR shows a slight improvement and detects three of the five compressors and all three tank units. Interestingly, FASTER-RCNN shows superior performance accurately recognizing all infrastructures without error.

Case C: This scenario presents a view of a typical compressor type found in the Permian Basin. In this specific instance, all algorithms correctly identify the six compressors and a single tank unit.

Case D: This case showcases a view of two sites each with a well. Unlike Case A, the image resolution in this case is lower and the wells are more widely spaced. YOLO v8 and DETR successfully recognize the two wells, while FASTER-RCNN fails to recognize either.

As highlighted in Case B, the visual representation of compressors in the Permian Basin is variable. The OG database contains a few cases where compressors are protected by a roof (as shown in the Case B images). To evaluate the detection capabilities of the algorithms in these particular circumstances,

tests were extended to three additional images from the test database showing covered compressors (see Figure 8 results).

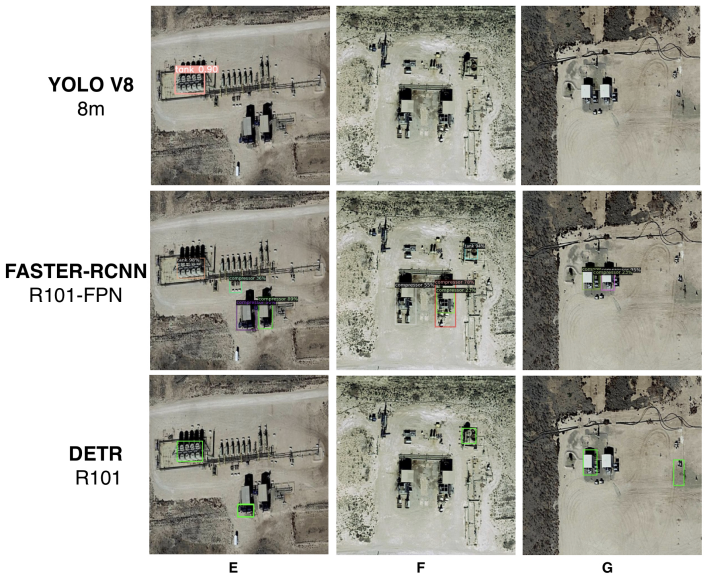

Fig. 8: Comparison of pre-trained YOLO v8, FASTER-RCNN and DETR on a special (less representative) compressor architecture in the Permian Basin @Google Earth

Case E: This scenario involves two unusual compressors along with a tank unit. FASTER-RCNN manages to identify all infrastructures, but it also mistakenly recognizes an additional compressor. DETR delivers an intriguing result by identifying a compressor through only a small segment protruding from the roof, as well as the tank unit. As for YOLO v8, it only manages to recognize the tank unit.

Case F: This scene provides a view of two unusual compressors and a tank unit. Both YOLO v8 and DETR fail to recognize the compressors, with DETR only acknowledging the tank unit. Yet again, FASTER-RCNN successfully identifies all the infrastructures as expected.

Case G: This scene presents a view of two unusual compressors exclusively. YOLO v8 is unable to detect either of them, while DETR correctly identifies one of the two compressors. It also detects an additional one, which does not correspond to a compressor but rather a small piece of infrastructure. Finally, FASTER-RCNN is also able to correctly recognize both compressors in this case.

6 Conclusion

To effectively mitigate methane emissions in the oil and gas (O&G) sector, a comprehensive emissions profile of each O&G infrastructure is essential which allows for an in-depth understanding of individual emission trends. This requires, as first step, an efficient solution to automatically detect and identify these infrastructures, a task that can be well addressed by detection algorithms. However, these algorithms come in myriad forms, each with a unique architecture and range of performance. While previous studies have compared specific algorithms for automatically detecting O&G infrastructures, these have typically focused on a single infrastructure type. In response to this limitation, this study presents a comparative analysis of three main supervised algorithms-YOLO v8, FASTER-RCNN, and DETR-for the simultaneous detection of tanks, wells and compressors. A unique database of various aerial snapshots of O&G infrastructure in the Permian Basin, USA, was used for the study. The tests performed showed that YOLO v8 outperformed FASTER-RCNN and DETR in terms of accuracy, although it was not pre-trained. Interestingly, FASTER-RCNN showed a superior ability to detect compressor styles that are significantly underrepresented in our database (and in the Permian Basin) on certain occasions.

References

1. IPCC. https://www.ipcc.ch/report/sixth-assessment-report-working-group-3.
2. Hao Sheng, Jeremy Irvin, Sasankh Munukutla, Shawn Zhang, Christopher Cross, Kyle Story, Rose Rustowicz, Cooper Elsworth, Zutao Yang, Mark Omara, et al. Ognet: Towards a global oil and gas infrastructure database using deep learning on remotely sensed imagery. *arXiv preprint arXiv:2011.07227*, 2020.
3. Bryan Zhu, Nicholas Lui, Jeremy Irvin, Jimmy Le, Sahil Tadwalkar, Chenghao Wang, Zutao Ouyang, Frankie Y Liu, Andrew Y Ng, and Robert B Jackson. Meterml: A multi-sensor earth observation benchmark for automated methane source mapping. *arXiv preprint arXiv:2207.11166*, 2022.
4. Tony Lindeberg. *Scale Invariant Feature Transform*, volume 7. 05 2012.
5. Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 1:886–893 vol. 1, 2005.
6. Pin Wang, En Fan, and Peng Wang. Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, 141:61–67, 2021.
7. Gabriel Huang, Issam Laradji, David Vazquez, Simon Lacoste-Julien, and Pau Rodriguez. A survey of self-supervised and few-shot object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
8. Xiao Youzi, Zhiqiang Tian, Jiachen Yu, Yinshu Zhang, Shuai Liu, Shaoyi Du, and Xuguang Lan. A review of object detection based on deep learning. *Multimedia Tools and Applications*, 79, 09 2020.
9. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

10. Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
11. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
12. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. pages 779–788, 2016.
13. Juan Terven and Diana-Margarita Cordova-Esparza. A comprehensive review of yolo: From yolov1 to yolov8 and beyond. 04 2023.
14. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.
15. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
16. Lu Tan, Tianran Huangfu, Liyao Wu, and Wenying Chen. Comparison of retinanet, ssd, and yolo v3 for real-time pill identification. *BMC Medical Informatics and Decision Making*, 21, 11 2021.
17. Débora F. Dos Santos, André O. Françani, Marcos R. O. A. Maximo, and Arthur S. C. Ferreira. Performance comparison of convolutional neural network models for object detection in tethered balloon imagery. In *2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)*, pages 246–251, 2021.
18. Maroš Jakubec, Eva Lieskovská, Boris Bučko, and Katarína Zábovská. Comparison of cnn-based models for pothole detection in real-world adverse conditions: Overview and evaluation. *Applied Sciences*, 13(9), 2023.
19. Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. pages 213–229, 2020.
20. Kanchan Bhil, Rithvik Shindihatti, Shifa Mirza, Siddhi Latkar, Y. S. Ingle, N. F. Shaikh, I. Prabu, and Satish N. Pardeshi. Recent progress in object detection in satellite imagery: A review. In Sagaya Aurelia, Somashekhar S. Hiremath, Karthikeyan Subramanian, and Saroj Kr. Biswas, editors, *Sustainable Advanced Computing*, pages 209–218, Singapore, 2022. Springer Singapore.
21. Yi Wang, Syed Muhammad Arsalan Bashir, Mahrukh Khan, Qudrat Ullah, Rui Wang, Yilin Song, Zhe Guo, and Yilong Niu. Remote sensing image super-resolution and object detection: Benchmark and state of the art. *Expert Systems with Applications*, 197:116793, 2022.
22. Zheng Li, Yongcheng Wang, Ning Zhang, Yuxi Zhang, Zhikang Zhao, Dongdong Xu, Guangli Ben, and Yunxiao Gao. Deep learning-based object detection techniques for remote sensing images: A survey. *Remote Sensing*, 14(10), 2022.
23. Junhyung Kang, Shahroz Tariq, Han Oh, and Simon S. Woo. A survey of deep learning-based object detection methods and datasets for overhead imagery. *IEEE Access*, 10:20118–20134, 2022.
24. Min Li, Zhijie Zhang, Liping Lei, Xiaofan Wang, and Xudong Guo. Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster r-cnn, yolo v3 and ssd. *Sensors*, 20(17), 2020.

25. Kemal Hacıefendioğlu, Hasan Basri Başağa, and Gökhan Demir. Automatic detection of earthquake-induced ground failure effects through faster r-cnn deep learning-based object detection using satellite images. *Natural Hazards*, 105:383–403, 2021.
26. Dmitry Demidov, Rushali Grandhe, and Salem Almarri. Object detection in aerial imagery. 11 2022.
27. Qingyun Li, Yushi Chen, and Ying Zeng. Transformer with transfer cnn for remote-sensing-image object detection. *Remote Sensing*, 14(4), 2022.
28. Arsalan Tahir, Hafiz Suliman Munawar, Junaid Akram, Muhammad Adil, Shehryar Ali, Abbas Z. Kouzani, and M. A. Parvez Mahmud. Automatic target detection from satellite imagery using machine learning. *Sensors*, 22(3), 2022.
29. Jiahe Zhu, Xu Chen, Huan Zhang, Zelong Tan, Shengjin Wang, and Hongbing Ma. Transformer based remote sensing object detection with enhanced multispectral feature extraction. *IEEE Geoscience and Remote Sensing Letters*, pages 1–1, 2023.
30. Yi-Jie Yang, Suman Singha, and Roberto Mayerle. A deep learning based oil spill detector using sentinel-1 sar imagery. *International Journal of Remote Sensing*, 43(11):4287–4314, 2022.
31. Yi-Jie Yang, Suman Singha, and Ron Goldman. An automatic oil spill detection and early warning system in the Southeastern Mediterranean Sea. In *EGU General Assembly Conference Abstracts*, EGU General Assembly Conference Abstracts, pages EGU22–8408, May 2022.
32. Lu Zhang, Zhenwei Shi, and Jun Wu. A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(10):4895–4909, 2015.
33. Pengfei Shi, Qigang Jiang, Chao Shi, Jing Xi, Guofang Tao, Sen Zhang, Zhenchao Zhang, Bin Liu, Xin Gao, and Qian Wu. Oil well detection via large-scale and high-resolution remote sensing images based on improved yolo v4. *Remote Sensing*, 13(16), 2021.
34. Guanfu Song, Zhibao Wang, Lu Bai, Jie Zhang, and Liangfu Chen. Detection of oil wells based on faster r-cnn in optical satellite remote sensing images. In *Image and Signal Processing for Remote Sensing XXVI*, volume 11533, pages 114–121. SPIE, 2020.
35. Zhibao Wang, Lu Bai, Guangfu Song, Jie Zhang, Jinhua Tao, Maurice D. Mulvenna, Raymond R. Bond, and Liangfu Chen. An oil well dataset derived from satellite-based remote sensing. *Remote Sensing*, 13(6), 2021.
36. Boris Gašparović, Jonatan Lerga, Goran Mauša, and Marina Ivašić-Kos. Deep learning approach for objects detection in underwater pipeline images. *Applied Artificial Intelligence*, 36(1):2146853, 2022.
37. Nannan Zhang, Yang Liu, Liqun Zou, Hang Zhao, Wentong Dong, Hongying Zhou, Hongyan Zhou, and Miaofen Huang. Automatic recognition of oil industry facilities based on deep learning. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 2519–2522, 2018.
38. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. pages 248–255. Ieee, 2009.
39. Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. pages 740–755, 2014.
40. Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, jun 2010.