EC[X]

Scientific Contribution on Smart Data Management in an Iterative Context

Document reference number for ANR

contact@confiance-ai.fr | www.confiance.ai

Document reference: XXX

Contributors

	Name	Organisation	Role
Responsible for the deliverable	Nicolas Granger	CEA	Data Scientist
Scientific responsible	Nicolas Granger	CEA	Data Scientist
Co-authors	Anca Molnos	CEA	Data Scientist
	Fritz Poka Toukam	CEA	Data scientist
	Louis Lerbourg	CEA	Data scientist
	Oriane Simeoni	Valeo	Data Scientist

Document Control

Revision	Date	Commentary	Author
v1.0			

Contents

A. Introduction and abstract

A.1. General Introduction

Recent advances in Machine Learning for Computer Vision has paved the way for real-world automation of perception tasks on images such as image classification, object detection or semantic segmentation. The last decade rapid improvements were enabled by the progress of hardware, more capable deep models and large datasets to fit their parameters. Although Deep Neural Networks are now understood and implemented well enough to consider their use in practical applications, they are usually trained in a fully supervised fashion and therefore require large datasets to be manually annotated specifically for the downstream task considered. This remains a limitation in industrial setups, where the deployment of a new model requires adaptation to a specific domain and task. These models are also usually trained to identify a restricted amount of classes, but they lack the ability to be easily trained with new data, to identify new classes, and in the same time not forget the old ones (no catastrophic forgetting).
This report present experimental studies to tackle the above listed challenges, extending what we already did in the previous batches of Confiance.ai program. On the one hand, we continue to explore active learning that aims at reducing annotation costs by selecting only a limited number samples to annotate. We present different strategies to select these relevant samples with new designed acquisition functions in chapter B. On the other hand, we continue our experimental studies on incremental learning, which is a paradigm where a machine learning agent evolves continuously, learns new tasks and accumulates the knowledge from previous tasks. In chapter C, we focus on incremental learning with a memory buffer in the context of object detection and we investigate the application of pseudo-labeling in addition to the memory buffer.

A.2. How to use this document

This document details experiments and studies on active learning and incremental learning for 2D object detection. The observations and insights gained throughout this work are compiled in a methodological guidelines report which provides a good entry point toward implementing the proposed methods, with practical and operational observations. But to gain better insights on the covered subjects, the present document will provide additional details about the scientific methodology, the experiments and the performance metrics.
Finally, this scientific study will be useful for future work that might target a different scope or to address new use-cases.

B. Active Learning for 2D Detection

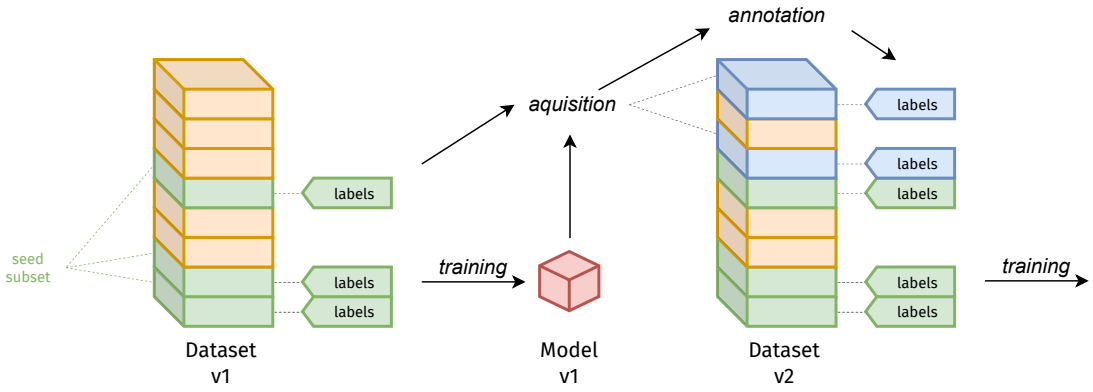


Figure B.1: Overview of the Active Learning process. At its core, the acquisition function selects samples to annotate for the next revision of the dataset.

B.1. Introduction

Deep neural networks for perception need to be trained on huge annotated databases. This appears to be a limit for many real-world problems as well as it is expensive. Indeed, the annotation of industrial datasets require the definition of strict annotations rules, the training of annotators and the long annotation process itself. Annotation for the object detection task is particularly expensive, indeed over 30s per box in an image were reported at the creation of the COCO dataset Lin et al. (2014). In order to mitigate such costs, several solution exist: semi-supervised learning Liu et al. (2021a); Chen et al. (2021), weakly-supervised learning Wu et al. (2021); Chen et al. (2021) or even unsupervised learning Siméoni et al. (2021); Wang et al. (2022). Another very relevant solution in an industrial setup is the *active learning* framework Vo et al. (2022); Brust et al. (2019); Choi et al. (2021) which consists in selecting *most important data* to be annotated therefore avoiding the annotation of redundant and not interesting data. We have proposed a first study in batch one (the report can be found here) of the interest of such *active learning* methods and shown that it was interesting particularly for little represented classes. We have also studied in batch two the application of the active learning framework for some of confiance use-cases (report here).

We summarize in Figure B.1, the active learning process which consists in iteratively selecting unlabeled data to be manually annotated using an *acquisition function*. Given the newly annotated data, a new version of the task model is trained, which is then used again for the acquisition. The process is performed in different *cycles*, each increasing the dataset with newly labelled data. We present the active learning framework in more details in subsection B.2.1.

Throughout this chapter, we investigate how to improve active learning results for our considered use cases by developing new acquisition functions. We present our new approaches adapted to the object detection task in section B.4. Finally, we tackle in section B.5 the question of the initial labeled set selection which is crucial to bootstrap the model for the active learning iterative process.

B.2. Background and experimental setup

B.2.1 Related work

We develop in this section a brief related work allowing to situate this work.

B.2.1.0.1 Learning with little supervision. As discussed before, limiting costs of annotation is an important problematic that has been widely studied by academic research. In particular, semi-supervised learning mixes annotated data with unlabeled data Liu et al. (2021a); Chen et al. (2021), weakly-supervised learning Wu et al. (2021); Chen et al. (2021) uses 'weaker' type of annotation e.g. only class-level information, few-shot learning Kang and Cho (2022); Liang et al. (2022) goal is to learn from very few samples and unsupervised learning aims at exploiting only information from unlabeled data Shin et al. (2022); Siméoni et al. (2021).

B.2.1.0.2 Self-supervised learning. In self-supervised learning, the idea is to pre-train a model to produce a good image representation by solving a pretext task (e.g., jigsaw solving, colorization, or rotation prediction) on unlabeled data Noroozi and Favaro (2016); Larsson et al. (2016); Gidaris et al. (2018); Caron et al. (2021); Gidaris et al. (2021); Chen et al. (2020a); Tian et al. (2020); He et al. (2020); Chen et al. (2020b). Such method have been adapted and shown great potential Caron et al. (2021); He et al. (2022); Zhou et al. (2022) for vision transformers Dosovitskiy et al. (2021). For instance, DINO Caron et al. (2021) uses a teacher-student framework in which both networks are provided a different and randomly transformed input parts; the student network learns to predict the output of the teacher network. Recent MAE He et al. (2022) proposes to mask an input image and the pretext task aims at learning to reconstruct the missing pixels by auto-encoding.

B.2.1.0.3 Semi-supervised object detection improve fully-supervised learning on little fully-annotated data by integrated information extracted from unlabelled data. Notably there are two different type of methods; those based on consistency Jeong et al. (2019); Tang et al. (2021) which integrate losses that verify that several versions of an image have the same representation, and those which integrate pseudo-labels Li et al. (2020); Radosavovic et al. (2018); Sohn et al. (2020); Wang et al. (2018); Xu et al. (2021); Zoph et al. (2020).

B.2.1.0.4 Active learning for object detection is another solution to limit annotation costs. It consists in carefully *selecting* the data to be fully annotated given a certain *criteria*, such as to minimize human annotation efforts. Such criteria is defined for a particular model, i.e. the task model. Typically the selection is performed using *data diversity* Geifman and El-Yaniv (2017); Sener and Savarese (2018) or *model uncertainty* Brust et al. (2019); Choi et al. (2021). These strategies have first been designed for the simple classification tasks Settles (2009) and recently adapted for the more complicated object detection Choi et al. (2021); Yuan et al. (2021). The data diversity ensures that images selected depict various scenes/objects and are not redundant; to select images it is possible to use k-means Zhdanov (2019), the k-means++ initialization Haussmann et al. (2020) or the widely used core-set – a *representative* subset of a dataset Agarwal et al. (2020); Geifman and El-Yaniv (2017); Sener and Savarese (2018). On the other hand, other strategies try and find images which confuse the model the most. In particular, model uncertainty at the level of an image can be computed by aggregating confusion scores computed over the prediction score per box Brust et al. (2019); Haussmann et al. (2020); Pardo et al. (2021), by comparing predictions from different steps of model training Huang et al.

(2021); Roy et al. (2018), using ensemble of networks Beluch et al. (2018); Chitta et al. (2019); Haussmann et al. (2020), Bayesian Neural Networks Gal et al. (2017); Haussmann et al. (2020) or single forward networks mimicking an ensemble Choi et al. (2021); Yuan et al. (2021). It is also possible to assess the difficulty of a sample by using the gradients Ash et al. (2020), the influence of data on gradient Liu et al. (2021b) or to learn to predict the detection loss Yoo and Kweon (2019).

B.2.1.0.5 Combining both Related to us, works have investigated methods that integrate consistency losses which force a same representation between different transformed views of a same image Elezi et al. (2021); Kao et al. (2018); Gao et al. (2020).

B.2.2 Active Learning definition

In this section, we define the Active Learning (AL) paradigm. Given a new set of images with no annotation noted $\mathscr{U}$, the objective of an AL algorithm is select the most *useful* data to be annotated within $\mathscr{U}$ in order to maximize the performances of a given model whilst keeping annotation costs to a minimum.
The algorithm proceeds cyclically, taking the following actions:

1. Select a fraction of the samples to annotate–to be noted that for the first cycle, this selection is generally random for lack of a more informed criterion.
2. Annotate the selected batch of samples.
3. Train the model using all annotated samples.
4. Apply an *acquisition function*, which selects the next batch of samples to annotate.

The acquisition function is usually based on features extracted by the model and aims to find the most *useful* set of samples that can improve the training of the model. The definition of usefulness is not straight-forward, it has mainly been defined as samples which are hard for- or unseen by the model. In particular, experience shows that informative samples are often harder samples that lie close to the decision boundaries for classification, or samples that exhibit inconsistent or unconfident predictions with stochastic models. However, focusing on specific types of samples can introduce bias in the annotated dataset, thus the acquisition must satisfy dataset distribution properties such as diversity, representativeness, completeness and coverage.
For the classification task, well established methods are used as baselines in scientific publications:

random: Random selection simply selects samples randomly within the dataset. This is usually used as a lower bound for comparison. It preserves the distribution of the source dataset which can be advantageous for already well-balanced datasets. However, we argue that most academic datasets are artificially balanced by construction, which is not realistic for real-world application.

entropy: Brust et al. (2019); Haussmann et al. (2020) This acquisition function ranks samples by the entropy of the predicted classification probabilities. High entropy (indecisive prediction) samples are selected first. In the case of a Neural Network model with a softmax classification layer, this also tends to select samples with features close to the decision boundaries.

core-set: Sener and Savarese (2018) Core-set encompasses a distribution-based approach to dataset construction. Using the model as an embedding function toward a metric space, the acquisition function ensures that selected samples cover all the regions occupied by the whole dataset with minimal redundancy.

At each cycle, the size of the selected batch of samples is also referred to as the **annotation**

budget, and directly translate to an equivalent annotation effort and cost. In works on images classification, and most on detection, the budget is expressed in number of images, and all images incur an equal cost, but it could make sense to distinguish easy and hard to annotate images for certain applications. Recent approaches on active learning for object detection such as Lyu et al. (2023) express the budget in number of boxes to annotate, which is more representative of the annotation effort.

B.2.3 Acquisition functions for the object detection task

One crucial aspect of the 2D Object Detection task is the presence of multiple objects inside the images. Global image representations typically encode general domain information (ex: lighting conditions for BDD100k) which is insufficient to characterize diversity, coverage or difficulty of the objects in a dataset. As a result, the selection of the most useful samples by an active learning method should typically revolve around an object-wise criterion. However, these criteria must be reduced into image-level scores in order to decide which images should be annotated. Indeed, we purposefully exclude partial annotation or semi-supervised methods from this study as it would greatly reduce its scope and applicability. For active learning methods that revolve around an object-level score, several reduction strategies exist to compute an aggregated image-wide score: maximum, sum, mean.
The aforementioned acquisition functions are adapted for the object detection task. Classical works use **random** and **core-set** acquisition functions with no modification.
The ***entropy*** acquisition function is adapted to take into account the box predictions. In fact, each box is predicted with classification probabilities that can be used to compute an entropy value. At the image level, the entropy score considered is a reduction of the box entropy scores : the ***max-uncertainty*** uses the maximum entropy score of all the boxes and the ***mean-uncertainty*** computes the image entropy as the average of entropy scores of all the boxes in the image. The most used is the *max-uncertainty*, which usually gives better results than the *mean-uncertainty*. Previous results in the active learning benchmark for object detection on real-world datasets for the bacth 1 of the Confiance.ai program confirm this (report here).
Acquisition functions based on the **consistency** Elezi et al. (2022); Kao et al. (2018), have also been designed to suit the object detection context. The proposed acquisition functions relies on exploiting the discrepancies between boxes predictions of transformed views of a same image. The consistency criterion in the acquisition function measures the "difficulty" and thus usefulness for images. The more there are disagreements over predictions between views of the same image, the more the image is useful for the model and the acquisition function will chose that image for annotation. In particular, Elezi et al. (2022) have proposed to use two transformed views of the same image. In practice, samples in a batch are duplicated and augmented by different transformations independently, leading to the two related *views* of the same image. The computation of the consistency score for an image follows these steps:

1. Generate two views (augmentations) of each image
2. Infer model detections for both views (including Non-Maximal-Suppression step).
3. Invert spatial augmentations (flipping, scaling, etc.) such that both sets of predictions share the same spatial coordinates.
4. Greedily match spatially close boxes in pairs, eliminating those with an IoU below 0.5 : *box-matching*.

Having matched detections by pairs, the consistency criterion combines the KL divergence on the classification predictions to the classic uncertainty criterion . In particular, the usual maximum entropy criterion is multiplied by the maximum KL divergence between box pairs.

$$S_{cons} = \underbrace{\max_{k} \left[H(\mathbf{y}_0^k) \right]}_{\text{max entropy}} \times \max_{i,j \in \mathscr{M}} \underbrace{\left[\frac{1}{2} \left(KL(\mathbf{y}_0^i, \mathbf{y}_1^j) + KL(\mathbf{y}_1^i, \mathbf{y}_0^j) \right) \right]}_{\text{KL divergence}} \tag{B.1}$$

where $\mathbf{y}_0^k$ $\mathbf{y}_1^k$ are predicted box confidences from each view, $\mathscr{M}$ contains the indices of matched boxes that are assumed to enclose the same object in both views. This expression favors images containing examples that are either hard to classify (and consequently close to the decision frontier of the classifier), or confusing for the model, resulting in inconsistent prediction. It should be noted that box regression consistency is not used in this expression.

Elezi et al. (2022) also leverages the large set of unlabeled samples that is available during the active learning session with an unsupervised loss and takes advantage in the same time of "easier" samples via *pseudo-labelling*. With the matched detections pairs over the two augmented views, they integrate a *consistency loss* computed by adding up the KL divergence on the classification predictions with the L2 loss between box regressions. This consistency loss is used in conjunction with the fully-supervised training loss. The unsupervised objective enforces consistency between augmented views of a given image.

We implemented this pipeline of Elezi et al. (2022) in the batch 2 of the Confiance.ai program, and our experiments showed using that unsupervised loss exploiting the large unlabelled set during active learning cycles was of high interest (report here). Ablation studies proved that the consistency loss had more impact on the performance than the selection that uses the consistency score itself. This unsupervised loss combined with the pseudo-labelling are the main ingredients of the good results observed by Elezi et al. (2022). The consistency-based acquisition function has a limited performance in the active learning results obtained. Moreover, using pseudo-labelling an unsupervised loss has a high computational cost and those two approaches can be developed in parallel to improve the overall results, independently to the acquisition function. This leads us to focus on improving the acquisition with a new implementation of the consistency score (see section B.4).

B.2.4 Benchmark protocol

To compare the performance of each method, we use the following combinations of experimental settings:

Dataset : Pascal VOC Everingham et al. (2010) is a standard dataset for Active Learning benchmarks which contains photographies from Internet, BDD100k Yu et al. (2020) contains road driving images taken from dashcam cameras, and Valeo Deep Perception contains road driving images from fish-eye cameras mounted around a vehicle.

Model : The FCOS model is used for its good compromise between speed and detection performance.

Seeds and annotation budgets : To train the first version of the detection model (Figure B.1) which is used by the acquisition model, a random set of images called seed is drawn and annotated. Then the acquisition/annotation/training cycle is run, each time spending a portion of the annotation budget. For our experiments, we consider the budget configuration $[700, 350, 350, 350, 350]$ meaning we begin with a random initial set of 700 images and use an annotation budget of 350 for each cycle after.

Acquisition function : Experiments are run on the methods presented in the previous section.

Evaluation Metric : We evaluate the performance of the model in term of Mean Average Precision or mAP. More specifically, we use the mAP at IoU $>$ 0.5 metric as implemented

for the COCO 2017 detection benchmark. The performance metrics is put in correspondence with the annotation budget which is measured in number of images or total number of annotated boxes, which will also be referred to as *image count* and *box count* in our figures.

B.3. Data Characterization for active learning

To select data samples that bears desired properties (ex: coverage, non-redundancy, utility, etc.), active learning selection methods rely on metrics that quantify diversity or coverage to perform sampling. For instance, the entropy computed on the class probabilities per predicted box serves as a proxy to measure the *difficulty* of a sample. Alternatively image features can be exploited to select *diverse* data. We provide more details in section B.4.
However, performing a good selection assumes features and predictions of good enough quality, which is not necessary the case, in particular in the first cycles of the active learning process. Moreover, the notion of sample utility is hard to define and more generally to understand for Deep Neural Network.
This section aims to clarify and analyze experimentally some of the hypotheses made by active learning methods in order to select data samples with specific properties.

B.3.1 Object distribution in the embedding space

As discussed before, image features can be used to select *non-redundant* samples that maximally cover the embedding space occupied by the dataset, e.g. following coreset Sener and Savarese (2018). However, with the aim to select images which contains *diverse objects*, we explore in this section the embedding space at the *box-level* using individual object representations.
Moreover, the last few years has seen the emergence of good quality self-supervised features Caron et al. (2021); Chen et al. (2020b) which are learnt without any annotation, on unlabelled data. Such self-supervised models are often used to initialize model backbones, but have also shown good image representation properties, achieving high k-nn classification scores. Indeed, the representations produced by these models have useful properties such as proximity between objects of similar nature and similar visual appearance—such properties have for instance been exploited for the task of unsupervised object localization Siméoni et al. (2021); Siméoni et al. (2023); Wang et al. (2022, 2023).
In our work, we propose to utilize these self-supervised models to produce representations for each object detected by the model obtained at the last iteration of our active learning cycle. Firs, this can be more practical than trying to extract internal representations from the detector itself. Indeed, some detectors do not have a clearly identified embedding layer that is related to the final predictions. For example, SSD **?** has independent prediction heads for different object size levels that work on separate input spaces. Moreover, for models that do have such an embedding layer such as YOLO v5 or FCOS, we observed that the embeddings are strongly correlated with the output class and size, but information about appearance seems absent upon inspection of clustering results as shown in Figure B.2 This can be explained by the nature of supervised training itself, which does not enforce the model to retain such information.
Based on these observations and early experimental results, we speculate that a third party self-supervised model can produce better object-level representations for an active learning algorithm. Indeed, self-supervised feature extractors are trained on millions of object-centric images, several orders of magnitude more than typical datasets where active learning is employed. These models are susceptible to produce more robust and sensible representations, especially for rare object classes and outliers. This intuition is supported by visual inspection as shown in Figure B.3.
Despite the aforementioned practical advantages, a few important properties must be accounted for in order to apply them in a use-case such as object detection:

- The representation models are not trained to encode contextual information, whereas a

Figure B.2: UMAP Sainburg et al. (2021) projection of FCOS model Tian et al. (2022) embeddings for predicted objects on VOC dataset. The embeddings seem to only weakly encode notions of object class, as different classes appear inside a given cluster. Bounding box size visibly influences the clustering.

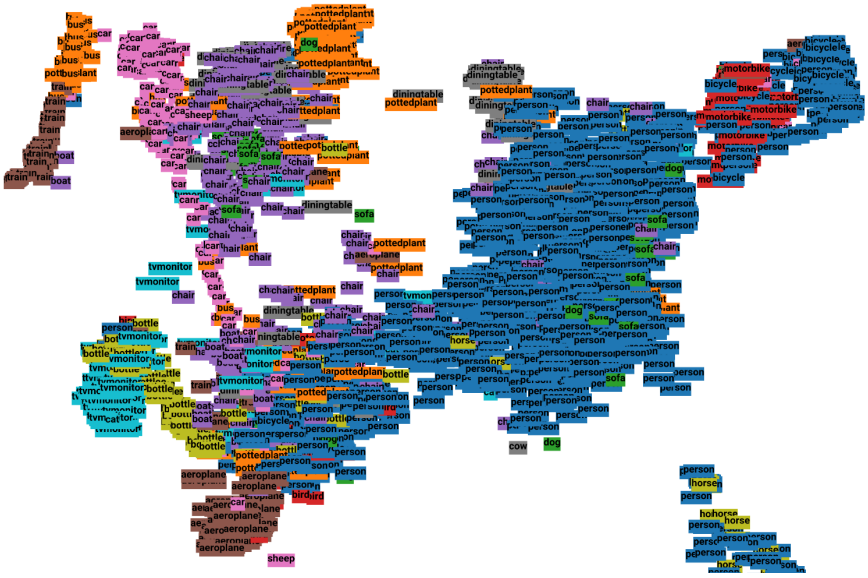


Figure B.3: UMAP projection of DINO Caron et al. (2021) (Deit small, imagenet unsupervised pretraining) embeddings for objects in VOC dataset. Visual inspection reveals an interpretable cloud structure with visual continuity and consistency within clusters.

detection model could be influenced by the surroundings of an object.
- Representation models are biased toward certain classes: representations vector of common classes (ex: people, cars, dogs and cat) are more varied, encode finer variations.
- It results from the above that the density of points in the embedding space is not constant between different classes: some occupy a greater volume of the embedding spaces whereas rarer classes collapse altogether into smaller regions.

To visualize how embedding models project different objects classes differently, we propose to evaluate the distance between box representations obtained after applying different type of augmentation (e.g. scaling, color jittering, ..). Per class results are shown in Figure B.4 and we observe that features distances vary depending on the class. In other words, for changes in appearance of identical type and strength (image augmentations), the embeddings vary more or less depending on the class because the model clusters certain classes more tightly than others.
Given previous observations, we propose a method to leverage such self-supervised features in the context of the active learning framework. Such features have been employed by recent active learning works Samet et al. (2023) with a focus on the image classification task. Here, we propose to leverage box-level features in a box-level sampling strategy order to perform efficient diverse object sampling (more details in subsection B.4.2). We also propose a new initial seed selection (see section B.5) which exploits such features and allows an interesting first selection without requiring any manual annotation.
Finally, we also study using the *consistency* between different views of an images in order to perform a good selection and propose this as an alternative to relying solely on a diversity criteria.
In particular, we discuss in subsection B.4.1 how the robustness to augmentation can be utilized to produce a difficulty metric.

B.3.2 Hard example mining and difficulty measurement

We further discuss in this section the concept of *hard example mining* which is at center of the active learning process.

B.3.2.1 Active learning hypotheses

Active learning strategies often aim at selecting 'hard' samples, hoping that they will be useful samples from which the model can learn something new. This objective results from the combination of several factors:
- In its typical setting, the active learning process starts with an initial model trained on a subset of annotated data (randomly selected). Usually the first model is capable of detecting 'easy' objects (eg. from very frequent classes, large objects, not occluded) which are well represented in the initial selection. In order to further improve the model capacity, annotation effort need to be invested on harder samples.
- Moreover, annotation for 'easy' examples can be produced without labels by using semi-supervised learning techniques, such as pseudo-labeling Lee and others (2013).
- Typical images of benchmarks considered here depicts multiple objects, so selecting images with hard examples will likely lead to the annotation of more easier instances as well. Indeed when an image is selected, all objects depicted are annotated (easy or hard).

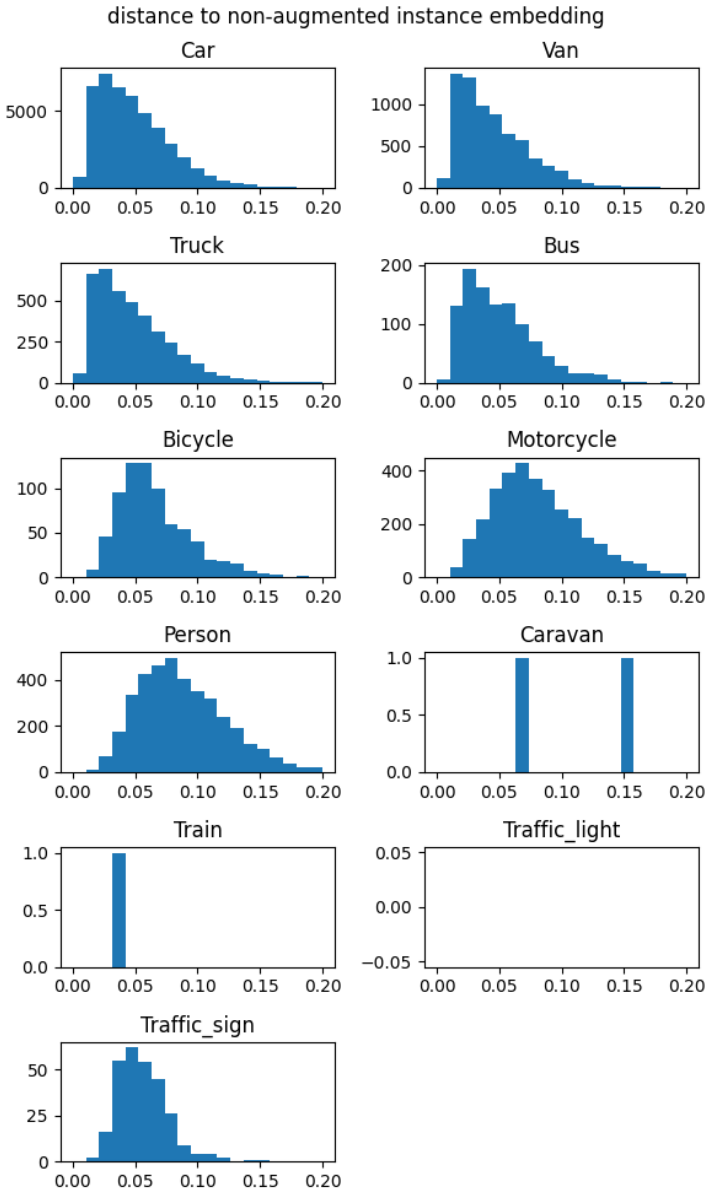


Figure B.4: Cosine distance distributions between embeddings of non-augmented and their augmented views. Object are extracted from BDD100k dataset and embeddings produced by a DEIT-S model pretrained with DINO method.

B.3.2.2 Hard example mining by scoring function

In order to find 'hard' and useful samples, three active learning strategies can typically be used (they are discussed in greater details in subsection B.2.1):

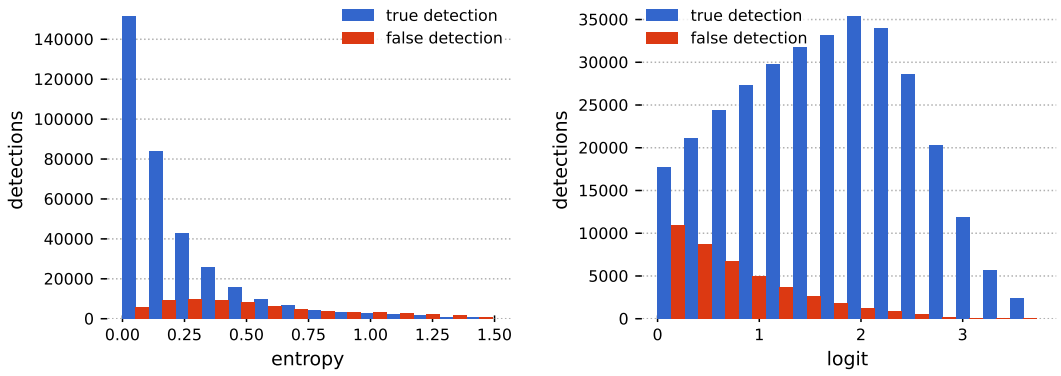
Entropy measures the confidence of the model based on the sharpness of the class probability prediction.

Logit uses the raw un-normalized logit from the predicted object class.

Consistency measures how stable the detector is across multiple augmented views of a given image. Refer to Equation B.2 for exact expression.

Model confidence can also be estimated by using calibration techniques, but we are not aware of application to active learning. Indeed, such method typically require a lot of annotated data, which is the opposite setup to the one considered here.

We now compare in Figure B.5 the active learning scores obtained with the three metrics versus 'true' and 'false' detections. Experiments are run with an FCOS model trained on 700 images of BDD100k dataset. Model detections are matched with ground-truths when the IoU > 0.5 following COCO evaluation protocol.



(a) Entropy

(b) Logit

(c) Consistency

Figure B.5: Active learning score distribution for true and false detections.

Neural Networks trained in a supervised fashion are known to be poorly calibrated, so it is expected to not obtain perfect separation between correct or not predictions. If true predic-

tions have good entropy/logit/consistency scores, we observe that a non-negligible proportion of false detections are mixed among the most confident ones for all methods. Interestingly, the consistency approach appears less discriminative than the entropy, even though it computes a stochastic estimation of the confidence. In spite of its lower ability at discriminating errors, we suspect that the consistency method may be able to select interesting hard examples that happen to be correctly detected nonetheless.

B.3.2.3 Hard example mining in the embedding space

Alternatively, an Active Learning sampling method can focus on selecting diverse data in the dataset. The objective of the core-set approach is to exactly maximize diversity. We focus here on object-level predictions and in particular compute an embedding *per detected boxes*.
We aim here at studying the quality of the self-supervised embedding space, and in particular if objects that are difficult to detect due to their appearance might be clustered together. More specifically, we wonder if it is possible to find unlabeled images with hard examples by sampling near known false detection.
To verify this hypothesis, we project run the following algorithm, which could also work as an active learning selection method:

- Train an FCOS detector on 700 annotated images randomly on the dataset.
- Infer detections on all images (annotated or not).
- Compute an embedding vector for each detection using a DINO pre-trained ViT model.
- For annotated images, identify true detections and false detections by matching predictions to the ground-truth.
- For each of the above detections, find the detection with the nearest embedding (cosine distance) in the unlabeled set.
- Verify whether the nearest neighbour is a true detection (this would require annotation work in an actual active learning pipeline, but in our case our study dataset in fully annotated already).

Using the above method, we find, on the BDD100k dataset that the nearest neighbor of a *false detection* has a probability of 26% of being a *false detection* as well. For a true detection, that probability drops to 15%, so this sampling strategy might be effective at selecting hard examples that cause false detections. However, this strategy is ineffective for selecting other type of useful samples: hard examples with correct prediction and missed detections. As shown in Figure B.6, the false detections found by this method are mainly low confidence ones, therefore missing more important high confidence false detections.

B.3.2.4 Observations

From these two experiment, we devise the following categories of samples to account for in the design of an object-level active learning selection strategy:

- Correct and confident detections with appropriately low selection metrics.
- Ambiguous / hard examples correctly identified by low confidence and high selection metric.
- Missed detections.
- Incorrect class of box where the model is incorrectly confident and/or selection score low.
- Correct detections where the model is not confident and/or selection metric high.

We explored several clustering techniques and metrics to organize detections and try to identify more challenging objects. Incorrect *detections* are not easily characterized through a represen-

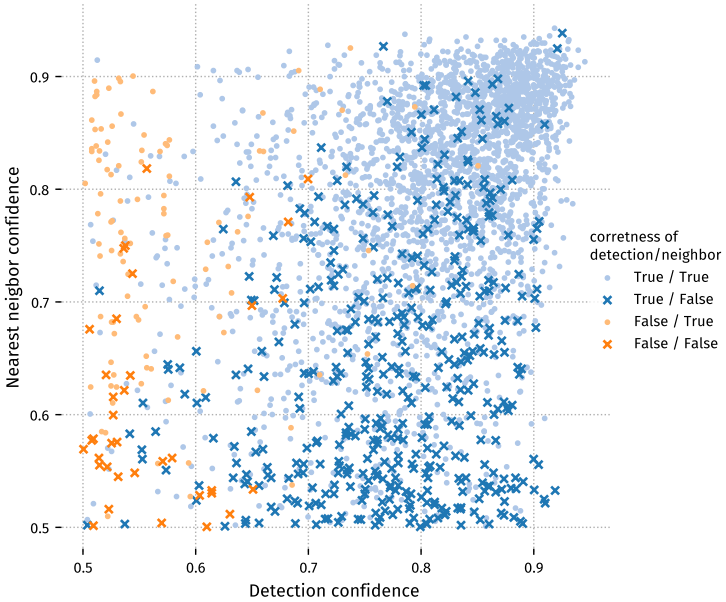


Figure B.6: Confidence predictions of detections and their nearest neighbor in the embedding space. Prioritizing selection nearby false detection helps find more false detections but introduces a bias toward low confidence detections.

tation from the detection model or an external model. It might indicate that such examples are in fact outliers for which the detectors and embedding models fail to produce meaningful representations. We also confirm this impression through visual inspection of the objects, which are indeed visual outliers.
Since all of the studied methods revolve around detection-wise scores, it results that none of them can prioritize the selection of a *missed detection*. It should be noted however that experimenting with a lower detection threshold to recover more missed detection proved unfruitful in our preliminary experiments. With the FCOS detector used for our experiments, the configuration of hyper-parameters that performs best overall leads to a detector that produces few false detection in the background (object detected where no object actually exists). Instead, errors usually originate from an actual object with erroneous classification or an incorrect bounding box regression.

B.3.3 Conclusion

In order to obtain good active learning results, selection strategies must seek to select samples that present a combination of properties. Some methods such as maximum entropy or consistency focus on the utility of samples. Others such as coreset focus on coverage and non-redundancy. For a particular dataset, identifying the priority between these properties is a crucial step to determine the most adequate selection method.
The way active learning methods achieve the selection of samples with desired properties relies on assumptions. These must be taken into account and monitored closely. For instance, maximum entropy assumes the entropy provides an estimate of model confidence, but this estimate is only reliable if over-fitting and therefore overconfidence is prevented. The coreset method assumes the detector will train well on diverse data, but some datasets contain outliers on which

annotation effort is wasted. To implement active learning, one must identify the underlying assumptions associated to each method, and monitor the dataset, data distribution, model, etc. to ensure their requirements are met.
In the next section, we propose contributions on active learning methods which take these properties and requirements into account. Our objective is to propose methods that perform well on a wide range of applications, with minimal assumptions that are met by most 2D detection scenarios.

B.4. Improving AL strategies

We discuss in this section different possible strategies to improve the quality of the active learning process. In particular, we discuss an improved consistency strategy in subsection B.4.1 when in subsection B.4.2 we investigate the benefit to perform a diversity sampling at the level of the boxes. Finally, we compare all active learning strategies in subsection B.4.3.

B.4.1 Leveraging consistency

We have presented in subsection B.2.3 the work of Elezi et al. (2022) that leverages unlabelled data during active learning and proposes an acquisition function based on a consistency score. We also have recalled the outcomes of our implementation of this work, which showed a limited efficiency of their consistency acquisition function, showing that their success was relied on the unsupervised loss and pseudo-labelling. Therefore, we aim here at improving that consistency score in order to have a better selection of images to annotate (improving the acquisition function).
Also, ensemble-based active learning methods Roy et al. (2018); Beluch et al. (2018) have proven to be effective to select images to annotate by measuring the disagreement over inferences on multiple models, but have a high computational cost. Lyu et al. (2023) explores measuring disagreement over multiple stochastic views of the inputs that they consider as their committee members.
Inspired by such works, we propose here to use more views in the consistency computation than what proposed Elezi et al. Elezi et al. (2022) (only 2 views) and to add more augmentations (not only a *horizontal flip*). We show that such adjustments allows us to improve the characterization of the model consistency for a given sample.

B.4.1.1 Our consistency-based acquisition strategy

In this new design, for each image, we use one original view and *num_views* augmented views. The multiple augmented views are processed by the model and each produced set of predictions are compared to those of the original view with our box-matching scheme. Similarly to Elezi et al. (2022), we consider the following steps to compute the new *consistency score* on unlabelled images:

1. Generate multiple views (augmentations) of each image
2. Infer model detections for all the views and the original image (including the Non-Maximal-Suppression step).
3. For all augmented views, invert the spatial augmentations (flipping, scaling, etc.) such that the sets of predictions share the same spatial coordinates with the original view.
4. Greedily spatially match the boxes in pairs between those of each view and those of the original view. We eliminate overlapping boxes with an IoU below 0.5.
5. Produce a single consistency score for each view k. We compute for each paired boxes in $\mathscr{M}_k$, the *KL divergence* computed between the classification scores of the two boxes and the *IoU loss* (corresponding to $1 - IoU$). We sum those scores over all matched pairs to obtain a score per view. Such expression is defined in the right sum of the Equation B.2.
6. The consistency score for each unlabelled image is the sum of scores across the seven views scaled by a factor α.

We summarize in the following equation the computation of the consistency score for a single

image:

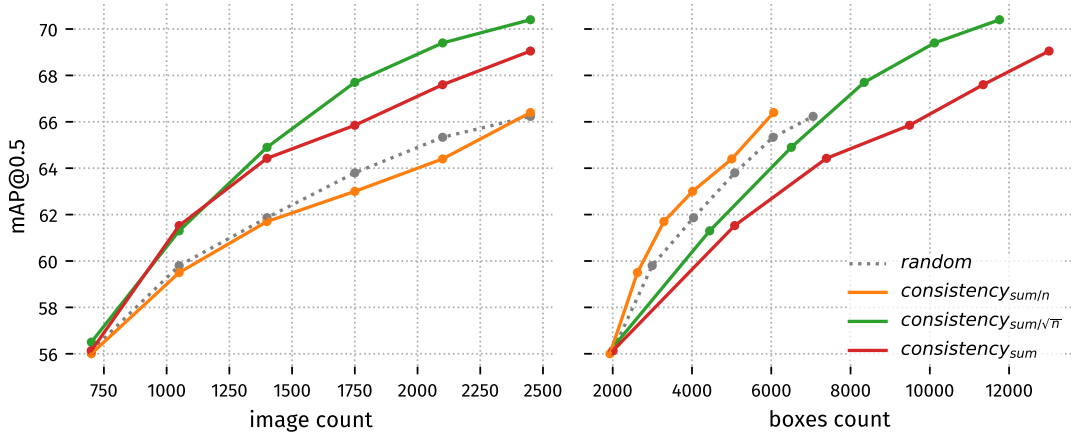
$$S^{\alpha}_{cons} = \alpha \times \sum_{k=1}^{num\_views} \sum_{i,j \in \mathscr{M}_k} \beta \times \underbrace{[1 - IoU(\mathbf{b}^i_0, \mathbf{b}^j_k)]}_{\text{IoU loss}} + \underbrace{\left[\frac{1}{2}\left(KL(\mathbf{y}^i_0, \mathbf{y}^j_k) + KL(\mathbf{y}^i_k, \mathbf{y}^j_0)\right)\right]}_{\text{KL divergence}} \quad \text{(B.2)}$$

with $k \in [1...num\_views]$ the index of the augmented view and $\mathscr{M}_k$ the set of indices of matched boxes that are assumed to enclose the same object in the original view and the augmented view k. We give the index 0 to the original view, and we note $\mathbf{y}^i_0$ and $\mathbf{y}^j_k$ the predicted box confidences, $\mathbf{b}^i_0$ and $\mathbf{b}^j_k$ the predicted box coordinates respectively from the original view and the augmented view k. The number of views *num_views* is set to $num\_views = 7$. We rescale the *IoU loss* with $\beta = 3$ in order to provide scores of similar amplitude to those of the KL loss.
We have observed that the factor α impacts results and investigate using an α which depends on the number of matched detections over all the augmented views. In particular, we experiment with $\alpha = 1$ (the score is then noted $consistency_{sum}$), $\alpha = 1/n$ (noted $consistency_{sum/n}$) and $\alpha = 1/\sqrt{n}$ ($consistency_{sum/sqrt(n)}$).

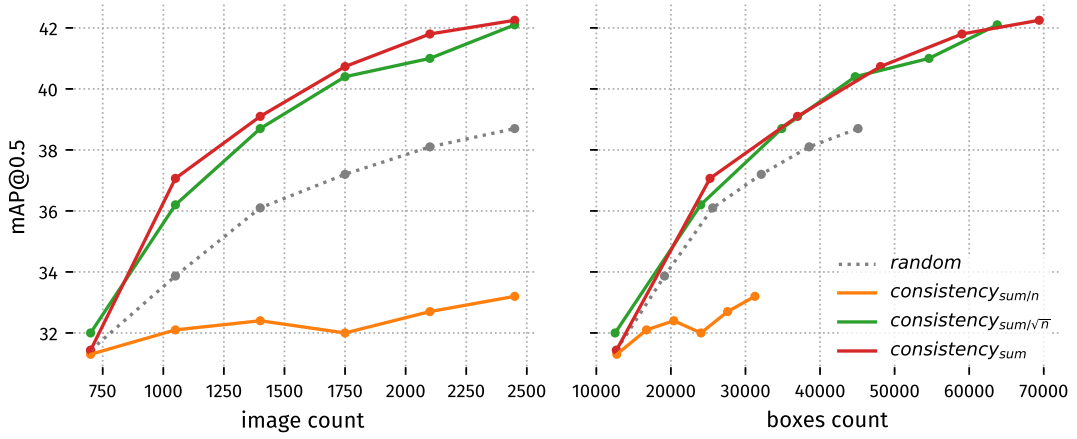
B.4.1.2 Evaluation

We evaluate the three variations of our consistency-based acquisition function on various datasets and report them in Figure B.7. We report scores obtained when considering the *image count* and the *boxes counts* (the metrics are detailed in subsection B.2.4). We observe that when considering one evaluation or the other, the order of the methods changes.

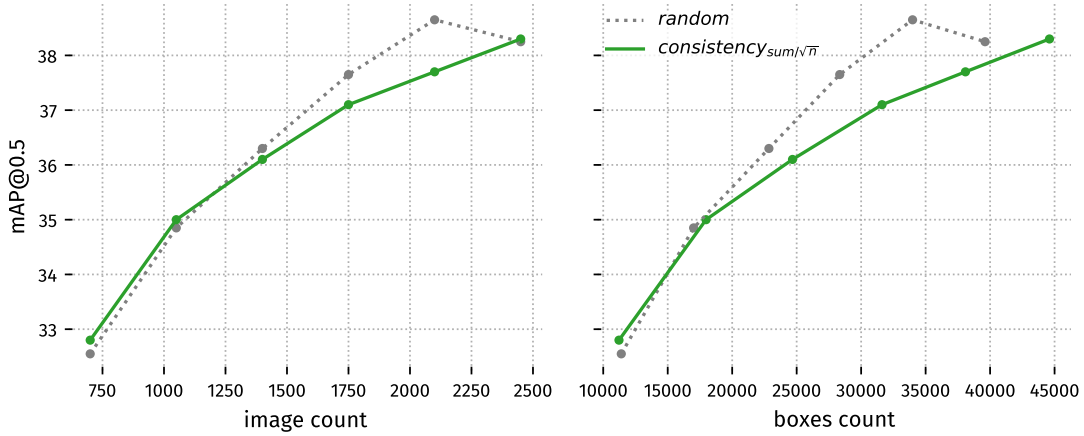
B.4.1.2.1 Impact of the aggregation factor α. We observe that the reduction from object-wise into image-wise scores, controlled by α, greatly affects the efficiency of the selection method, particularly in terms of map@0.5 v.s. the number of boxes. Indeed, we notice different behavior between the three variations on the different datasets. For instance, *sum* reduction improves mAP metric rapidly at the cost of using images with numerous objects.It strongly favors busy images as sufficiently many low scoring objects eventually add up to a large image score. One should note that this introduces bias in the selected images which may cause a lack of diversity in the selected images. To study this, we plot selected images from BDD100k with $consistency_{sum}$ and $consistency_{sum/n}$ in Figure B.8. We observe that the first (Figure B.8a) tends to select *busy* images which correspond to city centers and do not select much in other environments (suburban, night time, etc.). Conversely, the *mean* reduction do not have this bias for large number of objects but actually favors images that depicts hard examples. Again, this hurts the diversity on BDD100k dataset where this type of images corresponds to close to empty scenes with distant objects (Figure B.8b).
This discussion around the aggregation of object-wise scores to image-wise scores is also valid for the *entropy* acquisition function and has been studied in previous batches of this program, as recalled in subsection B.2.3.
We have briefly explored alternative reduction strategies but were not able to identify a robust and generalizable one. There is an inherent compromise for datasets that contain both useful and redundant objects in the same image. This compromise can only be effectively resolved with semi-supervised learning or smart annotation. We also observe that a dataset with fewer objects per image such as VOC (around 3 objects per image in average) is less sensitive to this issue. Consequently, we argue that active learning methods for 2D detection should benchmark their contributions on datasets with more objects per images, and measure performances with respect to annotated *boxes*.



(a) VOC



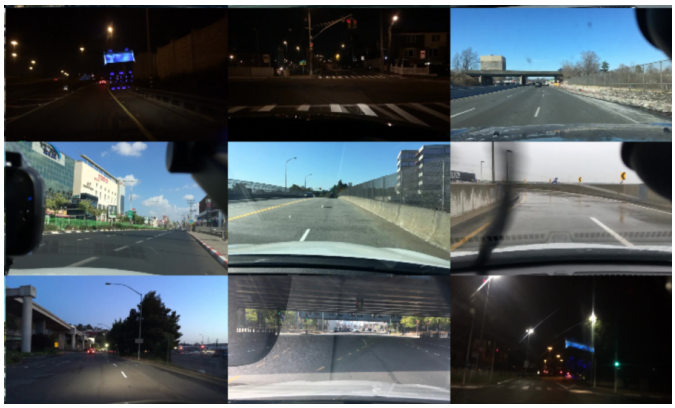
(b) BDD100k



(c) VDP

Figure B.7: Evaluation of the three variations of the consistency-based acquisition function on the three datasets VOC, BDD100k and VDP. The mAP@0.5 results are reported versus the number of annotated images at each AL cycles (left column), and versus the number of boxes annotated (right column) which also increase cycle after cycle.

(a) Sum reduction



(b) Mean reduction

Figure B.8: Example of images selected with consistency acquisition on BDD100k dataset. Mean reduction favors images with hard examples only, which translates to scenes with a few distant objects for datasets such as BDD100k.

B.4.1.2.2 Impact of the number of augmented views. The consistency method revolves around a stochastic estimation of model uncertainty. The empirical metric is estimated on a number of views which requires to infer detection multiple times per image. In order to find a compromise between performance and computation cost, we ablate the impact of the number of views in Figure B.9. We notice that using only 3 augmented views (plus the original image) provides unreliable results, whereas increasing from 7 to 11 augmentations does not provide additional benefit. This result is in line with Lyu et al. (2023) although the performance gaps we observed for our models and methods are narrower.

B.4.2 Box-level diversity

We have previously presented the interest to perform image-level diversity sampling in order to select images which depicts various object/scenes. We focus here on creating a diversity based method which aims at selecting images depicting *diverse objects* rather than diverse scenes. In particular, we consider the *core-set* strategy but this time at the box-level.
Our proposed box-level diversity selection process, illustrated in Figure B.10, follows the next steps:

- Run the last detection model trained on the unlabelled data to generate box predictions.

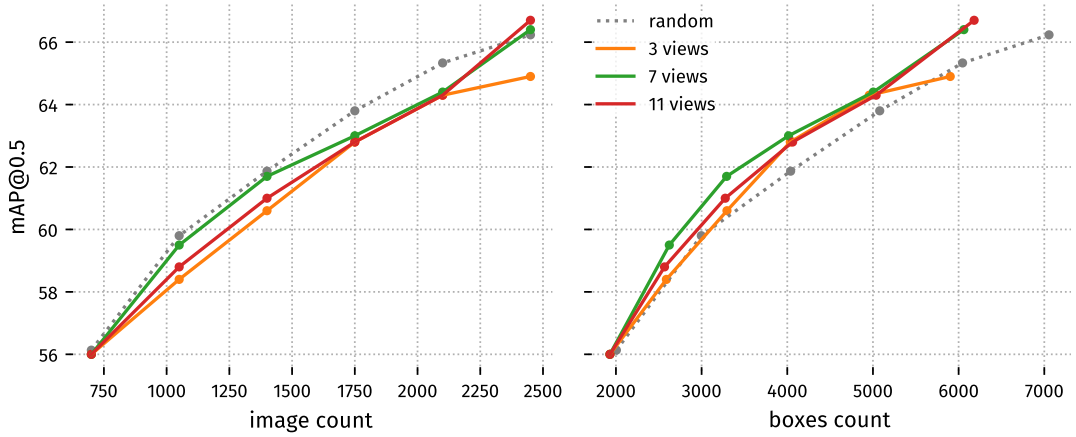


Figure B.9: Evaluation of our consistency-based acquisition function ($consistency_{sum/n}$) on VOC dataset when considering 3,7 and 11 views. The mAP@0.5 results are reported versus the number of annotated images at each AL cycles (left column), and versus the number of boxes annotated (right column) which also increase cycle after cycle.

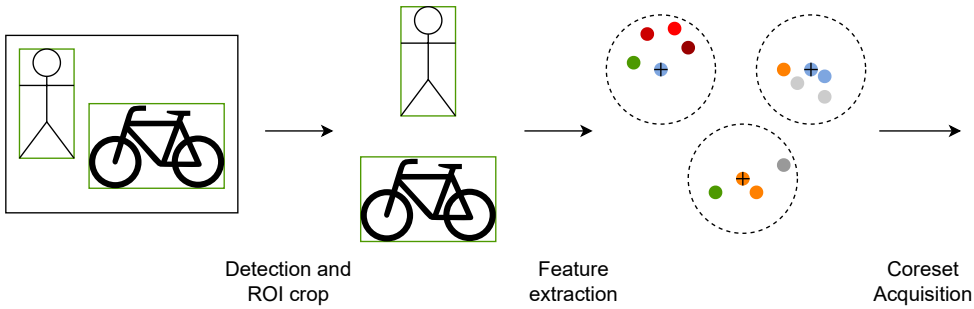


Figure B.10: Overview of our adaptation of the coreset acquisition method to a diverse selection at the box-level.

- Extract for each detected box an embedding vector, either using an intermediate representations produced by the detector, or via an external models that runs on cropped object views.
- Perform a diversity based selection of these embedding. Every iteration, an object (ie. a detection) is selected which causes all other detections from the same image to be selected as well. We test two different diversity-based selection strategies: Farthest Point Sampling (FPS) and KMeans (more details below).

B.4.2.1 Different diversity criterion

We consider here two different box-level diversity selection criterion which aim at ensuring the selection of data covering the dataset space.
We first adapt the classic core-set strategy Sener and Savarese (2018) which ensures the maximal distance between selected images. Indeed, the algorithm selects the samples that are the furthest away from already annotated objects and previously selected objects. Moreover, in order to avoid selecting outliers, we propose a filtering strategy which eliminates any data point further away to its centroid by more than the 60% percentile (we compute such centroid by using a k-means clustering algorithm with $k = 8 * nbclasses$). In practice, we observe that using this filtering is crucial to obtain good results.
We also consider an adaption of k-means clustering strategy, proposed by Kim et al. (2023) . K-means clustering is applied on all object embeddings, then the closest objects to each centroid is selected, which in turns select the corresponding images. This approach is less prone to selecting outliers since centroids are unlikely to be isolated, however it does not account for redundancy with already annotated data.

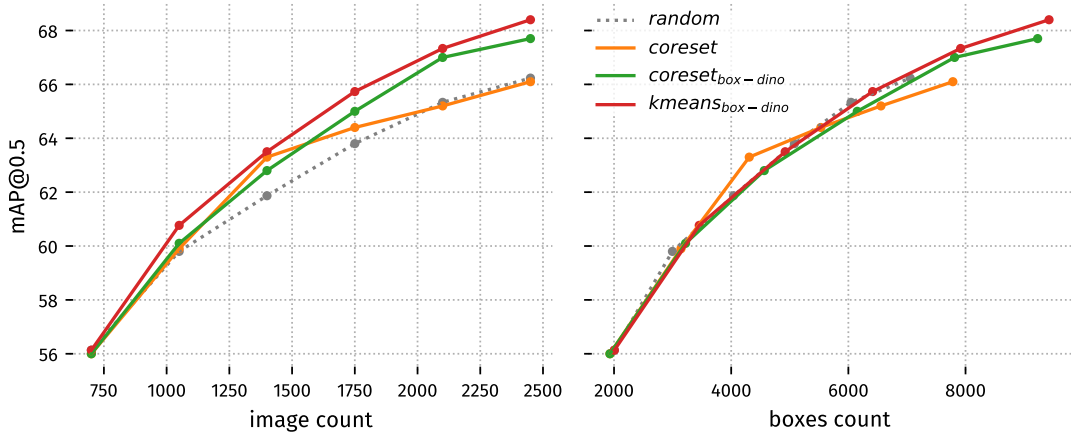
B.4.2.2 Different type of features

In order to extract box-level feature, we explore using the internal representations of the task-specific model; typically are used the intermediate features obtained before the regression and classification heads of the detector. However, we have observed in early experiments that such representations contain little information besides what is already explicitly encoded in the final prediction of class and size. In particular, appearance information seems absent; we further discuss this point in subsection B.3.1.
Alternatively, we propose to extract self-supervised features per box; we crop the image using the box coordinate before and feed the crop to the off-the shelf self-supervised model. In this work we use a ViT small model trained with the DINO Caron et al. (2021) self-supervised method. We produce results when using the two type of features.

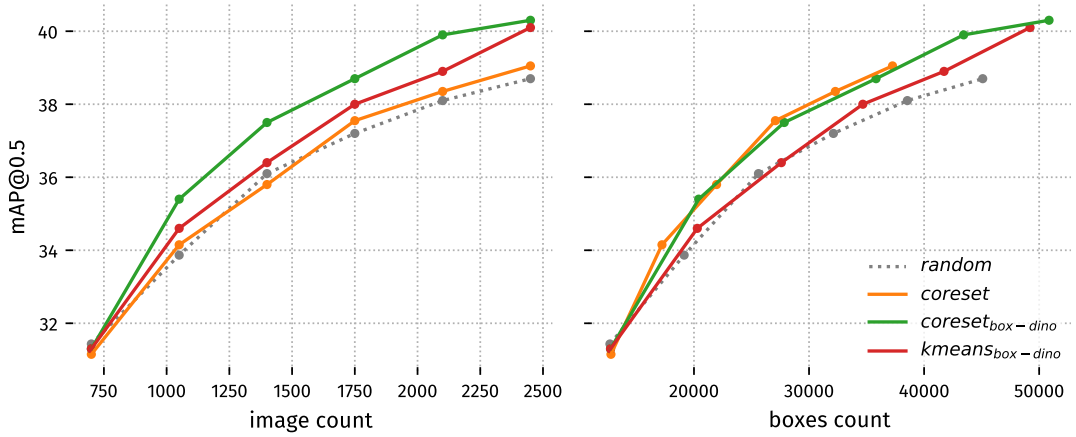
B.4.2.3 Evaluation

We evaluate here the different variations of the diversity-based method when using both types of feature detailed above. We compare the methods to classic image-level *coreset* Sener and Savarese (2018). We denote $coreset_{box}$ our box-level core-set when using the features obtained by average pooling over the l-previous-cycle model intermediate features. We note $coreset_{box-dino}$ and $kmeans_{box-dino}$ when using instead DINO self-supervised features with our adapted coreset and kmeans algorithms described above.

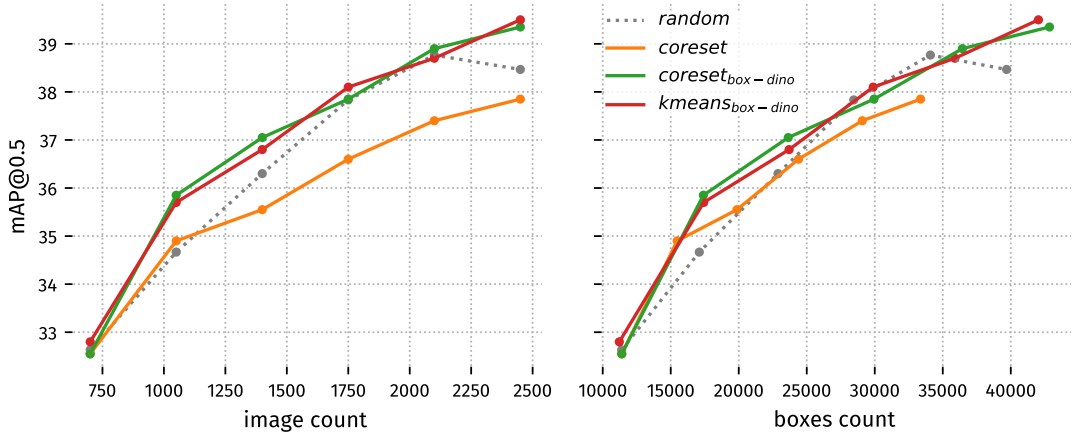
B.4.2.3.1 Experimental results. We report the results in Figure B.11. Here again we observe that the best methods vary depending on the datasets, which we know have very different



(a) VOC



(b) BDD100k



(c) VDP

Figure B.11: Evaluation of our box-level diversity-based acquisition function on the three datasets. The mAP@0.5 results are reported versus the number of annotated images at each AL cycles (left column), and versus the number of boxes annotated (right column) which also increase cycle after cycle.

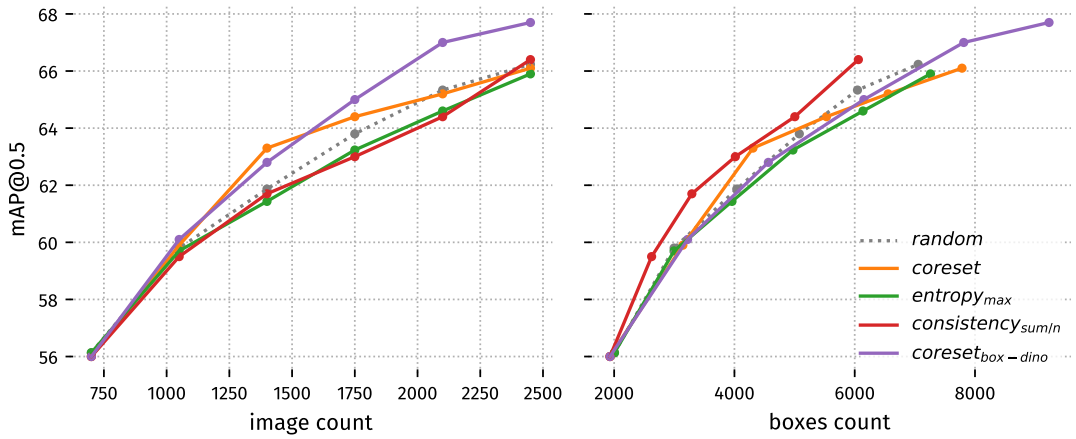
distributions. We first observe that the different strategies are better than a random selection on VOC when considering the image count, but this gain reduces when considering a box budget. We hypothesize this is due to VOC already having a homogeneous domain (well lit images, image composition with large centered objects) and relatively well balanced classes. On the other hand, BDD100k and VDP have different subdomains of scenes. Interestingly, on BDD100k, all three strategies investigated significantly improve over a random baseline both when considering the image count and the boxes count. BDD100k combines a series of challenges that are each a basis for the design of the reviewed AL method: a variety of domains and object appearances, imbalanced distributions, and a range of detection difficulty levels. As such, all methods contribute to the selection but this suggests a mixture of method might benefit even more.
The VDP dataset is centered around one central domain (daylight city center) with minor subdomains which do not weight heavily in the validation split either. As a result, vanilla coreset is unsuitable for this dataset. The scenes contain many objects with a wide variety of appearances and size, which is due to the usage of cameras with a large field of view. As a result, we suspect it is difficult to isolate images that can contribute significantly more than the others to the training of the model.
Between classic core-set and the K-Means sampling strategies, the first provides more consistent benefit and we suspect it may provide better asymptotic results (ie. on higher annotation budgets) since it explicitly avoids redundancy with already annotated data.

B.4.3 Comparison of active learning Methods

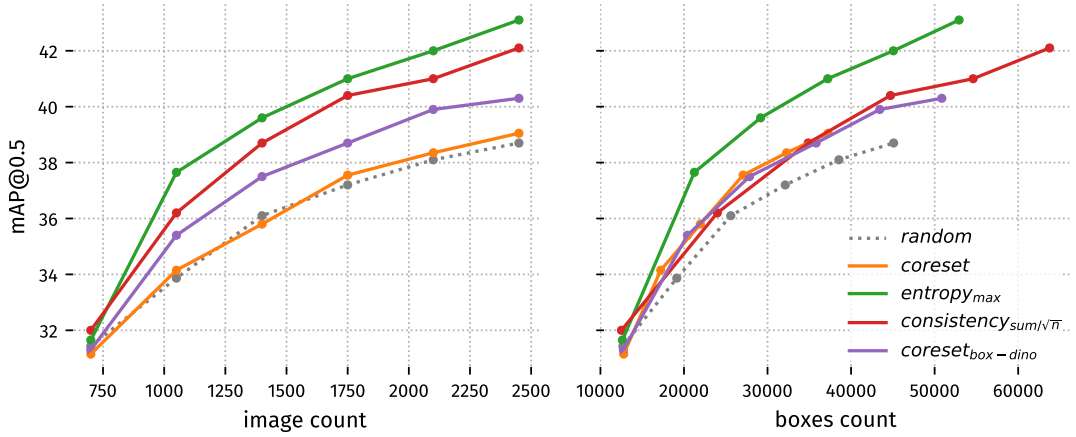
We finally compare our new designed acquisition functions to the results obtained with classical acquisition functions. Details about classical methods have been studied in a previous work (*random*, *entropy* and *coreset*). For clarity, we only report results obtained with the best variations per dataset of each method as reported in the previous section.
All results are summarized in Figure B.12. First, we note that the random baseline remains fairly competitive, in particular on the hard dataset VDP or when considering the box count metric on VOC. We attribute this result to multiple factors:

- To comply with existing benchmark protocols, academic publications use an older model (SSD) which behaves differently to ours (FCOS). FCOS performs better overall and is more robust, which causes the margin of improvement from AL contributions to be compressed.
- Active learning benchmarks focus on VOC and COCO datasets which are well curated images, balanced in class distribution and contain many large centered objects. This is in stark contrast with BDD100k and VDP which are composed of a variety of distinct domains (day/night, urban/suburban/highway, crowded/empty, etc.).
- Most published contributions on active learning for 2D detection focus on an image budget. Our experiments demonstrate that the results on a box budget can be widely different. We argue that the number of box is a better indication of the time spent annotating images.
- Due to the computational cost of experiments, most of our analyses are limited to 2450 images. As budgets grow larger, the performance/budget reaches its asymptotic limit. It is at this point that the cost difference in number of images (or respectively boxes) becomes more significant for a given detection performance target.

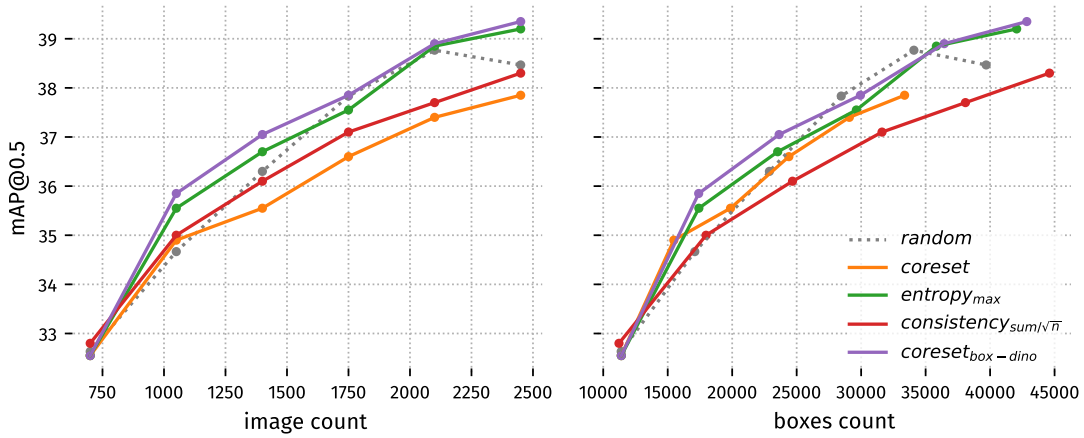
Overall, the box-level coreset (coreset$_{box-dino}$), using self-supervised features, provides a consistent improvement across all datasets, and never performs worse than a random selection. In comparison to the vanilla image-level coreset, it performs equally or better, especially on the basis of an image budget. Furthermore, visual inspection is facilitated as one can observe the



(a) VOC



(b) BDD100k



(c) VDP

Figure B.12: Comparison of active learning methods.

individual detections that caused the selection of an image, and each object bears an embedding vector which facilitates data exploration and manual review.
We believe the consistency approach provides pertinent scores about the difficulty to detect objects, but it remains undermined by the choice of box to image reduction method, to which it is highly sensitive. As it does not account for data redundancy, consistency acquisition is susceptible to select many difficult samples of a particular type, but this issue could be circumvented by updating the detection model more frequently and spending the annotation budget in smaller batches.
The max entropy method performs exceptionally well on BDD100k, but not on VOC. This setback is in contrast with the results observed using the same method on the same dataset but with a different detector such as SSD. The success of the entropy method will depend on a few factors that are not easily adjustable in an active learning scenarios:

- The rate of hard objects that are not spurious detections (adjusted with detection threshold).
- The proportion of misclassified object where the model is confident (ie. entropy is low).

To conclude, from our experiments, performances of acquisition functions highly depend on the nature of the dataset. We also know with our previous experiments that the detector and the scenario (initial labeled set size and budget) have an impact in the performance of these acquisition functions. The evaluation on the image-level budget as widely adopted in the object detection active learning literature also fails at selecting accordingly to the realistic human annotation workload estimated in number of boxes. This is shown in our results by the discrepancies between the acquisition methods performances when considering image counts and box counts. However, in the setup adopted in this experimental study, our box-level coreset approach benefiting of self supervised features from the external DINOv1 pretrained ViT small model, namely the $coreset_{box-dino}$ is particularly consistent and interpretable.

B.5. Seed Selection

B.5.1 The problem

The iterative process of active learning needs an initial labeled subset referred to as "seed", which the model is trained on in the first cycle. This initial set is typically chosen randomly in the active learning literature. In the batch 2 (report here), we studied the impact of the random seed selection and noticed that for a 700 images seed on BDD100K , the gap of mAP@0.5 between 2 different seed trainings of a YoloV5s detector could reach 2.4 points. The same observation has been made in this batch with the FCOS detector where we had a 1.8 points difference with 2 runs in the same setting. This gap is sometimes carried across the active learning cycles depending on the acquisition function.
Therefore, improving the selection of the seed can be crucial for active learning, and help achieve better performances at the first cycle but also throughout the AL process. Moreover, designing approaches to select data to annotate when the model has not been initialized also responds to scenarios with a single selection of data to annotate, where the model will only be trained once.

B.5.2 Seed selection design

When the initial seed is built, no annotated data nor trained detection model are available. Therefore, we propose to leverage self-supervised features–which are trained without any annotation– to get good image representations Caron et al. (2021); Chen et al. (2020b). In particular, we can use such features to perform a diversity-based selection in order to build a first set which already contains diverse scenes/objects.
A simple approach to perform a diverse selection at the image-level is to use a diversity sampling strategies (e.g. core-set, k-means) on image descriptors obtained with off-the-shelf features. Such a strategy has been well studied in the image classification context Lang et al. (2022); Pourahmadi et al. (2021). Provided that for the classification task, images contain only one object of interest, the features at the image level are good to characterize the informativeness of the sample. In Pourahmadi et al. (2021), the authors show that a simple K-means clustering algorithm works fairly well to build a good seed that covers diverse classes.
Here we are interested in the object detection task, which datasets are more diverse and typically depict several objects per image. We investigate using an image-level diversity-based sampling strategy as described before. But we also consider designing a seed selection specific to our more complex problem. In particular, following our experiments with box-level diversity strategies (detailed in subsection B.4.2) we propose to build a seed using a similar box-level approach.

B.5.2.0.1 Selection at the image level We first proceed with the simple image-level approach which follows these steps:

- Use a DINO pretrained ViT model to extract a global rich representation per image of the unlabelled dataset.
- Perform a K-Means clustering on these features and select the closest image to the center in each cluster with the number of clusters k being equal to the initial annotation budget.

We refer to this this seed selection method hereafter as $kmeans_{dino}$

B.5.2.0.2 Selection at the box level Here we want to design a seed selection adapted to object detection, and aim at producing a box-level diversity based strategy. For this, we adopt the approach detailed in subsection B.4.2, namely $coreset_{box-dino}$ (box-level coreset with DINO

embeddings). However, in order to use $coreset_{box-dino}$, we need predicted boxes likely around objects in order to produce object-specific features. During a classic active learning process, we have a trained model which is trained using the portion of annotated data from the previous step. That is not the case for the seed selection–all data are unlabeled and no model has been trained. In order to produce tentative boxes, we could use proposal algorithm such as *selective search* Uijlings et al. (2013) or unsupervised object localization methods Wang et al. (2023); Siméoni et al. (2023) which produce interesting object localization predictions (without having been trained on any labeled data). However, in order to evaluate the validity of our method, we first perform preliminary experiments and consider a *perfect scenario* where we have perfect boxes; we use here the ground-truth boxes. This experiment will serve us as *upper-bound*.
The box-level coreset approach implemented in this preliminary study does not include a filtering of outliers as the $coreset_{box-dino}$ described in subsection B.4.2. We refer to this seed selection method hereafter as $coreset_{gtbox-dino}*$.

B.5.3 Preliminary experiments

We produce here our preliminary experiments on the construction of such a seed. In order to compare the different methods, we evaluate our detector model trained solely on the selected seed. We consider two different budgets of 700 and 2000 images and report results on BDD100k in Table B.1, on VOC in Table B.2 and on VDP in Table B.3.

	700	2000
random	31.1	37.1
kmeans_{dino}	32.3	38
$\text{coreset}_{gtbox-dino}*$	**32.8**	**39.4**

Table B.1: mAP@0.5 results after training a FCOS detector on the seed images for the 2 bugdets on **BDD100k**. These results are the average of 2 runs.

	700	2000
random	54	63.7
kmeans_{dino}	**54.8**	**65.2**
$\text{coreset}_{gtbox-dino}*$	44.6	63.4

Table B.2: mAP@0.5 results after training a FCOS detector on the seed images for the 2 bugdets on **VOC07+12**. These results are the average of 2 runs.

We observe in Table B.1 and Table B.2 that the seed selection with both $kmeans_{dino}$ and $coreset_{gtbox-dino}*$ on BDD100k and $kmeans_{dino}$ on VOC surpass a random selection when considering a seed of both 700 and 2000 images. Suprisingly, in the case of VOC, the box-level strategy $coreset_{gtbox-dino}*$ underperforms the random selection, with a large drop when using 700 images. The surprising gap of 9.4 points could be explained by the propensity of the coreset method to select outliers objects since they are the least redundant with other data. Further experiments should investigate how to filter outliers, for instance using a filtering strategy like the one proposed in subsection B.4.2.
If the good results of $coreset_{gtbox-dino}*$ on VOC (+1.7 and +2.3 points respectively with a budget of 700 and 2000) are promising, they are computed with the ground-truth boxes and more

experiments should be performed of a realistic scenario, e.g. when using boxes obtained without annotation. This makes $kmeans_{dino}$ our best overall approach to-date to select seed images on BDD100k and VOC07+12.

	700	2000
random	32.7	37.7
kmeans_{dino}	32.4	37.5
$\text{coreset}_{gtbox-dino}*$	32.7	37.6

Table B.3: mAP@0.5 results after training a FCOS detector on the seed images for the 2 budgets on **VDP**. These results are the average of 2 runs.

Regarding the VDP dataset, we report results in Table B.3 and we can observe that the proposed methods achieve similar results to a random sampling, therefore bringing no contribution. This result contrasts with those observed on BDD100k and VOC. The VDP dataset is particularly challenging for most active learning methods we have tested in subsection B.4.3, it is possible a diversity-based criterion is not the most suitable sampling strategy for this dataset. As for the box-level method $coreset_{gtbox-dino}*$, the presence of outliers might also hinder potential benefits, but more investigation are required for confirmation.

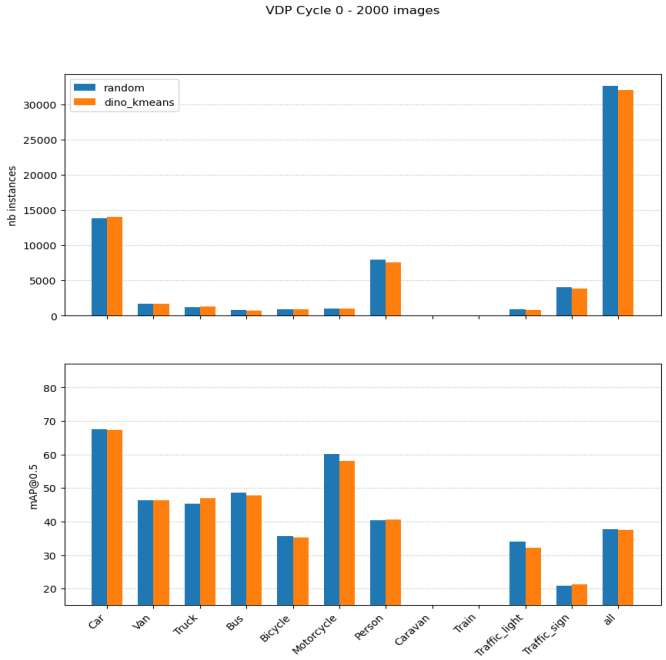
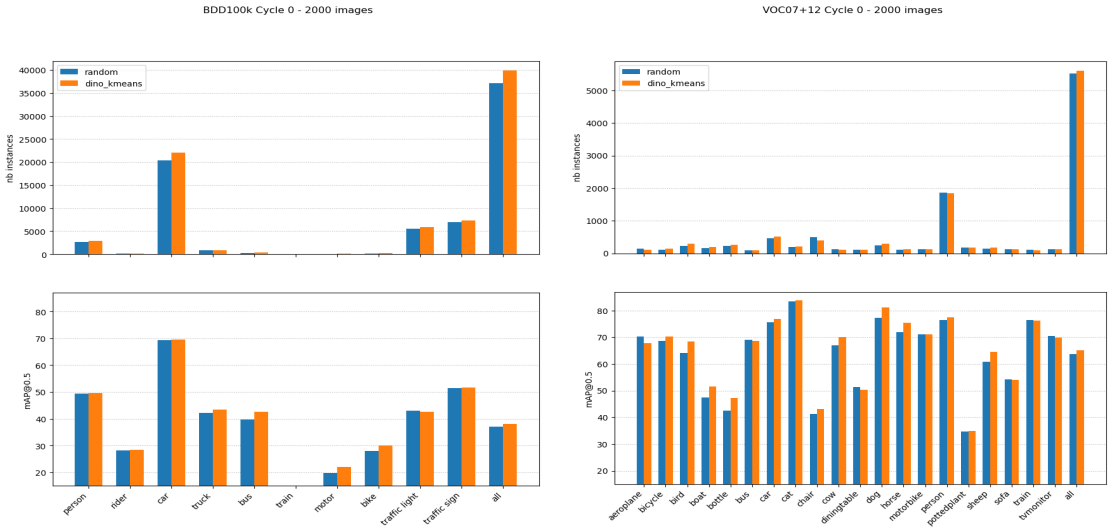


Figure B.13: Comparing instance distributions of seed images and mAP@0.5 score per class after training on those seeds of 2000 **VDP** images.

B.5.3.0.1 Per-class investigation In an attempt to better understand results, we compare in Figure B.13 and Figure B.14 the distributions of seed images per class mAP@0.5 scores. In Figure B.14, we observe that for both BDD100k and VOC, $kmeans_{dino}$ globally selects images with have more instances and in particular more of the less populated classes : *chair*, *bottle*, *boat*



(a) BDD100k (b) VOC

Figure B.14: Comparing instance distributions of seed images and mAP@0.5 score per class after training on those seeds of 2000 images.

for VOC and *motor*, *bike*, *bus* for BDD100k. Accordingly, the model performs better on those classes and the performances on more populated classes do not decrease, which overall makes the model perform better. We find in Figure B.13 that the behaviour of $kmeans_{dino}$ on VDP is not similar, and in fact almost the opposite. On less populated classes such as *traffic light*, *bus*, *bicycle*, $kmeans_{dino}$ fails to select more instances. Globally, on the total number of instances in the 2000 images selected, $kmeans_{dino}$ seed selection contains fewer objects than the *random* method and the mAP@0.5 performance follows the same order. This opposite behaviour of $kmeans_{dino}$ on VDP than on BDD100k and VOC gives one explanation of the under-performance on this dataset compared to the other two.

B.6. Conclusion

Throughout this extensive experimental study we establish that performance claims of the state-of-the-art active learning methods are not trivially generalizable across datasets and models. We analyze multiple methods and exhibited important properties and prerequisite for their applicability.
Moreover, the performances/cost ratio of several methods remains close to a naive random selection when accounted in terms of number of boxes. We attribute this issue in part to the fact that useful and redundant objects coexist within the same images. As a result, more attention should be put on the method to reduce object-level selection scores into image-level scores. This also suggests that for the 2D detection task, active learning would be advantageously combined with pseudo-labelling and smart annotation methods, so that its contribution is not downplayed by the proportion of easy examples in a dataset.
Between the selection methods, maximum entropy is the easiest to implement and performs remarkably well on some datasets. We observed very pertinent selection from the consistency-based method regardless of the dataset. However this second method is sensitive to the way object-level scores are aggregated into image-level scores. Overall, we find that the box-level the coreset diversity criterion provides the most consistent and interpretable sampling strategy.
We also experiment to select more efficiently the initial set used to bootstrap the model for the active learning process, which is a question of high interest in active learning, especially in scenarios with few annotation. We noticed that a simple K-means approach using image representations from a self-supervised external model was efficient to build a better performing seed subset than the random one for two datasets studied. Our attempt to optimize this seed selection for the object detection task by using box-level representations needs further investigations.

C. Class Incremental Learning for 2D detection

C.1. Introduction

As industry's interest toward autonomous vehicles grows, efforts must be made to address the challenging computer vision problem of *Object Detection (OD)* where a model is train to predict bounding boxes around objects of interest and class them correctly.
Deep convolutional neural networks (CNNs) have been showing great success in addressing the problem of Object Detection. In particular, two types of methods stand out because of their state of the art results. One-step methods like SSD or YOLO models are among the fastest and two-steps methods like R-CNN have the best accuracy. More rcently, Vision transformers are also a solution. Most commonly, the desired classes are learn all-together in a single *Joint training*. The whole dataset must be available at time of training for the model to learn its parameters. This lacks the capability of adapting to new data arriving afterwards, without re-training from scratch.
Fine-tuning is an approach in which the weights of a pre-trained model are trained on new data. However, if one straightforwardly applies fine-tuning, the model will adapt too well to the new data and its accuracy on previous classes will severely drops. This phenomenon is already well known as *catastrophic forgetting* and results in general in a low mean accuracy for the model.
Incremental learning methods aim to alleviate catastrophic forgetting. More specifically, *task incremental* or *class incremental* learning is an incremental scenario where new classes or tasks (set of classes) can be learned by the model sequentially. It means that a model that knows previously learned classes will learn new classes, with no need of the data related to the previous classes. It relaxes the assumption that all the data is needed at a time for one large and long training for all the classes. Training may be done class by class or task by task. Incremental learning receives more and more focus from researchers as it is part of the way towards life-long evolving AI, and as it gives more flexibility concerning dataset memory management and training duration.
However, addressing catastrophic forgetting is not a trivial problem and research has not found yet a method where class incremental training reaches joint training performance. The subject has been studied more in depth for classification. The problem is more complex for object detection, as baseline classification typically aims to identify an object in an image, whereas object detection aims to identify and localize multiple objects in an image.
While Incremental Learning is now fairly studied in the literature for use-cases like image classification, the subject is yet not mature for Object Detection. In particular, where IL in classification shows that replay methods give the most robust results, there is almost no investigation on the subject for Object Detection.
In this report, we present two methods that alleviate catastrophic forgetting when incrementally learning new classes for an object detection problem applied to the use-case of road traffic vision. The first method uses the model to predict *pseudo-labels* (PL) of the old classes in the images of the new classes. These pseudo-labels are then used during training along the new classes labels, resulting in lowering catastrophic forgetting. The second method uses the replay of a fixed-size buffer storing previous task images to help retaining the accuracy on the previously seen classes. We present as well too distinct scenario where new task images do and do not have old class labels. While pseudo-labeling is only needed in the second case, the replay method can be used in both situations. The model is YoloV5, as it already utilised in a lot of real-life use-cases, because of its state of the arts results in term of inference speed, resource usage and performance. We train our model on both BDD100k and VDP datasets.

C.2. Background & Related work

In this section we introduce important definition to understand our work and we present some elements of context to locate Incremental Learning for Object Detection among related works. For an extended State of the art overview, refer to previous *Confiance.ia* report *SOTA in incremental learning for object detection and semantic segmentation*.

C.2.1 Definitions

In the literature, the terms "incremental learning" (IL), "continual", "sequential", and "lifelong" learning are employed sometimes as synonyms, to denote methods of machine learning that extend the model's knowledge and capabilities with increasingly more new input data available, over time. In other words, the model is further trained to accommodate for these new data. Class-Incremental Learning (CIL) enables the model to incorporate the knowledge of new classes, hence executing a retraining step each time a new set of classed should be learned. Here we denote the set of classes to be learned as a task.
To benchmark, typically class incremental learning methods are compared to an upper bound scenario called "joint training" or "offline", and to an lower bound scenario often called finetuning. The joint training is not an incremental training because at every task, the model is trained from scratch with all available previous and current data. This way it gives the best-case scenario for a given dataset. By finetuning we denote the re-training of the pre-trained model on only new task data, without any specific methods to counter catastrophic forgetting. For our work, we add the replay and pseudo-labeling methods on top of the basic finetuning, to form an incremental learning method. Offline training typically denotes training with all classes/tasks at once.

C.2.2 Brief Incremental Learning State of The Art

There are several ways to categorise the incremental learning approaches and the literature has yet to reach a terminology consensus. Some important overviews are in Delange et al. (2020); Parisi et al. (2019); Maltoni and Lomonaco (2019); Masana et al. (2020); Zhou et al. (2023). In general, methods are split in three main, fuzzy categories: *regularization* strategies, *rehearsal/replay* strategies, and strategies that imply the *growth of the network* (branches or entire networks) to accommodate new knowledge.
Regularisation strategies Delange et al. (2020); Parisi et al. (2019); Maltoni and Lomonaco (2019); Masana et al. (2020) impose constraints on the update of the neural weights. Extra regularization terms are introduced in the loss function, consolidating previous knowledge when learning on new data.
Replay strategies Delange et al. (2020), also called *rehearsal strategies* Maltoni and Lomonaco (2019), store samples in raw format or generate pseudo-samples with a model (generative or not). Previous tasks samples are either interleaved with new samples during the new training, or to constrain the loss optimization in the new training to prevent previous task interference.
In addition to these two, the overview in Delange et al. (2020) distinguishes also *parameter isolation strategies*, which dedicate different model parameters to each task, to prevent any possible forgetting. When no constraints apply to architecture size, one can grow new branches for new tasks, while freezing previous task parameters, or even dedicate a model copy to each task. Maltoni and Lomonaco (2019) includes these in *architectural strategies*, where specific architectures, layers, activation functions, along with weight-freezing strategies are used to mitigate forgetting. Furthermore, Maltoni and Lomonaco (2019) includes in the architectural strategies

also the *dual-memory models* attempting to imitate hippocampus-cortex duality, which could be seen as a rehearsal strategy.
In a more bio-inspired view, Parisi et al. (2019) denotes replay strategies as *complementary learning systems and memory replay* and parameter isolation strategies as *neurogenesis*.
Two other Confiance.ai reports address (among others) incremental learning in different contexts, Nabhan et al. (2021) and Montejano Villabla et al. (2021). Nabhan et al. (2021) discusses on incremental or continual learning methods in the context of long training-time and data-set update, mostly for image classification and time series. Montejano Villabla et al. (2021) includes a chapter on the state-of-the-art on incremental learning for image classification.

C.2.3 Incremental learning for object detection

The current solutions for incremental learning in object detection and semantic segmentation are significantly fewer than for classification and include mostly distillation methods, with or without a buffer of samples from past tasks. Besides that, the literature include one rehearsal approach that utilise a GAN to generate synthetic samples, or simply web-crawled data that matches the classes, and one parameter isolation approach. Furthermore, the weights of the encoder or the backbone of the network may be consider "frozen", by some approaches, i.e., trained once and never retrained. There are more approaches targeting semantic segmentation than object detection.
Existing work in incremental learning for image classification suggests that distillation does not scale up to a large number of classes Belouadah et al. (2021). Whereas it is unclear if this claim translates to other tasks besides image classification, what is sure is that the scaling problem is yet to be addressed for object detection and semantic segmentation. The number of tasks learned is small, typically 5, with a maximum of 10, in two approaches Cermelli et al. (2022); Maracani et al. (2021).
Furthermore, the data-sets are limited to rather simple ones, as PASCAL VOC, and ADE20K, with few exceptions, mostly in the semantic segmentation case Hu et al. (2018); Tasar et al. (2019). Industrially utilized networks, e.g., YOLOV5, and data-sets, (many classes, unbalanced, large training time) are still to be investigated. In addition, the cost of the method is typically not discussed, with small exceptions that report consequent numbers Tasar et al. (2019), however without discussing potential optimizations.

C.2.4 On Replay methods for Object Detection

Among the IL methods families, it has been established that *Replay methods* have a lot of potential because they cover the most commonly studied Incremental Learning Desideratas, as follows (reproduced from Delange et al. (2020)):

- **Constant memory.** To avoid unbounded systems, the consumed memory should be constant w.r.t. the number of tasks or length of the data stream.
- **Forward transfer** or zero-shot learning indicates the importance of previously acquired knowledge to aid the learning of new tasks by increased data efficiency.
- **Backward transfer** aims at retaining previous knowledge and preferably improving it when learning future related tasks.
- **No test time oracle** providing the task label should be required for prediction.
- **Graceful forgetting.** Given an unbounded system and infinite stream of data, selective forgetting of trivial information is an important mechanism to achieve balance between stability and plasticity

This is the reason why focus in particular on replay methods. However, the only work addressing Incremental Learning with Replay for OD, *RODEO* Acharya et al. (2020) uses a two-stage detector in an online setting and shows interesting results on PVOC2007 & MSCOCO. For One-stage detectors, this is quite an open research area.

To sum-up, most works on Incremental Learning address image classification rather than Object detection and only a few studies present mostly distillation techniques applied on two-stage detectors Hao et al. (2019). On our side, we chose to focus on the industrial-sized one-stage detector model Yolov5. In addition, as works in image classification show that replay allows better scaling than distillation or parameter isolation, we did focus on this method to address our IL problem.

C.3. Methodology

C.3.1 A class incremental Object Detector

An Object Detectors model is trained to find known objects on images. The model should classify those objects and predict bounding boxes around it. It is considered as supervised learning as numerous batches of annotated images are used to optimize the model's weights with a gradient descent algorithm.

C.3.1.0.1 YoloV5 (from `https://github.com/ultralytics/yolov5`) is the one-stage detector model we used to address the object detection problem. The main contribution of this 5th version of Yolo was the translation from the Darknet framework to the more user-friendly PyTorch framework. Yolo is fast because it uses only one neural network to process the data from images to box predictions compared to other two-stage R-CNN based models that have a Region Proposal Network combined with a Detection Network. Yolo's network is made of a *Backbone*, a *Neck* and a *Head*. The first one is a convolutional neural network that aggregates and forms image features at different granularities. The second part is a series of layers to mix and combine image features to pass them forward to prediction. The last one consumes features from the neck and takes box and class prediction steps. During training, yoloV5 uses data augmentation which means that the models sees images that are transformed and mixed together to improve the diversity of the data and improve performances. The output of YoloV5 gives - for 3 image resolution and for different parts of the image - multiple boxes predictions alongside an objectness score and a class score. The loss function is built taking into account the Intersection over Union (IoU) of the boxes, the objectness and class score. In order to filter the effectively predicted boxes during inference, an algorithm called Non Maximum Suppression (NMS) is used to select only the most relevant boxes among redundant and less confident predictions. When we validate the model on the validation dataset, predicted boxes are compared to the ground truth labels. For a fixed IoU threshold, the Precision-Recall(PR) curve is built by sampling many confidence thresholds from 0 to 1. Precision is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that were retrieved among ground-truth ones. Those metrics are used to compute mean average precision (mAP), the relevant metric that is used in object detection to compare models performances.

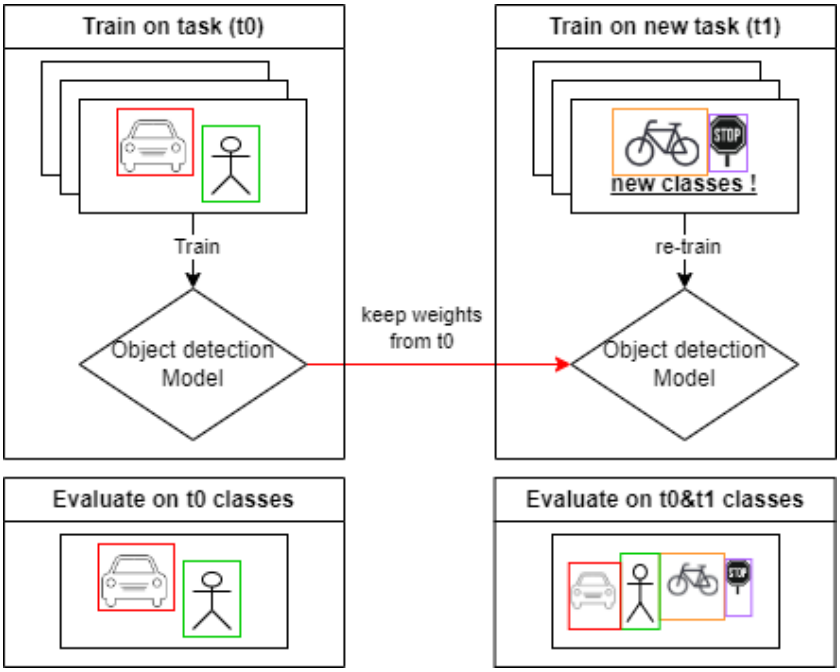


Figure C.1: In an incremental setting, new tasks of data are used to update an existing model. With class incremental learning, new classes are learnt in the new tasks and the model must keep being accurate on every class previously seen.

C.3.1.0.2 Class incremental learning relaxes the assumption that all the data for all classes is available when first training a model. It allows the Object Detection model to be trained on a subset of the classes and data, and then to be updated with newly available data with new classes (figure C.1. After every iteration of training, we evaluate the model on every seen class. The risk with Incremental learning is that the model forgets the past class when learning the new ones. Our Class Incremental Learning method for object detect try to counter that catastrophic forgetting effect.

C.3.1.0.3 Finetuning with and without old labels It is important to note that in the Object Detection problem, an image can contain multiple instances of different classes. Those classes can be **current/new classes** if they belong to the task being learnt, **old/past classes** if they were

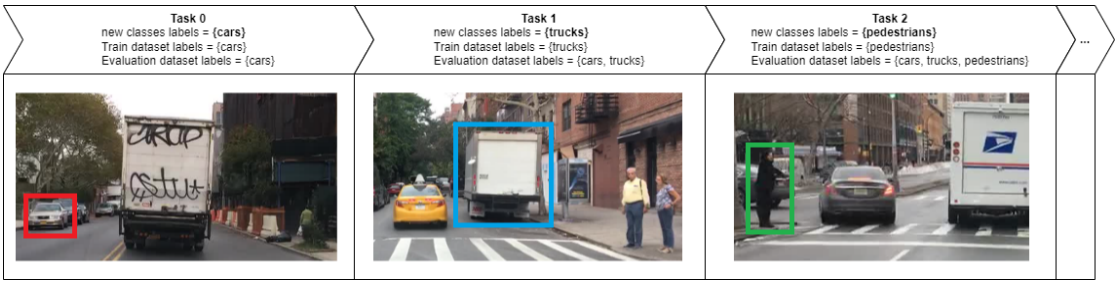


Figure C.2: Images from train dataset, with labels according to scenario finetuning no old labels (ftnol) describes an IL setting for Object Detection where the images of the new tasks do not have labels for old classes.

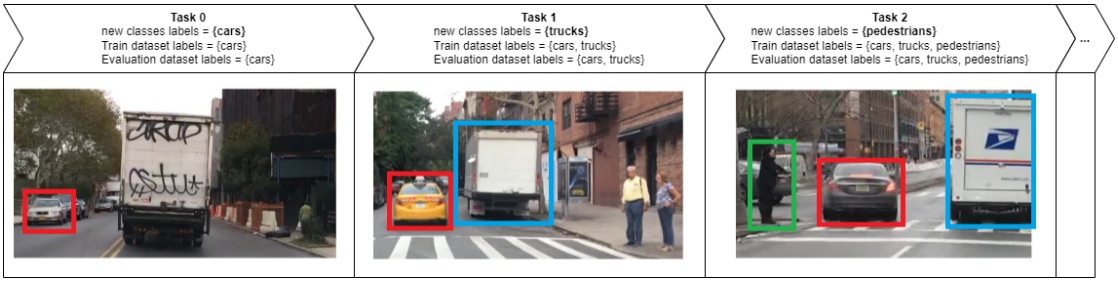


Figure C.3: Finetuning with old labels (ftwol) describes an IL setting for Object Detection where the images of the new tasks do have labels for old classes.

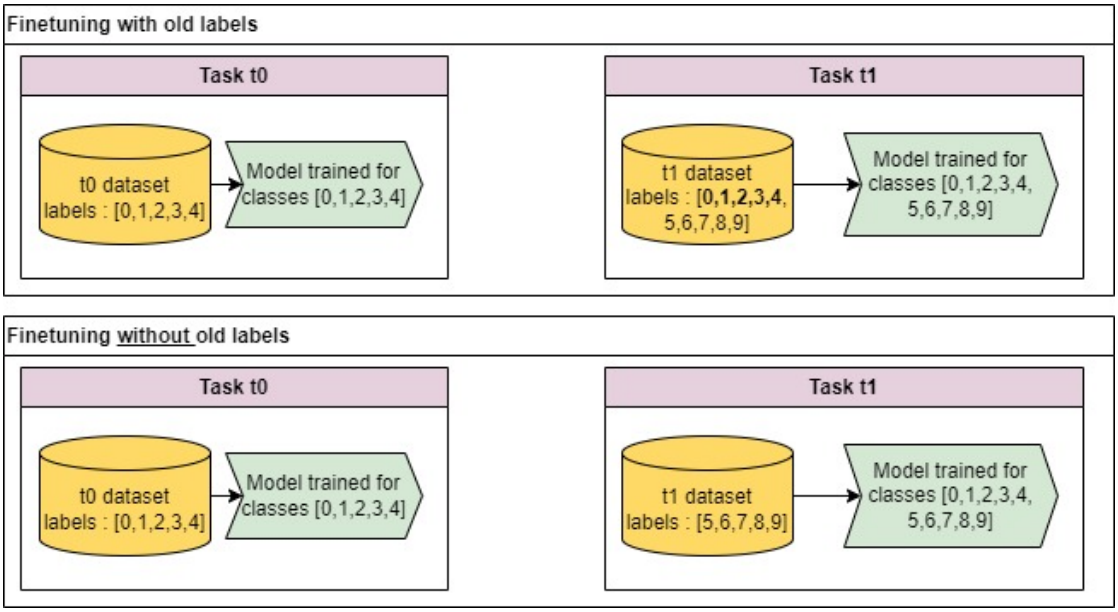


Figure C.4: 5x2 classes incremental learning differences when new task images have or do not have old labels

seen in previous tasks or **future classes** if we know that they will be learnt in future tasks. In order to understand our IL scenarios, it is important to find out which classes are included in which tasks because an image of a given task only have ground truth images for the included classes.

We note that an image can contain instances of classes that will be maybe learnt in future tasks, and those instance would not be annotated. In this case the model may learn false negatives of the futures classes. Nevertheless, during our experiments, it did not seem to harm the performance of those future classes, which got eventually learnt.

Two distinct scenarios for the old class labels can be evaluated when learning new tasks. We define the scenario finetuning with old labels (ftwol, Figure C.3), where the new task images do have old class labels, and finetuning without old labels (ftnol, Figure C.2), where the new task images do not have old class labels. We studied those scenarios distinctly as they show drastically different results, the second one being harder than the first, as the old class labels in the new task images help not forgetting the old classes while learning the new tasks.

C.3.2 Pseudo Labeling

We present here the first method we apply over basic finetuning in order to lower catastrophic forgetting when there is no labels of old classes in the images. In this scenario, the images used to learn new classes can also contain instances of the old classes. Those unlabeled instances may confuse the network as it may learn false negatives. However, the presence of previously learned object types can be used to reduce forgetting. Our approach is to use the model trained on the past tasks to predict pseudo-labels for the past classes in the new classes' images. Before each additional training on new task, we compute the pseudo-labels for all the task dataset at once, before any form of data-augmentation. The model predicts pseudo-labels for an image of the subset by making a conventional inference on this image using the last version of the model that was trained on the older classes. This process involves the non-max-suppression (NMS) algorithm which filters the boxes to keep. We used NMS with IoU threshold at 0.45 and Confidence threshold at 0.25 (default values for inference). The predicted boxes for old classes instance are used as pseudo-labels. Finally during incremental training of new classes, both ground-truth labels of new classes and pseudo-labels of old classes are used. We hypothesize that the effectiveness of this method is related to the proportion of new and old classes in the task dataset along with the capacity of the model to generate good pseudo-labels. Pseudo-labels are good for a class if the accuracy on this class was already good, but is also strongly determined by the NMS algorithm. For example, a high confidence threshold will lead to less but more precise pseudo-labels. On the contrary a low confidence threshold will lead to more and less precise labels. The investigation of the NMS configuration that brings the best performance to the network is sujbect to future work.

C.3.3 Buffer Replay

The second method to counter catastrophic forgetting is the replay of a fixed-size buffer of past images while learning the new data, as illustrated on Figure C.5. The images of the class instances from the previous tasks can help the model in not forgetting them. In Object Detection, we store along an image the labels of the classes that were associated to it. During our experiments, multiple buffer sizes are experimented with.

Two aspects need to be defined when designing a buffer-based replay strategy. The first is which (and how many) samples to store in the buffer. The second is, when the buffer is full, which samples to remove from the buffer.

First, we fill the dataset at each task with random images taken from the task dataset. The number of taken images is defined by the buffer size which can be adjust depending on the memory we can allow to it. This leads to a trad-off between the memory usage and the gain in performances added by the buffer. When learning the next task, the images from the buffer are added to that task's dataset and then naturally mixed into the training batches.

Second, we tested two different buffer replacement strategies :

- With the replacement strategy **"all"**, the buffer is cleared after each training session and is then fully filled by random images from the new task. this way, all images are replaced and we always have in the buffer images from the last task.
- With replacement strategy **"balanced"**, we replace $1/n$ of the buffer images with random images from the new task, n being the number of currently learnt tasks. This way, we statistically keep $1/n$ images from every seen task.

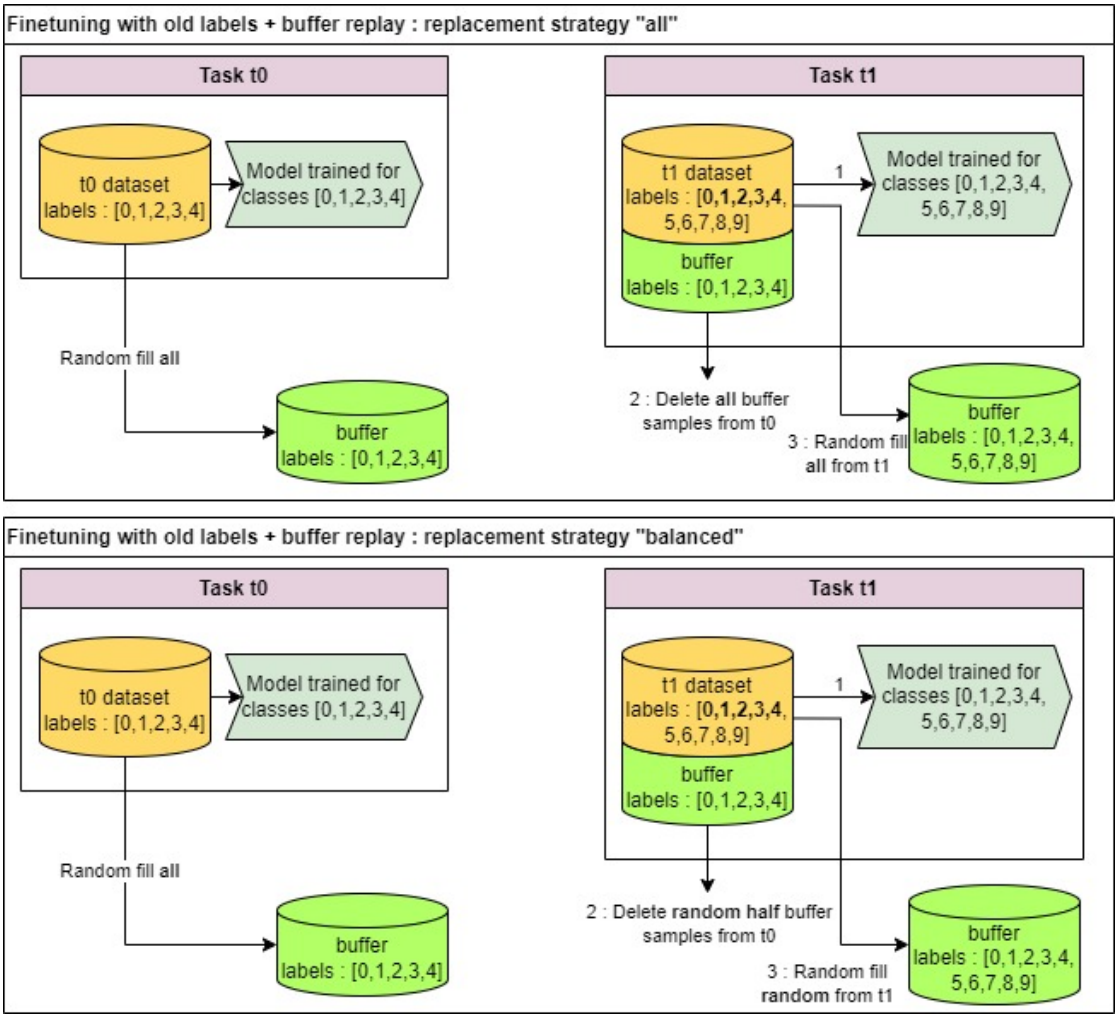


Figure C.5: 5x2 classes incremental using the replay of a fixed size image buffer

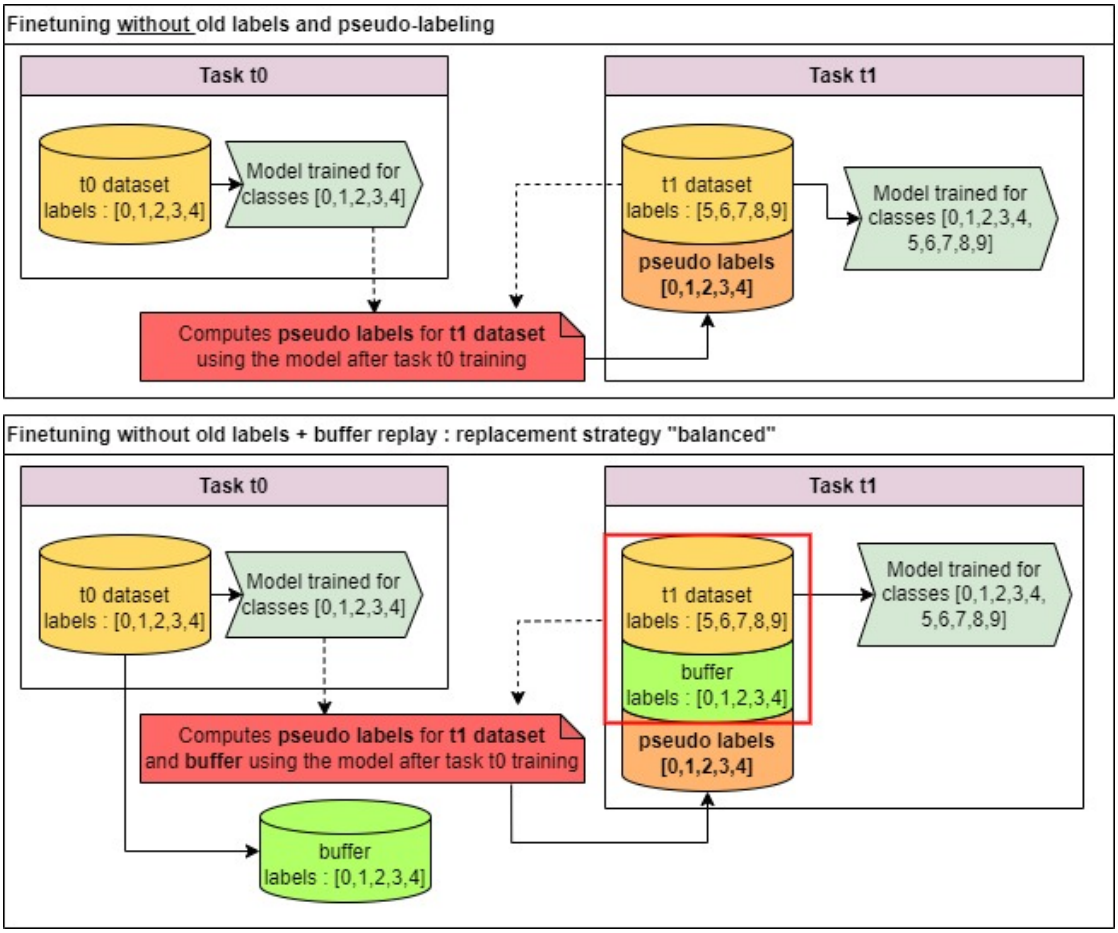


Figure C.6: 5x2 classes incremental using pseudo-labeling

C.3.4 Combined pseudo-labeling with replay

Our last take on incremental learning methods for object detection was to combine pseudo-labeling with buffer replay. As illustrated on figure C.6, we first add the replay buffer to new task dataset, and then we use the model to pseudo-label both the buffer and task dataset.

C.4. Experiments and Results

C.4.1 Dataset and Evaluation Metrics

In this work we employed two datasets, described as follows.

C.4.1.0.1 BDD100K Dataset (https://www.bdd100k.com/) consists in 100000 images of the traffic at different times of the day and under different weather conditions. It can be used for different applications, but here we use the data dedicated to the object detection problem. What we have is folders with 70K images for the train database and 10K images for the validation database. For each image, there is a text file containing one line for each label. On each line there is an index for the class and the normalized position and dimension for the box. Figure C.7 shows the ten labeled classes in the dataset. The number of class instances in the images is

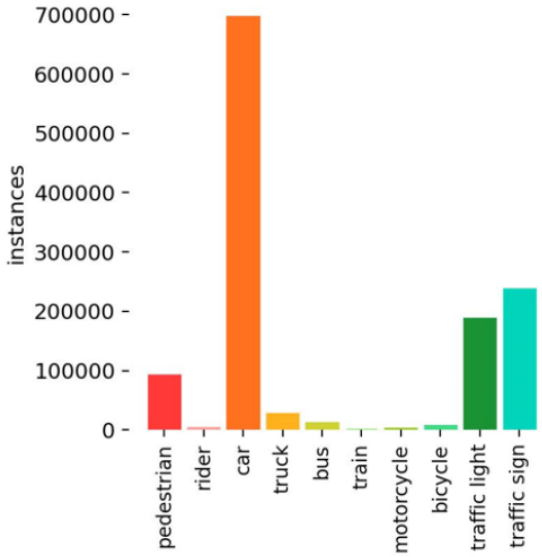


Figure C.7: Number of instances for each classes in the BDD100K train dataset

strongly unbalanced between classes.

C.4.1.0.2 VDP dataset has been shared by the industrial partner Valeo and contains almost 100K train images and 30000 test images of the traffic at different times of the day and under different weather conditions. As in BDD100K, the number of class instances in the images is strongly unbalanced between classes. Figure C.8 shows the 13 labeled classes in the dataset. For a better integration and comparison with BDD100k, we worked with only 10 of the 13 classes by removing the classes trailer, caravan, and other.

C.4.1.0.3 mAP50 and per-class mAP50 The mean average precision measure the ability of the model at having both a good precion and recall at given IoU threshold. We will use mAP at 0.5 IoU as it is the most common metric. Because we want to see how the model permforms on the different tasks classes, we will also have a peak at per-class mAP50.

C.4.1.0.4 Label instances by type We saw that we use three different types of label during our training. First, true-labels come directly from the dataset, second buffer-labels come from a buffer and thus from the previous tasks, and third the pseudo-labels are predicted by the model before training. As the datasets are unballanced, it is interesting to see how many objects of each class are in the dataset at each task, because this highly influences the final mAP result.

C.4.2 Experimental Settings

C.4.2.0.1 Dataset split We explain here how we pre-processed the dataset in order to construct two class IL scenarios. The scenario "*10x1c*" is a 10 tasks scenario with one class learnt per task, and the "*5x2c*" scenario is a 5 tasks scenario with two classes learnt per task. For a n tasks scenario, we randomly and evenly split the dataset in n *task subdatasets*. The result of this random split is that in a task dataset there might be (1) images that do not have objects from the current tasks, but have other objects, as well as, (2) images that have the objects of the current task but also other objects. In the case "no old labels", for a given task dataset we remove every

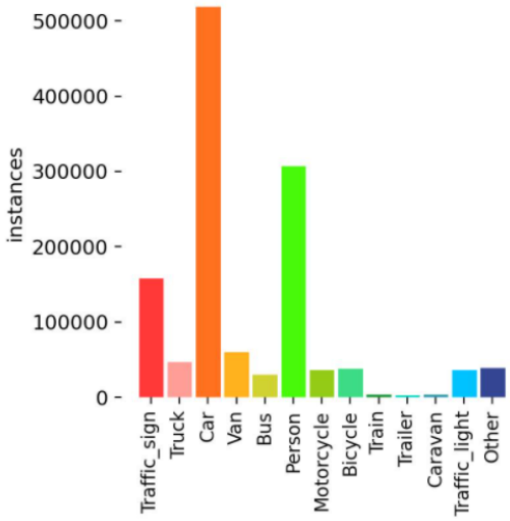


Figure C.8: Number of instances for each classes in the VDP train dataset

label of the classes not included in the task. When doing this, some images a left without any labels and so are discarded.

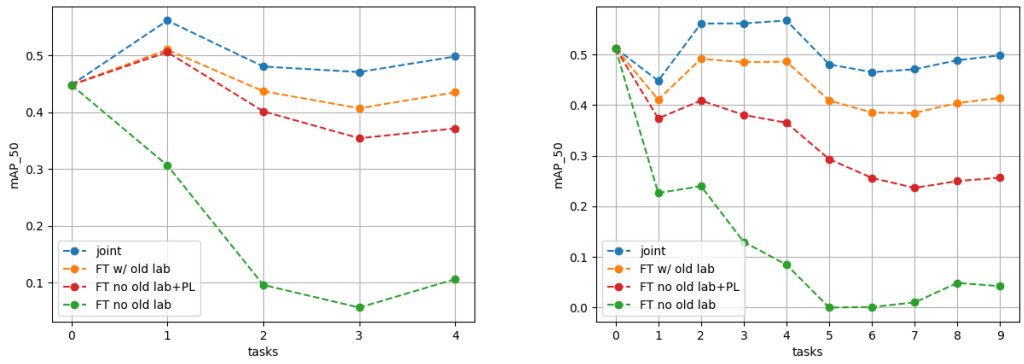
C.4.2.0.2 Model details There are multiple versions of the YoloV5 model. We chose *Yolov5s* because it is smaller and is faster to train, but that still offers good performances. YoloV5s model was pretrained on the dataset COCO. The code starts as a fork of the official github version that can be found at https://github.com/ultralytics/yolov5. There is quite a lot of code because it provides a lot of options to run and evaluate the model. As the code does a lot and is a bit complex, we did our best not to change its global structure to add our method. Experiments from Confiance's batch 2 show helped us in finding training parameters that allowed a good convergence of the model and a reasonable training time. We incrementally train our models using the provided We use batch size of 64 images, a cos-lr scheduler and the AdamW optimiser. We compare all training with only 50 epochs to slightly improve the training speed. We also disabled the mosaic data augmentation for the BDD100k dataset. With this parameters, we see an mAP50 of 0.52 for the joint scenario. We compared our joint baseline with other object detection models evaluated on BDD100k, from the GitHub repository at https://github.com/SysCV/bdd100k-models/tree/main/det. Here, the other state of the art Faster R-CNN model shows about 0.56 mAP at 0.5 IoU so our chosen Joint baseline seems a relevant compromise compared to state of the art.

C.4.3 Results & Analysis

In the following, we present the results of task incremental training scenarios with 5x2 classes and 10x1 class, on both BDD100k and VDP dataset. We discuss how our methods improve either finetuning without old class labels or finetuning with old class labels. The main figures show the mAP results of the model after training for each task and, toward a more in-depth comprehension of the results, additional figures will present per-class mAP and the number of class instances found per label type (true label, pseudo-label or buffer label).

C.4.3.1 Finetuning without old class labels + Pseudo-labeling

In the first experiment, we apply pseudo-labeling over finetuning without old class labels and get results Figure C.9a and Figure C.9b. On BDD100k, on both 5x2 and 10x1 scenarios that pseudo labeling improves significantly the mAP at each task, with +0.26 and +0.22 respectively. To explain the results, the number of pseudo-labels are represented in Figure C.10a. These pseudo-labels add to the true labels present in the tasks' datasets, presented in Figure C.10b. For reference, the number of true labels for the cases "joint" and "finetuning with old labels" are presented in Figure C.10d and C.10c.
We confirm by looking at per-class mAP in Figure C.11b that the classes are not catastrophically forgotten with pseudo-labeling, whereas without PL (Figure C.11a), they are. Importantly, we note that the main reason for global mAP loss is insufficient and unbalanced data from the dataset leading some classes to have lower base mAP than others.
As shown on Figure C.12b and xC.12b, we confirm our results with pseudo labeling on the VDP dataset with a final mAP gain of 0.27 for 5x2 and 0.2 for 10x1.



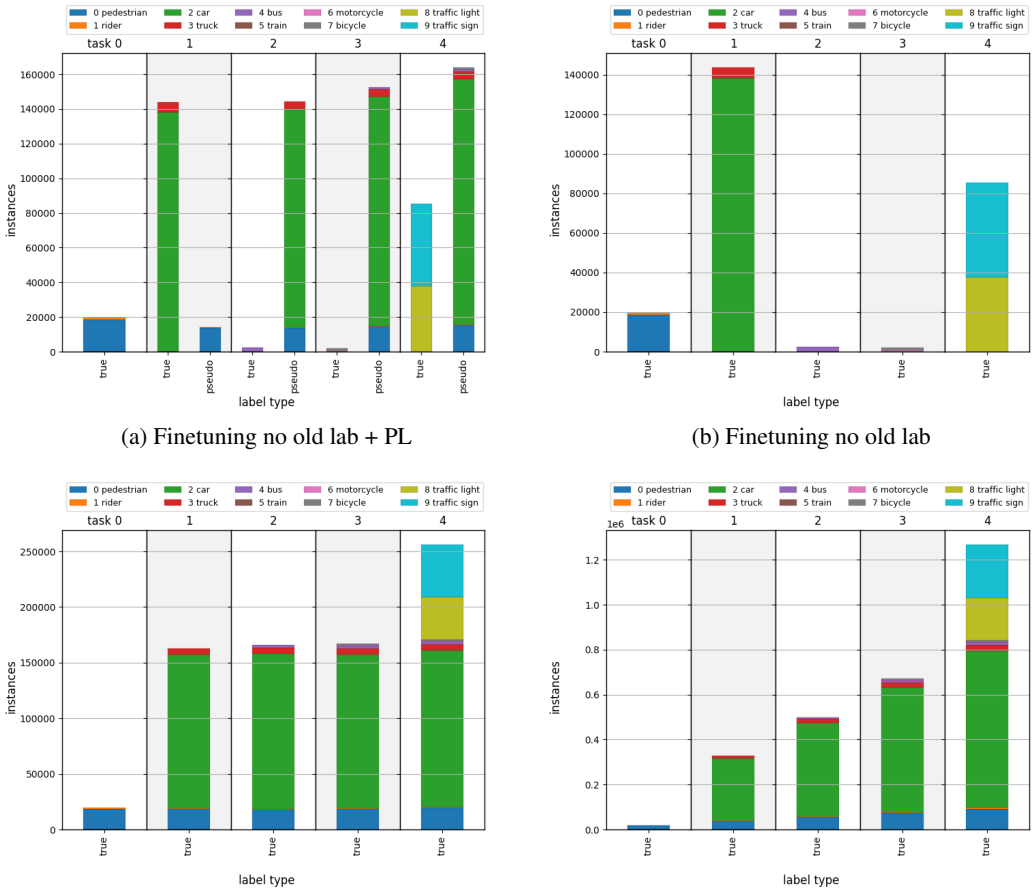
(a) 5x2 classes

(b) 10x1 classes

Figure C.9: mAP50 results of yoloV5 on BDD100K, class incremental training, with scenarios joint and variations of finetuning

C.4.3.2 Finetuning without old class labels + Replay

In this section we add replay to a finetuning scenario without old labels. The replay uses a fixed-size image buffer with sampling strategy balanced. On BDD100k, we present the results for 10x1 classes and 5x2 classes on Figures C.13a and C.13b. We first see that final mAP increases with the buffer size. Only for buffer size 10000 for 10x1 or 20000 for 5x2, the mAP drops a little. With a big enough replay buffer, the mAP is better than the variant with pseudo-labeling, namely a buffer size of 5000 leads to +0.2 mAP increase for 10x1 and a buffer size of 10000 to +0.24 for 5x2. We see as well an interesting spot where the replay exceeds pseudo-labeling and almost match finetuning with old labels on scenario 10x1, after tasks 1. It means that having the previous tasks images with true labels of the previous classes highly helps to retain mAP in this situation. Figure C.14 presents the proportion of labels of each classes kept into the buffer for a 5000 images buffer replay, on 5x2. In that configuration, the figure C.15 shows that the classes are not catastrophically forgotten after each task thanks to the replay. Here again, the main reason for global mAP loss is insufficient and unbalanced data from the dataset leading some classes



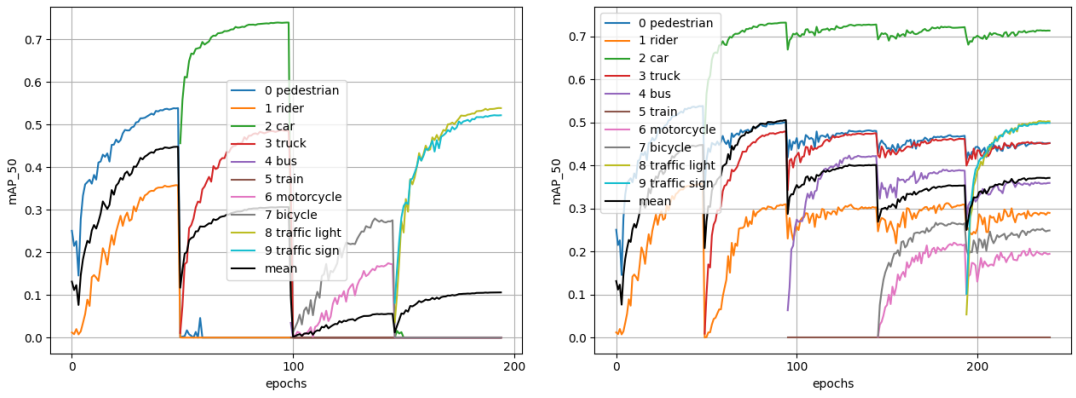
(a) Finetuning no old lab + PL

(b) Finetuning no old lab

(c) Finetuning w/ old lab

(d) Joint

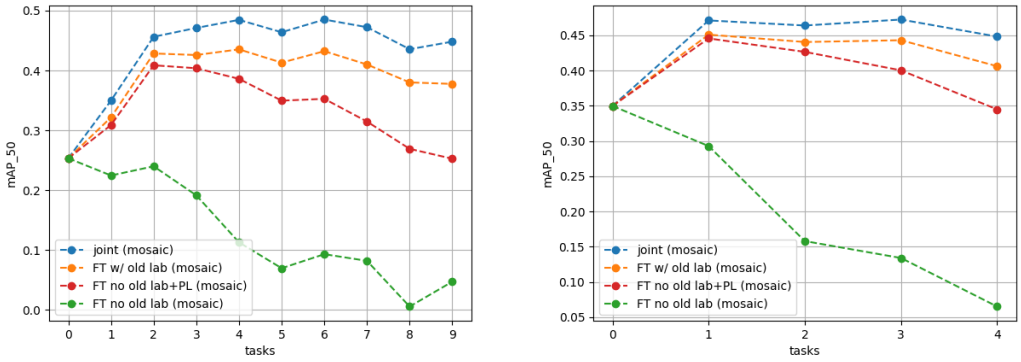
Figure C.10: True and pseudo labels used for 5x2 classes IL on BDD100k.



(a) Finetuning no old lab

(b) Finetuning no old lab + PL

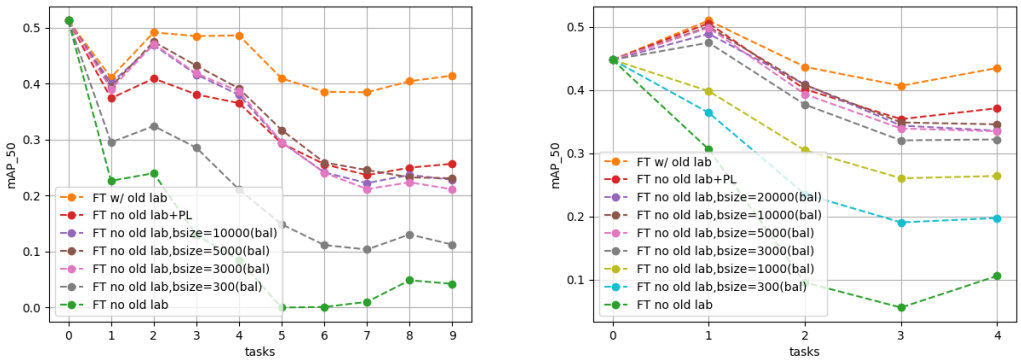
Figure C.11: Per-class mAP reults for 5x2 classes IL on BDD100k



(a) 10x1 classes (b) 5x2 classes

Figure C.12: mAP50 results of yoloV5 on VDP, class incremental training with scenarios joint and variations of finetuning

to have lower base mAP than others. We confirmed those results on the VDP dataset as shown in Figure C.16a and C.16b, with for example a +0.3 mAP increase from finetuning without old class labels, for 5x2 and a 20000 images buffer. Of course, depending on the application, one should try to find the better compromise between buffer size and mAP results to minimize the need for memory.



(a) 10x1 classes (b) 5x2 classes

Figure C.13: mAP50 results of yoloV5 on BDD100K, for incremental training with variations of finetuning. Shown experiments demonstrate, for finetuning no old labels, the replay of a fixed size buffer sampled with strategy “balanced”.

C.4.3.3 Finetuning without old class labels + Pseudo-labeling + Replay

We present here both replay and pseudo-labeling applied to a finetuning scenario without old labels. On BDD100k, the combination of the two methods shows interesting results, Figure C.17a and C.13a, for both 5x2 and 10x1 settings: compared to pseudo-labeling only, it leads in a final mAP increase of respectively +0.02 with buffer-size 3000 and +0.05 with buffer-size

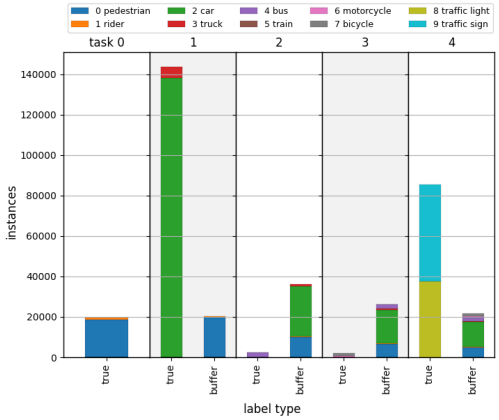


Figure C.14: True- and buffer-labels for scenario finetuning without old labels + replay, strategy balanced, buffer-size=5000, 5x2 classes on BDD100k.

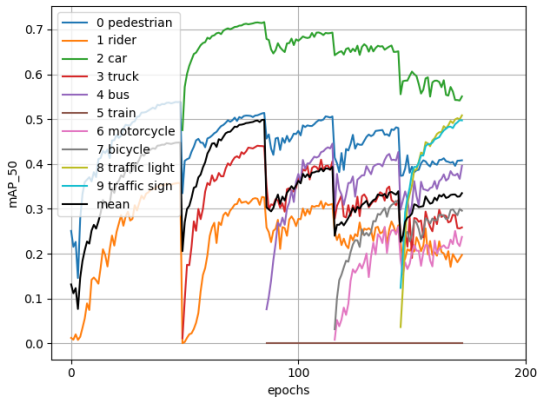
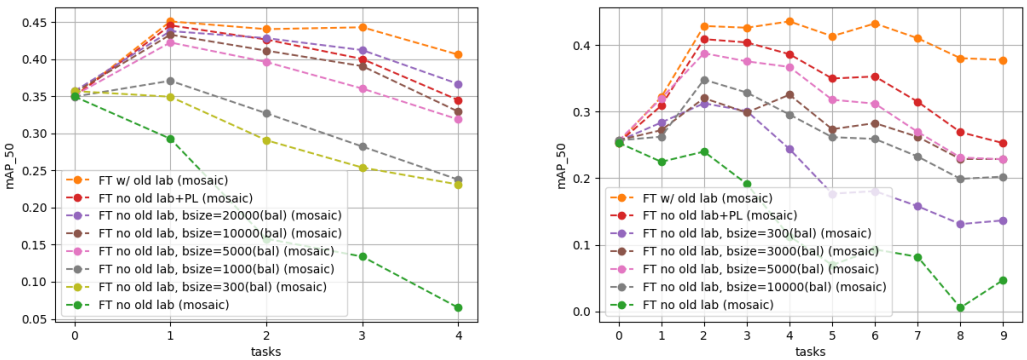


Figure C.15: per-class mAP for scenario finetuning without old labels + replay, strategy balanced, buffer-size=5000, 5x2 classes on BDD100k.

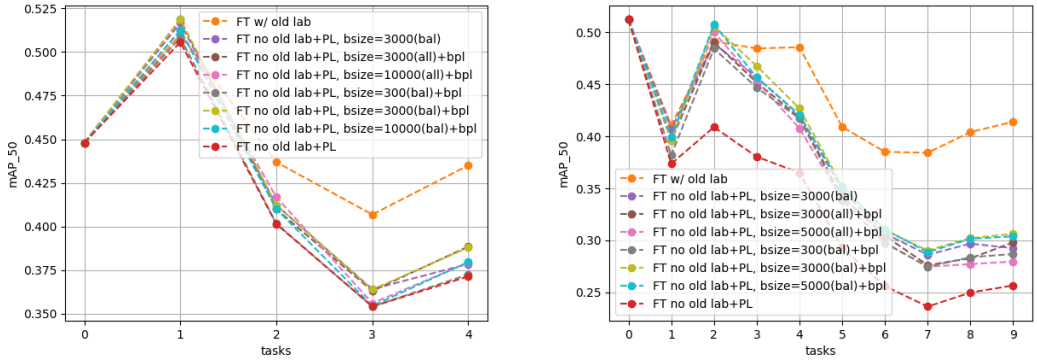


(a) 5x2 classes

(b) 10x1 classes

Figure C.16: mAP50 results of yoloV5 on BDD100K, for incremental training with variations of finetuning. Shown experiments demonstrate, for finetuning no old labels, the replay of a fixed size buffer sampled with strategy “balanced”.

3000. It is worth to note that in 10x1 for example, small buffers of 300 images already help in improving the mAP at each task. The combined methods seems really worth when the scenario is harder like 10x1, as pseudo labeling alone cannot keep mAP up as numerous new tasks are learnt. In addition, the buffer sampling strategy “balanced” seems slightly better than “all”, at equal buffer-size. This means that it is better to have in the buffer a mix of true labels of all the past tasks rather than only from the last task. Both settings give close results though, as we also create pseudo-labels for the images of the buffer.
Figures C.18 and C.19 present per-class mAP and labels used for 10x1 with pseudo-labeling and a buffer of size 5000, our best configuration. On figure C.19, we note that the pseudo-label count represents both the pseudo-labels from task’s dataset images and the pseudo-labels from the buffer images. Those additional labels from the buffer and pseudo-labeling let the model to retain every class mAP well. For example on figure C.18, class *rider*’s mAP climbs up to 0.33 after task three and does not per-class curves do not lower under 0.29 at end of task ten. All other classes show similar resistance to catastrophic forgetting.
On the dataset VDP however (Figure C.20a and C.20b), adding the replay method on top of pseudo labeling brings worse results with a decrease of 0.05 on final mAP. This is a surprising behavior because replaying a buffer should ideally not decrease the mAP: the model sees more true-data. We think this is not due to overfitting because it also happens with buffers of small size and the model never overfit with only that amount of additional data in other scenarios. An explanation could be that the mosaic data augmentation, which is applied only for the VDP dataset, mixes multiples images and labels from different tasks in a one image, and hence images from the buffer could induce false negatives, lowering the performance.



(a) 5x2 classes

(b) 10x1 classes

Figure C.17: mAP50 results of yoloV5 on BDD100K, for class incremental training with variations of finetuning no old labels and pseudo labeling combined with replay of a fixed size buffer sampled with strategy “balanced” or "replace-all".

C.4.3.4 Finetuning with old class labels + Replay

This section presents the results of replay buffer on top of finetuning with old class labels. On BDD100k, the strategy all shows little mAP improvement. We see an improvement at final task of 5x2 with buffer-size 10000, Figure C.21a, where mAP increases by 0.02 and at task 2 for 10x1, where mAP increases by 0.03 with buffer size 3000, Figure C.21b. Figures C.23 and C.22 shows labels and per-class mAP for a 10x1 setting with buffer-size 3000. The main reason for

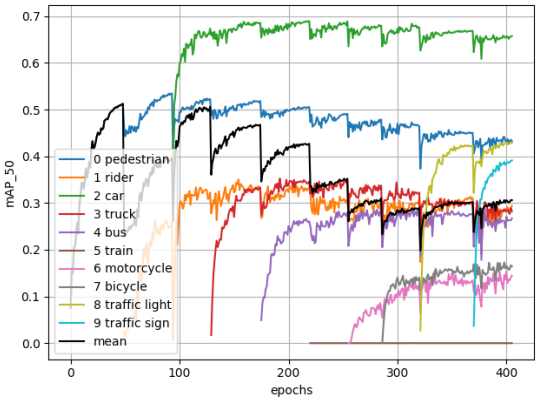


Figure C.18: per-class mAP for scenario finetuning without old labels + pseudo-labeling + replay, strategy balanced, buffer-size=3000, 10x2 classes on BDD100k.

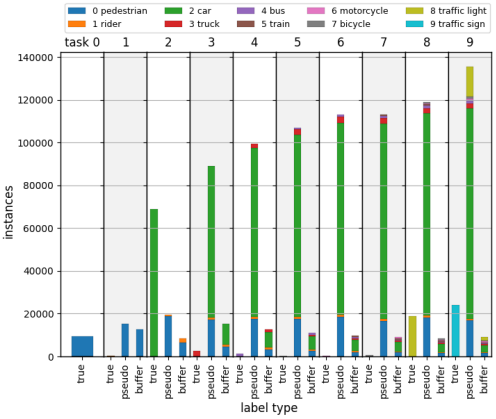
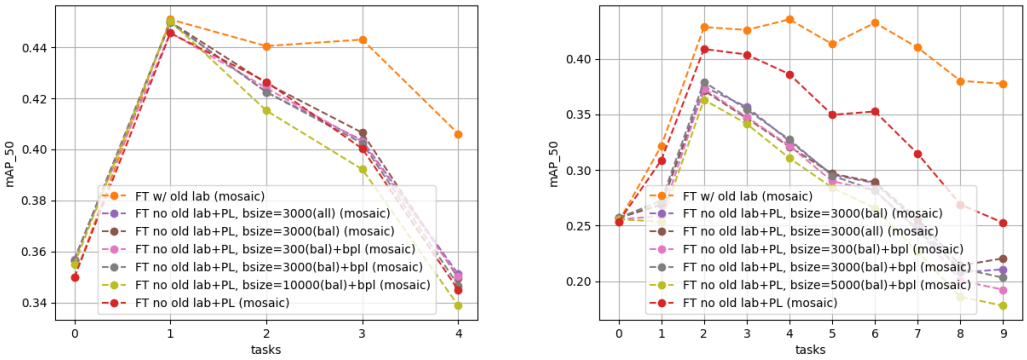


Figure C.19: True-, pseudo- and buffer-labels for scenario finetuning without old labels + pseudo-labeling + replay, strategy balanced, buffer-size=3000, 10x2 classes on BDD100k.

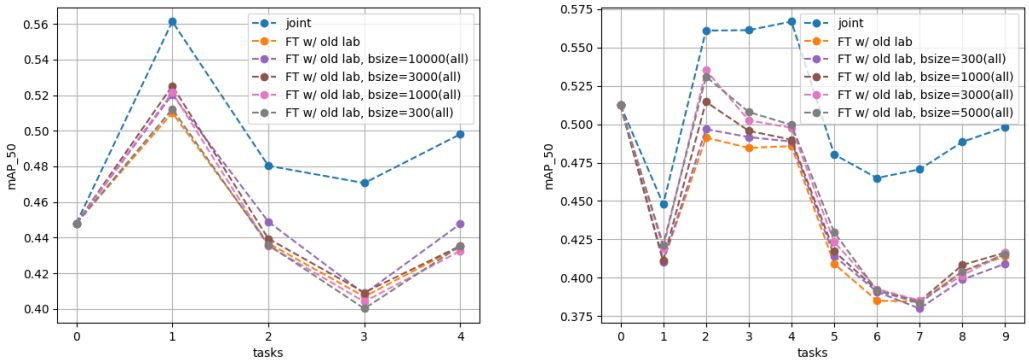


(a) 5x2 classes

(b) 10x1 classes

Figure C.20: mAP50 results of yoloV5 on VDP, for class incremental training with variations of finetuning with no old labels and the replay of a fixed size buffer

the global mAP loss is insufficient and unbalanced data from the dataset leading some classes to have lower base mAP than others. We see also that there are only a few labels from the buffer in comparison of the ground truth labels.
Results on VDP figure C.24a and C.24b are worse as we see that replay brings no improvement or a negative contribution to the base configuration. This is an issue that might be related to the use of mosaic data-augmentation that we applied to VDP only. It is a scenario with already little forgetting, so it is hard to improve, but some mechanisms could improve the replay. First, a better sampling strategy could prioritize the examples of the classes that suffers more from catastrophic forgetting. Second, the experiences could be launched with more epochs for the last training sessions as we see some classes could converge more.



(a) 5x2 classes

(b) 10x1 classes

Figure C.21: mAP50 results of yoloV5 on BDD100K, class incremental training with scenarios joint and variations of finetuning with old labels and pseudo labeling combined with replay of a fixed size buffer sampled with strategy “all

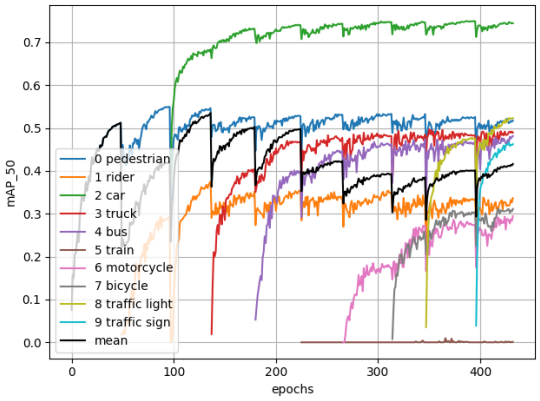


Figure C.22: per-class mAP for scenario finetuning with old labels + replay, strategy all, buffer-size=3000, 10x2 classes on BDD100k

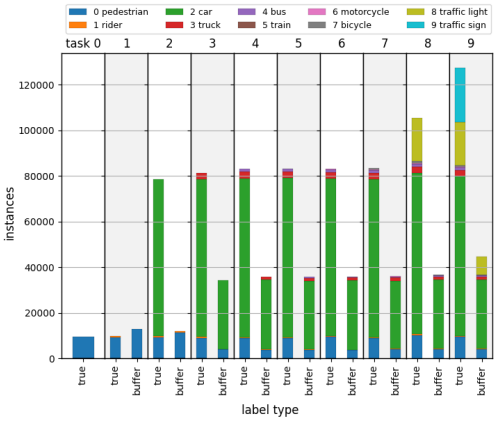
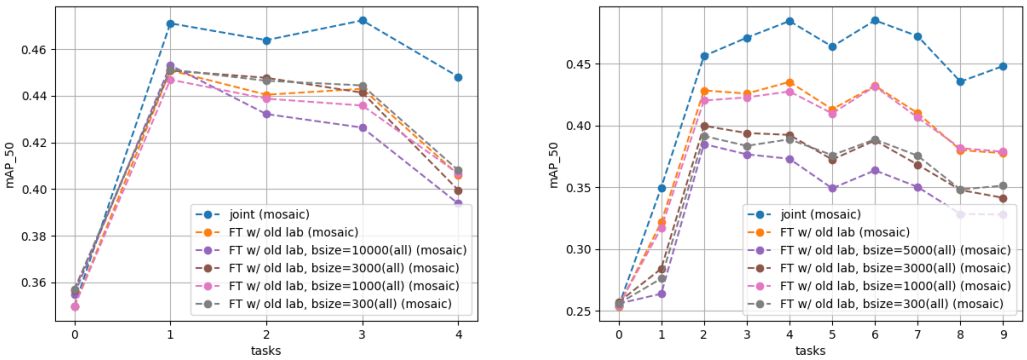


Figure C.23: True- and buffer-labels for scenario finetuning with old labels + replay, strategy all, buffer-size=3000, 10x2 classes on BDD100k.



(a) 5x2 classes

(b) 10x1 classes

Figure C.24: mAP50 results of yoloV5 on VDP, class incremental training with variations of finetuning with old class labels and the replay of a fixed-size buffer sampled with strategy “all

C.5. Future work

A few next steps towards a better understanding of the replay methods and better improvements of the scenario with old class labels are, as follows. First, the impact of mosaic data-augmentation is not clear as in some scenarios it improves the overall mAP but in the same time, it can worsen the contribution of replay methods. Second, we could explore other strategies than all and balanced. For example, we could explore to a buffer-sampling take into account the unbalance-ness of our dataset. To go further, YoloV5 knows in advance the number of maximum classes, and thus computes the loss for every class even if it has no labels of the future classes. This may cause the model take into account some false negative examples and this could be avoided.

C.6. Conclusion

Our study presented methods of pseudo-labelling and replay that alleviates catastrophic forgetting during class incremental training for object detection with the Yolov5 Model on BDD100K and VDP datasets. To implement incremental learning, an initial dataset is split in several tasks, each consisting of several classes. The images corresponding to a task are denoted as a task's train dataset. We validate the model on a unique test dataset, but only with the labels of the task we want to validate. We presented two different Incremental Learning scenarios for Object Detection corresponding to use cases where the new task images do or do not have labels for the old previously seen classes.
The use case without old class labels in the new task images is harder and, if no mechanism is put in place, the model catastrophically forgets the previously learnt classes. However, we succeed in countering catastrophic forgetting while matching the performance of a scenario with old class labels, using pseudo-labeling, replay and a combination of the two. In addition, the replay method makes the model more robust when the Pseudo-Labeling becomes less reliable, for example with a dataset with less occurrence of old class in the new task images. We point out as well that those methods also allow to train with less labels, reducing the labeling cost of new classes objects. The scenario with old class labels suffers a lot less from catastrophic forgetting as the occurrence of past classes's objects in the new task dataset helps the model not to forget them. In this situation, our replay method brings little to no improvement.
We found that evaluating an object detection model in an incremental setting was a challenging problem yet not mature in the state of the art. In order to improve the scenario with old class labels, there are a few next steps towards better YoloV5 tuning and improved buffer sampling strategies.

D. Perspectives and conclusion

Throughout this work, we addressed two general scenarios of iterative dataset construction: Active Learning and Incremental Learning. We analyzed different methods, including our own contributions. To benchmark methods and study their properties, we selected the 2D object detection task in driving environments. The reasons are two-fold. Firstly, this use-case differs in nature and content from the generic picture datasets from academic benchmarks, which in turns help challenge the adaptability of the tested methods. Secondly, the requirements of the automotive industry motivate the use of both subjects: active learning in order to efficiently address massive unlabeled datasets, and incremental learning in order to quickly address new classes while capitalizing on past data. A common observation for the two subject is how important it is to understand the structure and the distribution of the information about objects and how it relates to the training of the model and its performance. Throughout our extensive experimental study, we exhibited some of the most influential properties, observed which ones are general and which ones are specific. Based on these observations, we also proposed various methodological contributions.

For both active learning and incremental learning, we observed that the presence of multiple objects in a single image is a challenge. Future work should aim to propose models and methods that seamlessly handle partial annotations with minimal tuning of hyperparameters.

Bibliography

Acharya, M., Hayes, T. L., and Kanan, C. (2020). RODEO: Replay for Online Object Detection. Technical Report arXiv:2008.06439, arXiv. arXiv:2008.06439 [cs].

Agarwal, S., Arora, H., Anand, S., and Arora, C. (2020). Contextual diversity for active learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 137–153. Springer.

Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., and Agarwal, A. (2020). Deep batch active learning by diverse, uncertain gradient lower bounds. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Belouadah, E., Popescu, A., and Kanellos, I. (2021). A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54.

Beluch, W. H., Genewein, T., Nürnberger, A., and Köhler, J. M. (2018). The power of ensembles for active learning in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Brust, C.-A., Kading, C., and Denzler, J. (2019). Active learning for deep object detection. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers.

Cermelli, F., Geraci, A., Fontanel, D., and Caputo, B. (2022). Modeling missing annotations for incremental learning in object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 3700–3710.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. (2020a). A simple framework for contrastive learning of visual representations.

Chen, X., Fan, H., Girshick, R. B., and He, K. (2020b). Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297.

Chen, X., Yuan, Y., Zeng, G., and Wang, J. (2021). Semi-supervised semantic segmentation with cross pseudo supervision. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2613–2622. Computer Vision Foundation / IEEE.

Chitta, K., Alvarez, J. M., and Lesnikowski, A. (2019). Large-scale visual active learning with deep probabilistic ensembles. *arXiv preprint arXiv:1811.03575*.

Choi, J., Elezi, I., Lee, H.-J., Farabet, C., and Alvarez, J. M. (2021). Active learning for deep object detection via probabilistic modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2020). A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Elezi, I., Yu, Z., Anandkumar, A., Leal-Taixe, L., and Alvarez, J. M. (2021). Not all labels are equal: Rationalizing the labeling costs for training object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Elezi, I., Yu, Z., Anandkumar, A., Leal-Taixé, L., and Alvarez, J. M. (2022). Not All Labels Are Equal: Rationalizing The Labeling Costs for Training Object Detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 14472–14481. IEEE.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338.

Gal, Y., Islam, R., and Ghahramani, Z. (2017). Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR.

Gao, M., Zhang, Z., Yu, G., Arık, S., Davis, L., and Pfister, T. (2020). Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Geifman, Y. and El-Yaniv, R. (2017). Deep active learning over the long tail. *ArXiv*, abs/1711.00941.

Gidaris, S., Bursuc, A., Puy, G., Komodakis, N., Cord, M., and Pérez, P. (2021). Obow: Online bag-of-visual-words generation for self-supervised learning.

Gidaris, S., Singh, P., and Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations.

Hao, Y., Fu, Y., Jiang, Y.-G., and Tian, Q. (2019). An end-to-end architecture for class-incremental object detection with knowledge distillation. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Haussmann, E., Fenzi, M., Chitta, K., Ivanecky, J., Xu, H., Roy, D., Mittel, A., Koumchatzky, N., Farabet, C., and Alvarez, J. M. (2020). Scalable active learning for object detection. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. B. (2022). Masked autoencoders are scalable vision learners.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. B. (2020). Momentum contrast for unsupervised visual representation learning.

Hu, R., Dollár, P., He, K., Darrell, T., and Girshick, R. (2018). Learning to segment every thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, S., Wang, T., Xiong, H., Huan, J., and Dou, D. (2021). Semi-supervised active learning with temporal output discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Jeong, J., Lee, S., Kim, J., and Kwak, N. (2019). Consistency-based semi-supervised learning for object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kang, D. and Cho, M. (2022). Integrative few-shot learning for classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9979–9990.

Kao, C.-C., Lee, T.-Y., Sen, P., and Liu, M.-Y. (2018). Localization-aware active learning for object detection. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

Kim, S., Bae, S., and Yun, S.-Y. (2023). Coreset sampling from open-set for fine-grained self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7537–7547.

Lang, A., Mayer, C., and Timofte, R. (2022). Best practices in pool-based active learning for image classification.

Larsson, G., Maire, M., and Shakhnarovich, G. (2016). Learning representations for automatic colorization.

Lee, D.-H. and others (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Issue: 2.

Li, Y., Huang, D., Qin, D., Wang, L., and Gong, B. (2020). Improving object detection with selective self-supervised self-training. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Liang, K. J., Rangrej, S. B., Petrovic, V., and Hassner, T. (2022). Few-shot learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9089–9098.

Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In Fleet, D. J., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer vision - ECCV 2014 - 13th european conference, zurich, switzerland, september 6-12, 2014, proceedings, part V*, volume 8693 of *Lecture notes in computer science*, pages 740–755. Springer. tex.bibsource: dblp computer science bibliography, https://dblp.org tex.timestamp: Sat, 30 Sep 2023 09:39:22 +0200.

Liu, Y.-C., Ma, C.-Y., He, Z., Kuo, C.-W., Chen, K., Zhang, P., Wu, B., Kira, Z., and Vajda, P. (2021a). Unbiased Teacher for Semi-Supervised Object Detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Liu, Z., Ding, H., Zhong, H., Li, W., Dai, J., and He, C. (2021b). Influence selection for active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9274–9283.

Lyu, M., Zhou, J., Chen, H., Huang, Y., Yu, D., Li, Y., Guo, Y., Guo, Y., Xiang, L., and Ding, G. (2023). Box-level active detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23766–23775.

Maltoni, D. and Lomonaco, V. (2019). Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73.

Maracani, A., Michieli, U., Toldo, M., and Zanuttigh, P. (2021). Recall: Replay-based continual learning in semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7006–7015.

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and van de Weijer, J. (2020). Class-incremental learning: survey and performance evaluation. *CoRR*, abs/2010.15277.

Montejano Villabla, S., Hajri, H., HERBIN, S., Spyros, G., Ospici, M., Rami, H., Reyboz, M., Sanchez, E. H., and Tamaazousti, M. (2021). Data-driven ai design methods - state of the art. Technical report, Confiance.AI, EC4: Trustworthiness by design.

Nabhan, M., Pablo, M., Miorelli, R., and Dupont, C. (2021). Dataset improvement and construction iteration. Technical report, Confiance.AI, EC5: Data and knowledge engineering for trusted AI.

Noroozi, M. and Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles.

Pardo, A., Xu, M., Thabet, A. K., Arbeláez, P., and Ghanem, B. (2021). Baod: Budget-aware object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1247–1256.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Pourahmadi, K., Nooralinejad, P., and Pirsiavash, H. (2021). A simple baseline for low-budget active learning. *ArXiv*, abs/2110.12033.

Radosavovic, I., Dollár, P., Girshick, R. B., Gkioxari, G., and He, K. (2018). Data distillation: Towards omni-supervised learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4119–4128.

Roy, S., Unmesh, A., and Namboodiri, V. P. (2018). Deep active learning for object detection. *Proceedings of the British Machine Vision Conference (BMVC)*.

Sainburg, T., McInnes, L., and Gentner, T. Q. (2021). Parametric umap embeddings for representation and semisupervised learning. *Neural Computation*, 33(11):2881–2907.

Samet, N., Siméoni, O., Puy, G., Ponimatkin, G., Marlet, R., and Lepetit, V. (2023). You never get a second chance to make a good first impression: Seeding active learning for 3d semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*.

Sener, O. and Savarese, S. (2018). Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Settles, B. (2009). Active Learning Literature Survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences.

Shin, G., Albanie, S., and Xie, W. (2022). Unsupervised salient object detection with spectral cluster voting. In *CVPRW*.

Siméoni, O., Puy, G., Vo, H. V., Roburin, S., Gidaris, S., Bursuc, A., Pérez, P., Marlet, R., and Ponce, J. (2021). Localizing objects with self-supervised transformers and no labels.

Siméoni, O., Sekkat, C., Puy, G., Vobecky, A., Zablocki, É., and Pérez, P. (2023). Unsupervised object localization: Observing the background to discover objects. In *CVPR*.

Sohn, K., Zhang, Z., Li, C.-L., Zhang, H., Lee, C.-Y., and Pfister, T. (2020). A simple semi-supervised learning framework for object detection. In *arXiv:2005.04757*.

Tang, P., Ramaiah, C., Xu, R., and Xiong, C. (2021). Proposal learning for semi-supervised object detection. *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2290–2300.

Tasar, O., Tarabalka, Y., and Alliez, P. (2019). Incremental learning for semantic segmentation of large-scale remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9):3524–3537.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. (2020). What makes for good views for contrastive learning?

Tian, Z., Shen, C., Chen, H., and He, T. (2022). FCOS: A simple and strong anchor-free object detector. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(4):1922–1933.

Uijlings, J., van de Sande, K., Gevers, T., and Smeulders, A. (2013). Selective search for object recognition. *International Journal of Computer Vision*.

Vo, H. V., Siméoni, O., Gidaris, S., Bursuc, A., Pérez, P., and Ponce, J. (2022). Active learning strategies for weakly-supervised object detection.

Wang, K., Yan, X., Zhang, D., Zhang, L., and Lin, L. (2018). Towards human-machine cooperation: Self-supervised sample mining for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, X., Girdhar, R., Yu, S. X., and Misra, I. (2023). Cut and learn for unsupervised object detection and instance segmentation. *arXiv preprint arXiv:2301.11320*.

Wang, Y., Shen, X., Hu, S. X., Yuan, Y., Crowley, J. L., and Vaufreydaz, D. (2022). Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*.

Wu, T., Huang, J., Gao, G., Wei, X., Wei, X., Luo, X., and Liu, C. H. (2021). Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 16765–16774. Computer Vision Foundation / IEEE.

Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., and Liu, Z. (2021). End-to-end semi-supervised object detection with soft teacher. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Yoo, D. and Kweon, I. S. (2019). Learning loss for active learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning.

Yuan, T., Wan, F., Fu, M., Liu, J., Xu, S., Ji, X., and Ye, Q. (2021). Multiple instance active learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhdanov, F. (2019). Diverse mini-batch active learning. *ArXiv*, abs/1901.05954.

Zhou, D.-W., Wang, Q.-W., Qi, Z.-H., Ye, H.-J., Zhan, D.-C., and Liu, Z. (2023). Deep class-incremental learning: A survey.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A. L., and Kong, T. (2022). Image BERT pre-training with online tokenizer.

Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D., and Le, Q. V. (2020). Rethinking pre-training and self-training. *Advances in Neural Information Processing Systems (NeurIPS)*.

Confiance ai

Title: Title in english

Keywords: object detection, acquisition, diversity, consistency, catastrophic forgetting, memory buffer, pseudo-labeling, replay, incremental, continual

As industry's interest toward autonomous vehicles grows, efforts must be made to address the challenging computer vision problem of Object Detection. In this direction, today's AI models get more and more efficient at finding objects in images. In fact, this success has been made by the widespread use of deep neural networks that need a huge amount of labeled data to be trained on. Active learning attempts to reduce the quantity of data to label in order to reduce annotation costs which are particularly high for the object detection task. We proposed in this batch to develop new acquisition functions adapted to the object detection task by leveraging either diversity at the box level or consistency between several augmented views of an image. These approaches bring interesting results but lack generalization across the different datasets studied. Moreover, deep neural detectors are usually trained to identify a restricted amount of classes, but they lack the ability to be easily trained with new data, to identify new classes, and in the same time not forget the old ones (no catastrophic forgetting). Incremental learning is a paradigm in which a machine learning agent evolves continuously, learns new tasks and accumulates the knowledge from previous tasks. A known method of incremental learning originating from classification is the utilisation of a memory buffer storing old data examples. In this report we also study incremental learning with a memory buffer in the context of object detection. The methods cover the cases when new images do or do not contain the labels of old classes. Furthermore, we investigate the application of pseudo-labeling in addition to a memory buffer.


Air Liquide
AIRBUS
Atos
cea
Inria
NAVAL GROUP
GROUPE RENAULT
SAINT EXUPÉRY
SAFRAN
sopra steria
SystemX
THALES
Building a future we can all trust
Valeo
INVESTIR L'AVENIR