Conditioned Cooperative Training for Semi-supervised Weapon Detection

Jose L. Salazar González[a,*], Juan A. Álvarez-García[a], Fernando J. Rendón-Segador[a], Fabio Carrara[b]

[a]*Dpto. de Lenguajes y Sistemas Informáticos, Universidad de Sevilla, Spain*
[b]*Institute of Information Science and Technologies of the National Research Council of Italy (ISTI-CNR), Pisa, Italy*

Abstract

Violent assaults and homicides occur daily, and the number of victims of mass shootings increases every year. However, this number can be reduced with the help of Closed Circuit Television (CCTV) and weapon detection models, as generic object detectors have become increasingly accurate with more data for training. We present a new semi-supervised learning methodology based on conditioned cooperative student-teacher training with optimal pseudo-label generation using a novel confidence threshold search method and improving both models by conditional knowledge transfer. Furthermore, a novel firearms image dataset of 458,599 images was collected using Instagram hashtags to evaluate our approach and compare the improvements obtained using a specific unsupervised dataset instead of a general one such as ImageNet. We compared our methodology with supervised, semi-supervised and self-supervised learning techniques, outperforming approaches such as YOLOv5m (up to +19.86), YOLOv5l (up to +6.52) Unbiased Teacher (up to +10.5 AP), DETReg (up to +2.8 AP) and UP-DETR (up to +1.22 AP).

Keywords: Semi-supervised Learning, Self-supervised Learning, Supervised Learning, Weapon Detection, Knowledge Transfer.

1. Introduction

Gun violence is a major global problem, with numerous assaults and homicides recorded daily. According to Gun Violence Archive[1], the number of mass shootings in the US is on the rise (417 in 2019, 610 in 2020 and 692 in 2021). These events are often recorded by Closed Circuit Television (CCTV), which is almost ubiquitous in all public or private buildings[2]. Still, the recordings are usually only used for criminal investigations.

Using automatic object detection in video surveillance with security cameras is a novel and decisive factor in anticipating and preventing such events. Automatic object detection in images and video streams has achieved high performance for generic objects due to massive annotated image databases such as ImageNet[3] or COCO[4], among others, on which models can be trained in a supervised way.

However, when applying these detectors to more specific problems, such as weapons detection, fully supervised approaches are less viable, as the datasets for training or fine-tuning of detectors are often limited in the amount and variability of labelled data, which limits the generalisability of the detectors.

Manually collecting and annotating more data often increases performance, but it comes at a very high cost for large image sets. Accurate annotation of an object within an image can take 35 seconds, and an image can contain more than one object.

Although little annotated data can be found for specific tasks, there are several sources on the Internet of unannotated data, such as Instagram[5], where a large number of images are daily published that are often not taken into account due to the absence of labels. For these reasons, unsupervised learning techniques that can take advantage of large unlabelled datasets are currently of great interest but often require datasets at the scale of millions or billions [1] while being challenging to handle to solve the desired task.

A hybrid alternative, taking the best of both, is the semi-supervised learning approach, which combines a small amount of labelled data with a large amount of unlabelled data during training, usually based on pseudo-label generation, which uses a trained model to generate labels for the unlabelled data, keeping only the samples that the model relies on for labelling.

This study presents a novel semi-supervised methodology for detecting weapons, specifically firearms, in images based on *conditional cooperative training procedure*. Our proposed approach operates in cycles, in which two object detectors (the deformable DETR detection models [2]) follow a conditioned cooperative student-teacher training process, whereby the teacher model is trained on the small amount of manually labelled data. The student model is trained on the pseudo-labels generated by the teacher model on the unlabelled data. If the cycle has proved beneficial to the teacher model, a percentage of the student's

*Corresponding author.
Email addresses: jsalazar@us.es (Jose L. Salazar González), jaalvarez@us.es (Juan A. Álvarez-García), frendon@us.es (Fernando J. Rendón-Segador), fabio.carrara@isti.cnr.it (Fabio Carrara)

[1]https://www.gunviolencearchive.org/past-tolls

[2]https://www.comparitech.com/blog/vpn-privacy/us-surveillance-camera-statistics/

[3]ImageNet: https://www.image-net.org/

[4]COCO: https://cocodataset.org/

[5]Instagram: https://www.instagram.com/

new knowledge is transferred to the teacher model.

We assess the importance of choosing the confidence threshold to assign pseudo-labels and propose a novel threshold-finder method applied at each cycle. During the cooperative training, we conditionally improve the teacher model via knowledge transfer from the student model if this enhances the teacher's performance on a labelled control set, allowing it to perform a new secure cycle of pseudo-label generation and student training.

We perform experiments using existing labelled datasets for firearm detection and a novel non-labelled set of 458,599 images we collected for this study from Instagram by selecting posts with the "handgun" hashtag.

We introduce and evaluate two novel elements in the semi-supervised learning pipeline based on improving the pseudo-labelling process. The first element consists of adding a conditioning module to the iterative learning process between teacher and student models to prevent confirmation bias during the training cycles. As naive pseudo-labelling overfits in the presence of incorrect pseudo-labels [3], known as confirmation bias. The second element is a method called threshold-finder, which selects the best confidence threshold for each training cycle, allowing the models to learn from pseudo-labels with the best confidence threshold. Both are analysed using an ablation study and tested on two datasets, making our approach more robust than the state of the art.

In summary, the main contributions of this study are the following:

1. We propose a new conditioned cooperative semi-supervised learning methodology that outperforms state-of-the-art techniques for two datasets based on firearm detection[6].
2. To train the semi-supervised learning methodology and compare its effectiveness, we have created an extensive, unsupervised dataset pooled from Instagram using a hashtag-based search.
3. We have implemented state-of-the-art semi-supervised and self-supervised learning techniques to compare results further, demonstrating a higher precision rate than traditional supervised approaches.
4. Due to the importance of optimal threshold selection in semi-supervised learning systems, we have introduced a novel methodology for identifying the optimal threshold during the pseudo-labelling process that enhances the overall efficacy of such systems.

The results obtained with our methodology in a firearm detection problem show a mean improvement of up to 10.5 AP points compared to Unbiased Teacher [4], a state-of-the-art technique in semi-supervised learning. Also, we obtained a gain of up to 2.8 and 1.22 AP points against DETReg [5] and UP-DETR [6], respectively, both state-of-the-art self-supervised techniques, with a reduction of training time by a factor of six. Furthermore, applying our new dataset with UP-DETR, we improved up to 6 AP points compared to the ImageNet dataset. Compared to these alternatives, the proposed method reduces the complexity in the training phase, working with subsets instead of with the complete set of unlabeled images. This will be seen in more detail in the section 4.3.

The rest of the paper is organised as follows. In Section 2, related works on firearm detection for security and video surveillance and semi-supervised and self-supervised learning studies are introduced. Section 3 describes our methodology. Section 4 describes the experimental setup, including datasets, detection network and implementation details. Section 5 details the experimentation performed. Section 6 discusses the experimentation results. Lastly, conclusions are covered in Section 7.

2. Related work

2.1. Firearm detection for security and video surveillance

Most techniques used to detect objects in video surveillance footage adopt sliding window feature extractors, one of the most widely used techniques [7]. However, the use of sliding windows and Haar feature-based cascade classifiers is not as robust as more recent alternatives, such as Region Proposal Networks (RPN) with Convolutional Neural Networks (CNN) or Transformers. They require a handcrafted pipeline and preprocessing, which makes their generalisation more complex. Furthermore, more processing time is needed than other alternatives, which does not favour an early threat detection system, where time is critical for threat prevention.

Studies such as Olmos et al. [8], or Bhatti et al. [9] analysed the performance of detection systems using CNN-based classifiers and compared the use of sliding windows with RPN, being the combination of CNN with RPN the ones that obtained the best results. In addition, other studies [10, 11, 12] analysed the importance of image size in video surveillance environments with CNN-based methods.

Several works aim to reduce false positives, for example, using deep autoencoders [13], human poses in combination with object detection [14, 15], or binary classifiers to identify small objects handled that can be confused with other similar objects [16]. However, other studies take advantage of the temporal component of the video to detect violent behaviour, as is the case of the study by Rendón-Segador et al. [17], or use the temporal component of the video to reduce false positives as well, as seen in the study by Olmos et al. [18].

Although the current state-of-the-art in weapon detection achieves high predictive performance on different datasets, these datasets are generally based on violent movies in which weapons appear in close-up images. As part of a national security project [19], we analyse performance in real environments in our previous study [20] by collecting a new dataset obtained from our university's CCTV cameras during a mock attack. This CCTV set presented new difficulties we did not find in other datasets, such as distance to armed people of more than 7 metres, weapon poses, occlusions, and light conditions, among others. When the predictive performance of the best

[6]Code: https://github.com/Deepknowledge-US/conditioned-cooperative-training

weapons detection models was analysed in this CCTV dataset, the metrics showed worse results even using synthetic data augmentation (68.3 AP in the Olmos et al. dataset [8] compared to 14.6 AP in our dataset). These previous results made us realise the need for much more annotated data and the difficulty of finding images with realistic conditions; proposing this study, with the augmentation of training data using a set of unsupervised images unrelated to the original dataset, with the premise of bringing an improvement to the results.

2.2. Semi-supervised and self-supervised learning

Semi-supervised and self-supervised learning has been shown to achieve excellent results in state-of-the-art computer vision [21]. Especially in areas where the amount of available labelled data is limited, such as medicine [22], remote sensing [23, 24], vehicle detection [25, 26], or biology [27], among others.

Much of the state-of-the-art in semi-supervised and self-supervised learning applied to computer vision aims at improving training with little annotated data for label classification given an input image [28]. In this field, we can find several proposals that use different approaches to achieve a performance close to fully supervised learning. Such as the approach of He et al. with MoCo [29, 30] or Chen et al. with SimCLR [31], which used self-supervised techniques based on contrastive learning. Other popular models use different self-supervised learning techniques based on feature prediction using two models, such as Bootstrap Your Own Latent (BYOL) [32] or Swav [33]. Moreover, studies such as SimCLRv2 [34] demonstrate that they can outperform even supervised models using semi-supervised learning techniques such as knowledge distillation on unlabelled data or using the cooperation of two models in Meta Pseudo Labels [35].

The success of semi-supervised and self-supervised learning in image classification tasks motivated the study of these learning techniques in object detection tasks. Sohn et al. proposed STAC [36], a framework that improves predictive performance using unsupervised data by employing pseudo-labels with high confidence in the unsupervised samples and updating the model by forcing consistency through strong augmentations. The authors of this study also performed an exploration of relevant parameters in the framework, such as the confidence threshold, observing that in their case, a threshold of 0.9 performed the best, followed by 0.7; nevertheless, they found that lower thresholds (i.e., 0.5) improved the recall of the pseudo-labels. To train this system, they follow four steps: 1) train a teacher model; 2) generate pseudo-labels with the teacher model; 3) apply strong data augmentations to the unsupervised samples; and 4) compute the unsupervised and supervised loss to train a detector. Nonetheless, unlike our methodology, they do not have any control to prevent degradation of model accuracy, nor do they use dynamic thresholds to improve training.

Later, Dai et al. published Unsupervised Pre-train DETR (UP-DETR) for object detection [6], a proposal inspired by the success obtained in transformer pre-training applied to natural language processing. To transfer this premise to the object detection domain, they propose a pretext task based on getting the location of patches in images, called random query patch detection. Pre-training the detection model with unsupervised data and doing a subsequent supervised training phase, where UP-DETR performs better than DETR with faster convergence and better average precision.

Unbiased Teacher for Semi-Supervised Object Detection, published by Liu et al. [4], applies to the detection process already used techniques in semi-supervised classification learning. They propose gradual training with two models, a teacher and a student. The former learns in a supervised way from the labelled samples to subsequently generate pseudo-labels with unlabelled data. In contrast, the latter, which jointly trains from both pseudo-labelled and supervised-labelled data, helps to improve the teacher model by updating their weights using an Exponential Moving Average (EMA).

Student-teacher based learning techniques are not something new, they are already present in the state of the art with great results. This type of learning is based on making use of two models, acting as teacher and student. The teacher model is trained in a standard manner and the purpose of this approach is to transfer the knowledge to the student model. Examples of use include domain adaptation [37] or knowledge distillation [38]. It is however important to note that this type of learning can lead to a deterioration in accuracy during the learning progress. In order to address this issue, Meng et al. [39] propose to apply a conditioner to their domain adaptation problem to select pseudo-labelled or ground truth data based on the correctness of the teacher model. Even though they proposed that this conditioner could reduce errors by up to 12.8% and demonstrate the importance of conditioners in student-teacher approaches, this conditioner is not applicable to our problem as it requires all data to be labelled.

With the premise of pre-training an Transformers object detection Transformer architecture with unlabelled data using pretrain tasks and region priors, Bar et al. [40] published the study Detection with Transformers using Region priors (DETReg). In this case, they use two pre-training tasks: "Object Localisation Task", to train the model to locate objects in the image regardless of the category, and the "Object Embedding Task", to train the model to distinguish the categories of objects in the image.

Our proposal is based on the findings of these previous studies. From the STAC study, we exploited that the pseudo-label generation by a teacher model to train a student model can improve predictive performance. From the Unbiased Teacher for Semi-Supervised Object Detection study, we exploit the cooperation of two models with pseudo-labels in an iterative approach that can further improve this predictive performance. Furthermore, we adopt knowledge transfer through EMA, as Liu et al. [4] demonstrated its benefit in semi-supervised learning systems achieving higher prediction performance. From the STAC study, we also observed the importance of exploring confidence thresholds and that low thresholds can improve the recall metric. In addition, the use of transformers is gaining a reputation in the state-of-the-art with high results, as shown by the self-supervised learning with DETR and Deformable DETR carried out by the UP-DETR and DETReg studies, respectively.

3. Methodology

In the following, everything related to our methodology will be explained, including the definition of the problem, a first overview and a definition of the parameters that constitute the methodology, followed by an explanation of the phases that conform to the methodology.

Problem definition. The aim of this methodology is to perform enhanced semi-supervised learning that makes use of supervised datasets represented as $\boldsymbol{D_{sup}} = \{(x_i^s, y_i^s)\}_{i=1}^{N_{sup}}$, an unsupervised dataset represented as $\boldsymbol{D_{uns}} = \{x_i^u\}_{i=1}^{N_{uns}}$ with its pseudo-labelled dataset represented as $\boldsymbol{D'_{uns}} = \{x_i^u, y_i^u\}_{i=1}^{N_{uns}}$ and a control set represented as $\boldsymbol{D_{ctrl}} = \{(x_i^c, y_i^c)\}_{i=1}^{N_{ctrl}}$, where x_i is an image and y_i is the annotation of the image x_i. The annotation y_i contains the bounding box (**b**) and the label (**c**) of each object in the image x_i.

Overview. This methodology, shown in Figure 1 and Pseudo-code 1, is based on the conditioned cooperative learning of two models, namely: teacher and student models. The role of the teacher model is to learn in a supervised way on D_{sup}, and the role of the student model is to improve the teacher model through semi-supervised learning on D_{uns}. For this purpose, two phases are carried out: **warm-up phase** (Section 3.1) and **iterative cooperative phase** (Section 3.2).

Parameters. Relevant parameters define its proper performance, namely: **confidence threshold** (δ), which is automatically established by our threshold-finder, indicates the minimum confidence value that detection must meet to be considered and avoid being discarded, being present in the pseudo-label generation process that the teacher model goes through; **epochs per cycle** (**EPC**), which defines the number of epochs that our student model will be trained for each cycle; **attempts**, which indicates the number of improvement attempts during the iterative cooperation phase; and **number of unsupervised images** ($\boldsymbol{N_{uns}}$), which determines the number of unsupervised images to use in our learning during the iterative cooperative phase. The value of $\boldsymbol{N_{uns}}$ must be much less than that of N to allow several iteration cycles of the algorithm.

3.1. Warm-up phase

During the warm-up phase, the teacher model is trained using the supervised data D_{sup}, which subsequently allows the generation of annotations of the object(s) to be detected with a minimum accuracy that allows the algorithm to start the automatic pseudo-labelling process with enough AP, allowing the adequate training of the student during the iterative cooperative phase. These generated annotations will allow the knowledge of the teacher model to be used in a larger unlabelled dataset. Therefore, proper training in this phase is necessary for the successful performance of the next stage.

Once the warm-up phase is over, the weights of the teacher model are transferred to the student model ($\theta_s \leftarrow \theta_t$), thus generating a copy of the teacher model to train the student model with the accuracy obtained in the warm-up.

The loss of the teacher model is represented in Equation 1, being x_i^s and y_i^s the image and the annotation of the supervised

Algorithm 1 Pseudo-code of Conditioned Cooperative Training methodology

Require: EPC: Number of epochs per cycle.
Require: N_{uns}: Number of unsupervised samples.
Require: attempts: Number of attempts of improvement.
teacher.train(D_{sup}) ▷ Warm-up phase.
teacher_AP = teacher.evaluate(D_{ctrl})
teacher_weights = teacher.get_weights().copy
non_improvements = 0
while non_improvements < attempts **do** ▷ Iterative cooperation phase.
    δ = threshold_finder(teacher, D_{ctrl})
    D'_{uns} = teacher.generate_pseudo_labels(D_{uns}, δ, N_{uns})
    student.set_weights(teacher_weights)
    student.train(D'_{uns}, epochs=EPC, EMA_to=teacher)
    new_teacher_AP = teacher.evaluate(D_{ctrl})
    if new_teacher_AP > teacher_AP **then**
        teacher_AP = new_teacher_AP
        teacher_weights = teacher.get_weights().copy()
        non_improvements = 0
    else
        teacher.set_weights(teacher_weights)
        non_improvements = non_improvements + 1
    end if
end while

Algorithm 2 Pseudo-code of threshold-finder methodology

Require: N_{uns}: Number of unsupervised samples.
APs_by_threshold = {}
teacher_weights = teacher.get_weights().copy()
for δ in range(0.5, 1, 0.1) **do**
    D'_{uns} = teacher.generate_pseudo_labels(D_{uns}, δ, N=300)
    student.train(D'_{uns}, epochs=5, EMA_to=None)
    APs_by_threshold[δ] = teacher.evaluate(D_{ctrl})
    student.set_weights(teacher_weights)
end for
APs_by_threshold.sort_by_AP()
thresholds = APs_by_threshold.keys()
for δ in thresholds **do**
    D'_{uns} = teacher.generate_pseudo_labels(D_{uns}, δ, N_{uns})
    if len(D'_{uns}) is N_{uns} **then**
        return δ
    end if
end for

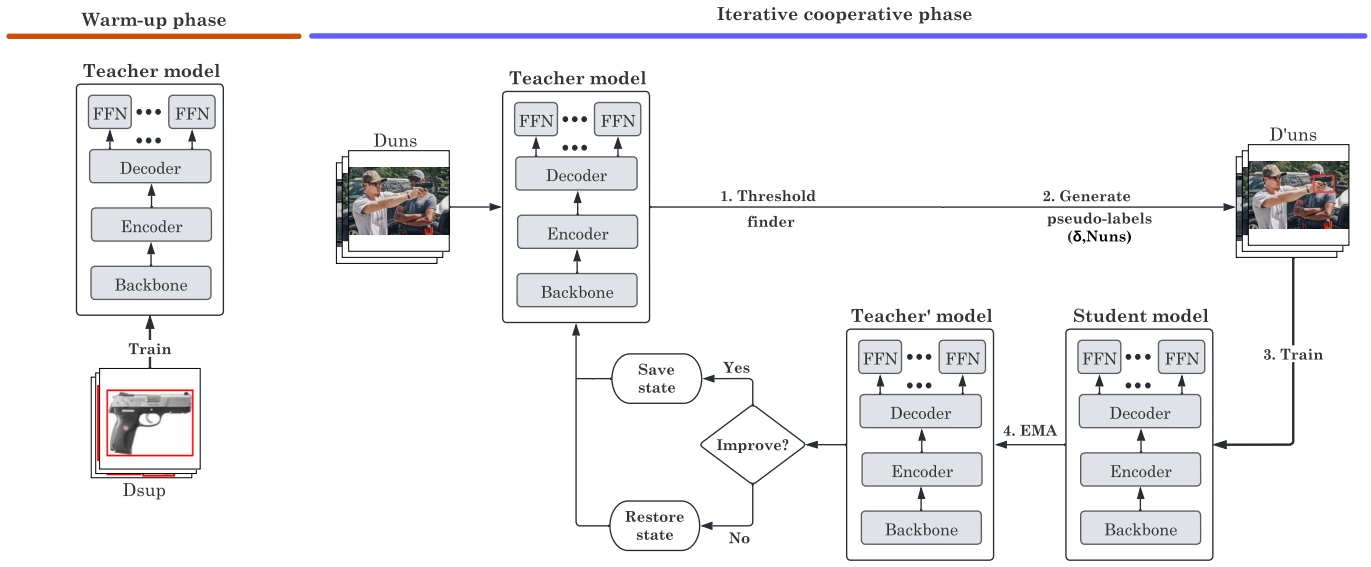


Figure 1: Diagram of Conditioned Cooperative Training methodology. Teacher and student models use Deformable DETR architecture. Teacher' refers to an exact copy of teacher, which is modified by a training cycle, obtaining an updated AP (AP').

dataset D_{sup}. This loss is calculated during this phase as the sum of the classification loss with Focal loss [41], the bounding box loss with L1 loss [42], and the Intersection over Union (IoU) loss with Generalised IoU loss [43].

$$\mathcal{L}_{\text{teacher}} = \sum_{i=1}^{N_{\text{sup}}} \mathcal{L}_{\text{cls}}(x_i^s, y_i^s) + \mathcal{L}_{\text{bbox}}(x_i^s, y_i^s) + \mathcal{L}_{\text{iou}}(x_i^s, y_i^s) \quad (1)$$

3.2. Iterative cooperative phase

The iterative cooperative phase starts from the weights obtained during the warm-up phase. It uses the teacher model to generate pseudo-labels of a random subset of D_{uns} with a size of N_{uns}, represented as D'_{uns}, which allows training of the student model with more significant variability of data. While the student model learns from these data, a knowledge transfer is made to the teacher model through EMA. Being its weights slightly modifies to learn from this new source, only if the new average precision ($\text{new\_teacher}_{\text{AP}}$) improves the previous one ($\text{teacher}_{\text{AP}}$) evaluated over D_{ctrl}. These steps presented in each cycle are:

1. **Threshold-finder:** A search for the optimal confidence threshold is performed by running a search algorithm. This algorithm trains the student model with a small random sample of 300 pseudo-labels obtained from the teacher model that satisfy a specific confidence threshold δ. Being δ initiated with a value of 0.5 and increased at regular intervals of 0.1 until it reaches 1, where the search algorithm will stop and choose δ with the best Average Precision (AP) achieved.
2. **Generate pseudo-labels:** The teacher model is used to generate pseudo-labels $D'_{uns} = \{(b_j, c_j)\}_{j=1}^{N_{uns}}$ from a random subset of the entire dataset D_{uns} that contains labels with confidence values greater than a δ selected in the previous step and a size of N_{uns}, where b_j is the bounding box and c_j is the label associated with the bounding box b_j.
3. **Train:** The weights are copied from the teacher model to the student model ($\theta_s \leftarrow \theta_t$, where θ_s are the weights of the student model and θ_t are the weights of the teacher model) and the student model is trained. The loss used in this phase for the student model is represented in Equation 2, which follows the same loss functions as the definition of teacher loss, except that in this case, the D'_{uns} is used instead of the supervised one, being x_i^u the i image of D_{uns} and y_i^u the pseudo-label generated with the prediction of x_i^u by the teacher model.

$$\mathcal{L}_{\text{student}} = \sum_{i=1}^{N_{\text{uns}}} \mathcal{L}_{\text{cls}}(x_i^u, y_i^u) + \mathcal{L}_{\text{bbox}}(x_i^u, y_i^u) + \mathcal{L}_{\text{iou}}(x_i^u, y_i^u) \quad (2)$$

4. **EMA:** The student model is trained with the pseudo-labelled samples D'_{uns} and the teacher model is updated using EMA during the process, which is defined in Equation 3.

$$\theta_t = (1 - \alpha)\theta_s + \alpha\theta_t, \quad (3)$$

Where α is the hyperparameter of the smoothing coefficient, θ_s are the weights of the student model, and θ_t are the weights of the teacher model. This process is performed for all weights and is applied after each training step (iteration of a batch of samples) of the student model.
5. **Conditional component:** Check if there has been any improvement in the teacher model over D_{ctrl} to save the new $\theta_{t'}$ in θ_t or ignore the cycle; this process was named "conditional component". The conditional component checks

whether each training cycle between the teacher and student models does not worsen the previous one in the set D_{ctrl}. After each cycle, D_{ctrl} is run on $\theta_{t'}$, and the AP is calculated. If the AP improves, a new θ_t is saved and again generates a new D'_{uns} to improve the next training of θ_s. Otherwise, $\theta_{t'}$ is restored to the previous best state (θ_t) and θ_s is reset and re-trained on a new cycle. This conditional component is defined in Equation 4.

$$\begin{cases} \theta_t \leftarrow \theta_{t'} & if\ AP_{t'} > AP_t \\ \theta_{t'} \leftarrow \theta_t & if\ AP_{t'} \leq AP_t \end{cases} \tag{4}$$

where θ_t are the weights of the teacher model in the best cycle, $\theta_{t'}$ are the weights of the teacher model after a new cycle of training.

These steps of the iterative cooperative phase are repeated until the teacher model converges, which means that the training of the student model does not generate further improvement over the teacher model in a number of attempts defined by the parameter attempts over the control set.

4. Experimental Setup

This section will analyse the datasets used, present the selected detection network, and explain the experimental setup for implementing this methodology.

4.1. Datasets

To accomplish the requirements of the proposal presented in this study, a set of annotated images for the warm-up phase and evaluation, and a set of non-annotated images for the iterative learning phase, are required for performing proper semi-supervised training. Therefore, as D_{sup}, we used the dataset published by the University of Granada (UGR) and presented in the Olmos et al. study [44], which contains gun images selected from different internet portals, which include mainly individual and multiple appearances of guns in close-up and medium shots. Although this dataset consists of 3,000 images, for this study, we split the dataset into a training set with 80%, (2,400 images), a validation set with 10% (300 images), and D_{ctrl} with 10% (300 images). However, a random subset of 300 of the 2,400 images in the training set was selected for parameter exploration experiments. This reduced set allowed us to analyse the behaviour of the model parameters with greater variety due to the computational time required. Figure 2 (a) shown a sample of this dataset in the upper left-hand corner.

For a more extensive comparison, an additional dataset of 5,000 frames of weapons from YouTube videos, published by Gu et al. [45] under the name YouTube-GDD, was also used as D_{sup} to compare with models from other studies. For this study, we split the training set with 70%, resulting in 3,500 images, validation set with 10% and control set with 10%. The remaining 10% corresponds to a test partition that the authors did not publish. Therefore, it was not considered for this study. Figure 2 (a) shows a sample of this dataset in the upper right-hand corner.

As D_{uns}, we created an unsupervised dataset by scraping 458,599 images from Instagram[7], with the hashtag "handgun" in their description, resulting in the largest unsupervised weapons dataset to date. Figure 2 (a) shows a sample of this dataset at the bottom. The hashtag is written by the user posting the image on Instagram, and it may indicate that what is described is included in the image. As can be seen, these images present higher variability compared with stock images or images from encyclopaedias.

Although our unsupervised dataset consists of 458,599 images, a pre-filtering was performed using the teacher model with a warm-up of 20 epochs to discard those images that do not produce any detection. This procedure avoids unnecessary costs in time and memory required for pseudo-labelling, resulting in an unsupervised dataset of 283,468 images since many users include specific hashtags to gain visibility. Hence, the object of the hashtag is not always included in the image

Nonetheless, this pre-filtering may not be entirely accurate, so images without firearms could be found in the pseudo-labelled dataset. This problem could be solved by limiting the confidence threshold of the detection; nonetheless, as demonstrated in the following experimental section, restricting the dataset to confident images is not desirable.

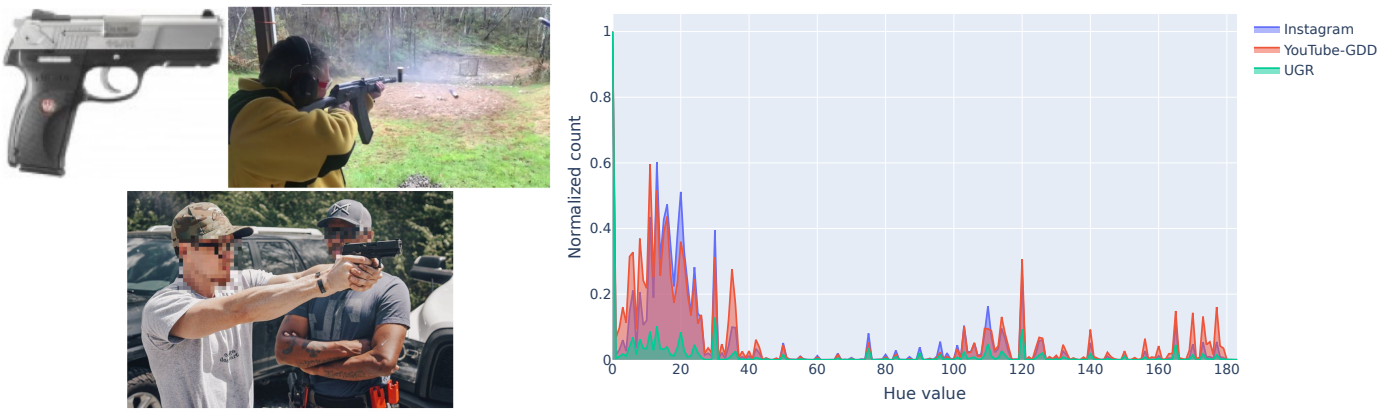
Median histogram of the images was calculated in HSV colour space per dataset to study the differences between the different datasets and prove if these new unsupervised samples present a higher variability, showing the distribution of hue, saturation and value per set; this is shown in Figure 2 (b). It can be observed that YouTube-GDD and unsupervised Instagram datasets present better colour variation than the UGR dataset, which indicates that the unsupervised dataset can improve over the UGR dataset as it will provide more variability. On the other hand, we see that the YouTube dataset shows a possible high presence of long guns since their mean aspect ratio of bounding boxes is 1.97, compared with 1.33 in the UGR dataset, which has a lower presence of long guns. Nonetheless, a ratio close to one cannot guarantee the absence of rifles in the dataset, as a diagonal rifle's bounding box may have a ratio close to one; on the other hand, a significantly different value, as in this case, confirms a different form of weapons between the two datasets.

The current state-of-the-art detection performance on these two datasets is given by Gu et al. [45] and Salazar et al. [20], whose maximum results can be seen in Table 1. Nevertheless, the results of Gu et al. are obtained using the non-available test split; hence, they are not directly comparable with the results of this study.

4.2. Detection network

In our implementation of Conditioned Cooperative Training, we used the Deformable DETR architecture [2] with a backbone of ResNet-50 [46]. The choice of a transformer-based architecture is due to its recent high impact on the state-of-the-art, as well as its good performance on large datasets, superior to architectures based mainly on convolutional layers, such as

[7]Instagram: https://www.instagram.com/



(a) (b)

Figure 2: Sample images (a) of datasets, UGR in the upper left-hand corner, YouTube-GDD in the upper right-hand corner and Instagram at the bottom; and median hue histogram (b) of datasets.

Dataset	Study	Method	AP	AP50	AP75
YouTube-GDD	Gu et al. [45]	YOLOv5s	52.1	77.3	-
UGR	Salazar et al. [20]	Faster R-CNN-FPN	68.3	92.6	74.1

Table 1: State-of-the-art results with supervised YouTube-GDD [45] and UGR [44] datasets.

Faster R-CNN [47] or YOLO [48, 49], which makes it suitable for this problem. Despite the possibility that the architecture of our methodology could be changed, it has not been demonstrated to be beneficial in this study, so there is no assurance that it will work with good predictive performance when the architecture is changed.

To properly compare our methodology, we implemented the architectures of four studies: UP-DETR, DETReg, Unbiased Teacher and YOLO. The first two frameworks are based on a self-supervised learning approach combined with supervised fine-tuning, which allows us to compare our methodology with a different approximation. Unbiased Teacher is based on a semi-supervised learning approach, which enables us to compare ourselves with an approach similar to ours. The supervised learning architecture YOLO is generally used for real-time object detection, as it allows efficient inferences to be made. Due to this inference speed advantage, it is also often used for weapon detection. For this study, we have used YOLOv5 [50] with medium (YOLOv5m) and large (YOLOv5l) backbones.

4.3. Implementation details

To implement the methodology proposed in this study, the IceVision[8] framework with MMDetection[9] and Fastai[10] has been used and a new training flow with Deformable DETR has been created.

All experiments were carried out on a machine with an NVIDIA A100 card with 40 GB, an NVIDIA 3090 card with 24 GB, 32 GB of RAM and a CPU AMD Ryzen™ Threadripper™ PRO 3955WX.

For this study, we considered standard metrics in the field of object detection, such as AP (average precision), AP50 (average precision with IoU >0.50), AP75 (average precision with IoU >0.75) and AR (average recall), which are the metrics used in the COCO[11] detection challenge.

For best performance, in our approach, pre-trained weights from "ImageNet-1k" were used to transfer learning in the ResNet-50 backbone, and fine-tuning was performed by running an epoch with these weights frozen during the warm-up phase. However, in Unbiased Teacher and YOLOv5, the best weights provided by their authors from COCO dataset were used as pre-training. On the other hand, in UP-DETR and DETReg, both based on self-supervised learning with a pretext task that generates the pre-training weights, this is obtained with self-supervised training performed with D_{uns}. However, their ImageNet pre-training weights were also used to compare fine-tuning with both pre-trainings.

A learning rate of $1e^{-4}$ was used for the warm-up stage. For iterative training, an adaptive learning rate was calculated before executing each cycle using the Fastai optimal learning rate search function based on Smith's study [51]. Nonetheless, fixed learning rate of $1e^{-5}$ was established for experiments in which a direct comparison was needed; this is indicated in the experiment description. All training was carried out using a variant of the 1-cycle optimiser policy [52], a proposal of Smith and Topin [53]; applied with some modifications by Howard and Gugger [54], which allows setting the hyperparameters that significantly reduce training time and improve performance. This

[8] IceVision: https://airctic.com/
[9] MMDetection: https://github.com/open-mmlab/mmdetection
[10] Fastai: https://github.com/fastai/fastai
[11] COCO Detection Evaluation: https://cocodataset.org/#detection-eval

policy is based on a learning rate schedule that starts from an initial value to an established maximum and then to a minimum value much lower than the initial learning rate, repeated for a certain number of epochs or steps.

The batch size was set at 6 for all experiments, and the EMA parameter α was set at 0.999. The parameters of our methodology are set for global comparison as dynamic δ, N_{uns} = 30,000 and EPC = 5. By contrast, for the parameter exploration phase, these values change according to the explored parameter and are detailed in each section.

For parameter exploration, all images used were resized to 512x512 with padding transformation and without data augmentation to reduce complexity and allow us to perform several iterations on each experiment. However, for the global comparison of methodologies, the size of the images was set to 800x800 with padding and random transformations as data augmentation, being: image resize, horizontal flip, shift scale rotation, RGB shift, random brightness contrast, and blur. All transformations are applied with a probability value of 0.5 or 1, depending on the type of transformation applied.

A single class called "pistol" was used since, for this problem, it is not necessary to identify any other class. Furthermore, it is the only weapon class available in both supervised datasets. Nevertheless, our method is not limited to a single class: a higher number of classes is supported and will be explored in future work.

5. Experiments

A series of experiments were carried out to evaluate the performance of the methodology proposed in this study, with an exploration of the most relevant parameters and a global comparison between different methodologies and datasets.

5.1. Parameter exploration

The three most relevant parameters of the methodology proposed in this study are δ, N_{uns}, and EPC. Here, we present a study of each of them, exploring how they affect the system. For this purpose, as discussed in Section 4.1, a subset of random samples from the UGR dataset will be used for this exploration. The behaviour of this parameter exploration has been studied using a single dataset, so the results may vary between different datasets. This exploration was only conducted to evaluate the relevance of the components of our methodology, and is therefore not comparable with a full set of this or any other dataset.

5.1.1. Warm-up phase

The first experiment studies the importance of an adequately supervised training or warm-up phase before the iterative phase. As the optimiser applied during training depends on the number of epochs selected due to the 1-cycle policy, described in Section 4.3, different training sessions were carried out, varying the number of epochs since the choice of the number of epochs influences the result.

Table 2 reports different executions based on the number of epochs. It can be observed how the model performs with a different number of epochs, being beneficial a training with 500 epochs but detrimental with 600 epochs since it may be causing overfitting with the training data, which leads to a degradation in the validation AP. Therefore, the training weights with 500 epochs will be used for the following experiments to avoid overfitting.

Experiment ID	Epochs	AP	AP50	AP75	AR
swarmup-20ep	20	56.5	78.1	59.1	70.9
swarmup-100ep	100	56.5	80.7	60.1	**71.6**
swarmup-300ep	300	59.5	**81.9**	**64.0**	70.7
swarmup-500ep	500	**61.0**	81.6	63.2	71.5
swarmup-600ep	600	58.7	80.5	62.9	70.1

Table 2: Supervised training with the different number of epochs. Experiment ID, epochs, AP, AP50, AP75 and AR are shown.

5.1.2. Ablation study

As mentioned in Section 3, two novel techniques were presented in this study: a conditioning module to avoid confirmation bias in the student-teacher strategy and a threshold-finder, which selects the best confidence threshold for each training cycle. In this section, we examine both methods using an ablation study.

We performed an ablation study to demonstrate the predictive performance of our student-teacher strategy with the conditioning module. We used 30 epochs in 10 cycles with 3 EPCs during ten different iterations. Figure 3 shows the median AP obtained during these 30 epochs with and without our student-teacher strategy. Our method achieves higher AP in fewer epochs and decreases AP degradation over time, demonstrating the superiority of our method.

The delta parameter is also of great importance in semi-supervised learning systems, as this value will indicate the confidence of the images used for learning, so it will be a critical parameter that will help improve or worsen the model. For this reason, it was decided to include a search function to find the optimal δ in each cycle. First, the performance of the threshold-finder method will be evaluated. Second, an example of how the distribution of AP changes by δ before and after a cycle will be shown.

Figure 4 shows the median AP over 10 cycles in 10 different iterations, using: EPC = 3, N_{uns} = 5000, a dynamic δ in the threshold-finder case and a δ of 0.7, 0.8, and 0.9 in the static threshold case. Higher thresholds, such as δ of 0.99, were not possible to consider, as they did not reach the minimum number of images for this experimentation, so the highest threshold considered was δ of 0.9. The threshold-finder search was performed with 300 images, five epochs per δ, and an increase in δ of 0.1. The result in this figure shows how, although static δ can reach a high value in the first and second cycles, it does not improve or stabilise, deteriorating the performance obtained in the supervised training. Moreover, the static δ of 0.9 produced

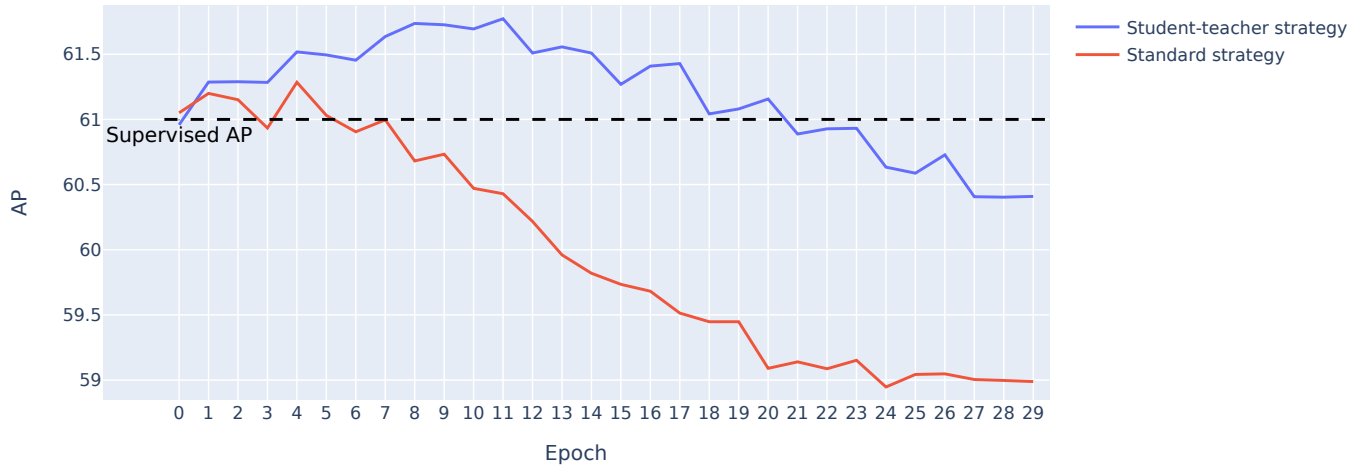


Figure 3: Ablation study of our student-teacher strategy with conditioning module during 30 epochs, with a median of AP values over ten iterations.

a deterioration greater than that found with δ of 0.7 and 0.8, obtaining an AP lower than the supervised one in the second cycle, indicating that the model does not learn better characteristics when supplied with high thresholds.

The search function was run in two cycles, before and after applying a δ value, to illustrate how AP varies according to selected δ. Figure 5 shows the results of this experiment obtained by calculating the median of all epochs of the corresponding δ with that precision. In these first results, the distribution presented has a maximum value at $\delta = 0.6$, gradually decreasing as we move away from this value. Note how the lower values of δ produce a much lower gain than the rest, possibly due to the low quality of the annotations obtained with shallow confidence.

Furthermore, to observe how median AP changes over δ values after the second cycle, the first cycle with a δ of 0.6 was performed, as this was the best result in the previous cycle. The threshold-finder was executed again to analyse how the distribution of the δ values changed, being these results reflected in Figure 5 (b).

Comparing both threshold-finder results in Figure 5, it shows a high degradation in performance of δ values greater than 0.55, which may be caused by previous cycle training performed with the δ value of 0.6, with which the model has already learnt enough. The δ values lower than 0.6 obtain better results, giving a maximum value at a threshold of 0.55.

However, the results of the second cycle, reflected in Figure 5 (b), present lower AP values than those found in the previous cycle (a). One reason for this could be the number of images used, which was set low (i.e. 300 images) to allow us to calculate the median of several epochs and obtain more reliable results. Therefore, these AP values are only comparable within this experiment.

Figure 6 shows an example of images with low confidence scores (i.e., between 0.50 and 0.55) and images with high confidence scores (i.e. between 0.95 and 1). From these images, we can observe that those with a low confidence score, even though they are less accurate, present different angles of the weapon, more variable backgrounds, a greater distance to the camera, the inclusion of people holding the weapon and the appearance of long weapons. On the other hand, in the images with a high confidence score, we find a more accurate annotation along with less variable backgrounds, closer proximity to the camera, fewer long weapons, and less confusing backgrounds. Nevertheless, when a low δ is set, most of the pseudo-labels generated have a high confidence score, except that by setting a low δ, we allow some of these more confusing images to be included, thus adding more variability to the set of pseudo-labels. Figure 7 shows an example of the distribution obtained with the pseudo-labels that can be seen with $N_{uns} = 300$ and $\delta = 0.5$.

An interesting conclusion of this experimentation is that it can be observed how, in the first cycle, our model obtains better results using lower δ (that is, δ with a value of 0.6) than expected. Since a high δ (i.e., δ with a value of 0.95) would expect better results by having more accurate annotations. However, in this experiment, we can observe that this is not the case, and this could be due to the need for the model to learn from unfamiliar images. Furthermore, with the threshold-finder method, an improvement of +1 points in AP was obtained compared to the result obtained in the warm-up phase. A degradation of -1.45 points in AP without the threshold-finder method was also obtained, proving the importance of this method.

5.1.3. Number of unsupervised images

The number of unsupervised images N_{uns} to be used is also an important decision to consider in the presented system, as it

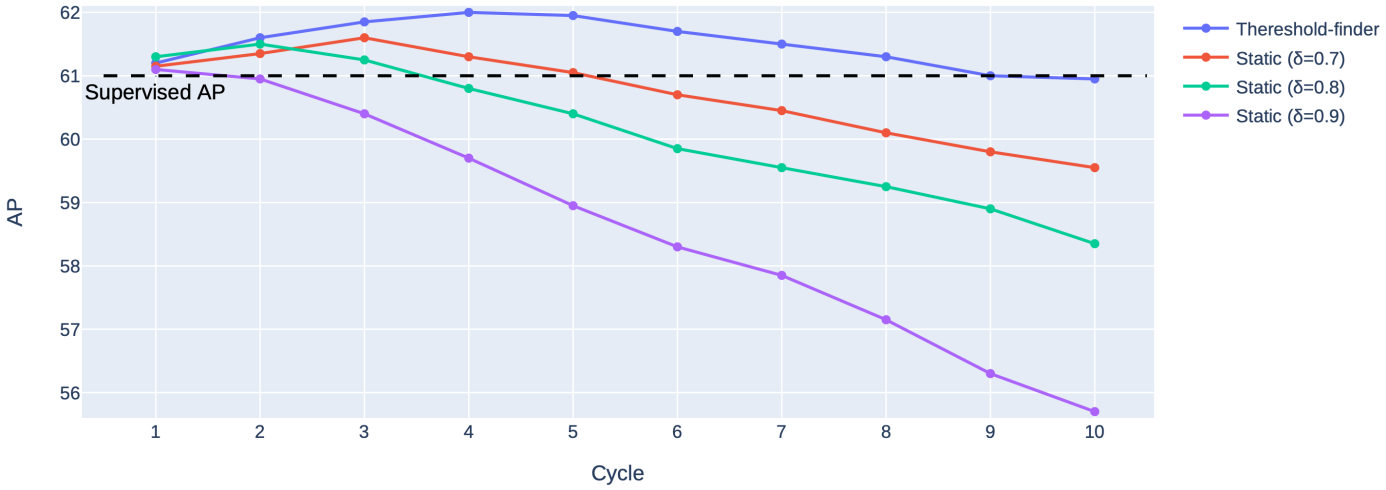
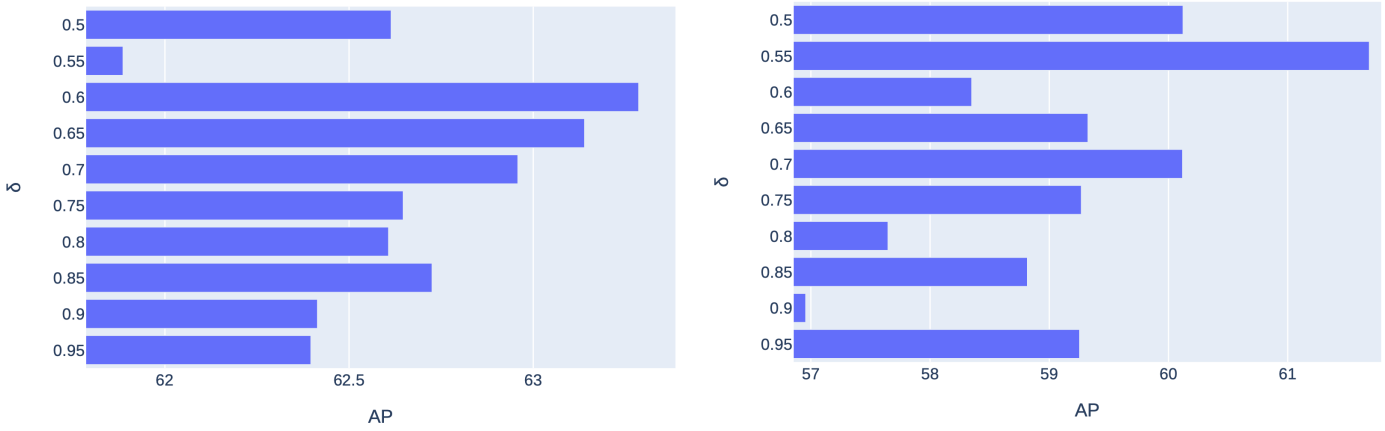


Figure 4: Ablation study of the threshold-finder method during ten cycles with a median of AP values over ten iterations.



(a) (b)

Figure 5: Median AP of the student model per δ in the first (a) and second (b) cycles.

can be limited by the source of the data or its acquisition capacity. Therefore, this experiment studies the relation between N_{uns} and the AP obtained in the student and teacher models.

Figure 8 shows the AP obtained in one cycle for both the teacher model and the student model with different N_{uns} over 10 different iterations with 5 epochs and a δ value of 0.6.

From these results, presented in Figure 8, it can be seen that both teacher and student models tend to achieve better results with a more extensive set of unsupervised images, as more variability is provided for training. However, there is a point (greater than 30,000 images) at which the model obtains less improvement when using many images. This fact is possibly due to the number of steps performed, as the more images per epoch, the more steps performed; thus, 30,000 images could give a better generalisation.

Therefore, in this experiment, setting N_{uns} at 30,000, an improvement of +0.86 points was obtained in AP compared to the result obtained in the warm-up phase, which proves the importance of this parameter.

5.1.4. Epochs Per Cycle (EPC)

This experiment studies the importance of Epochs per Cycle (EPC), as it is necessary to identify how much the student model must be trained in an unsupervised way to improve the teacher model. Therefore, a search for EPC was carried out, running 2-cycle training runs on different EPC values to study the model's behaviour under variations of this parameter. The results are shown in Figure 9. These training executions were performed ten times with different seeds for each EPC value, and the median AP of the last cycle of each EPC value was measured. The supervised phase "swarmup-500ep", δ fixed at 0.6 and N_{uns} of 30,000 images, was chosen, as these values obtained the best results in the previous experiments.

In Figure 9, it can be seen that EPC values lower than 3 obtain the best results, being the maximum AP got with EPC = 2. Achieved an improvement of +0.94 points in AP compared to the result obtained in the warm-up phase, but also a degradation of -0.81 points in AP with EPC = 10 was produced. The main factor for this degradation with high EPC values is due to over-training with unreliable data, which results in a de-

(a) Confidence score of 0.50-0.55.

(b) Confidence score of 0.95-1.00.

Figure 6: Sample pseudo-labels with low (a) and high (b) confidence scores of D_{uns}.

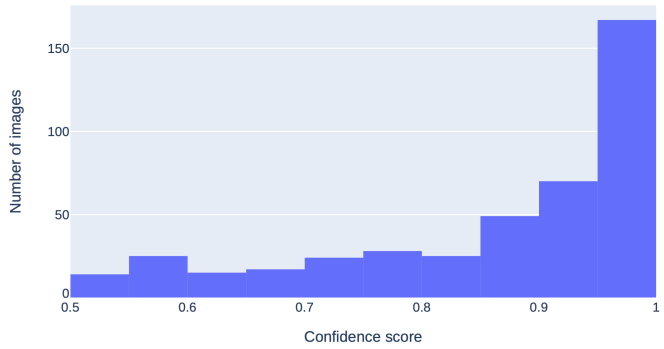


Figure 7: Distribution of number of images per confidence scores with N_{uns} = 300 and δ = 0.5.

terioration of predictive performance, worsening the weights previously obtained in a supervised manner. Therefore, using pseudo-labelled data for a few epochs is beneficial, but not beneficial if the model is over-trained with these images. These results proved how the correct selection of this value can lead to a significant improvement or deterioration of the model to lower results than the supervised one.

5.2. Comparison of methodologies

To evaluate the predictive performance of the methodology proposed in this study, four state-of-the-art frameworks, YOLOv5 [50], UP-DETR [6], DETReg[40] and Unbiased Teacher [4], were run on two different supervised datasets, presented in Section 4.1 during five different iterations. Tables 3 and 4 present the predictive performance of the frameworks on the UGR and YouTube-GDD datasets respectively, indicating the use or non-use of the D_{uns} dataset with the column "Unlabelled firearms", as well as the different metrics indicated in Section 4.3. In both Conditioned Cooperative Training and Unbiased Teacher, there is a warm-up phase where only D_{sup} is used, then a training phase where D_{uns} is used. However, in UP-DETR and DETReg, D_{uns} is applied during a pre-training phase, followed by fine-tuning with D_{sup}. To compare it with the absence of D_{uns}, training was also performed using the weights with ImageNet from the pre-training phase published in their study.

The parameters selected in our methodology for both datasets in this comparison were: 100 epochs for the warm-up phase; a dynamic δ, N_{uns} = 30,000 and EPC = 5. The EPCs differ from the exploration carried out in Section 5.1 since the complete set is used here, and this value performed better under these conditions. Therefore, finding a new optimal EPC value is necessary if the number of supervised images changes.

In Table 3, it can be seen how our methodology presents a mean improvement of 7.42 points in AP and 12.1 points in AR compared to the semi-supervised framework Unbiased Teacher. Comparing our architecture with the UP-DETR and DETReg self-supervised learning frameworks, we again obtain an improvement, in this case of 1.22 and 2.8 points in AP, respectively. Nonetheless, no improvement was obtained in AR. Moreover, when compared to other more common supervised methodologies in weapons detection, such as YOLOv5 with medium and large backbones, we find an average improvement of 19.86 and 6.52 points in AP, and 21.18 and 14.12 points in AR, respectively.

On the other hand, in Table 4 our methodology presents a mean improvement of 10.5 points in AP and 12.76 points in AR compared to Unbiased Teacher and a mean gain of 1.88 points in AP compared to DETReg. Nevertheless, on this dataset, our methodology obtains an AP similar to UP-DETR, a fact that will be discussed in the next section. Finally, we obtain an average improvement of 11.98 and 3.28 points in AP, and 23.40 and 14.62 points in AR, respectively, when compared to the supervised architecture YOLOv5 with medium and large backbones.

Furthermore, with the results obtained in Table 3 and Table 4, we can observe the improvement produced by our methodology over supervised training, represented in the table with the column "Unlabelled firearms" of our Conditioned Cooperative Training methodology with "No" for a supervised training approach and "Yes" with the execution of the conditioned cooperative training on unlabelled data. In this comparison, we obtained a benefit of 1.86 and 0.28 points in AP, and 2.44 and 0.34 points in AR, with the UGR and YouTube-GDD datasets, respectively. This improvement is more significant with the UGR dataset, potentially due to the more considerable difference between the unsupervised and UGR dataset, observed previously in Section 4.1.

Although UP-DETR presents similar results to those achieved with our methodology, UP-DETR follows a different learning methodology to ours, as does DETReg, since they are

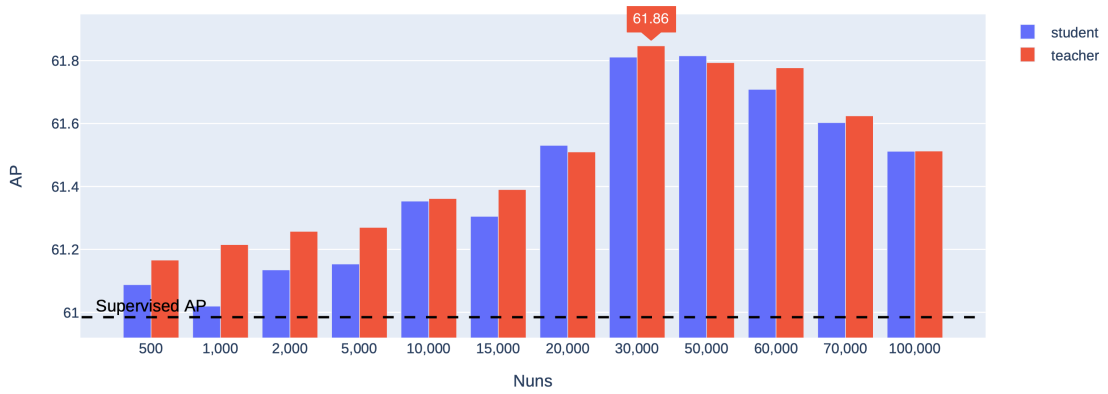


Figure 8: Results of median AP for ten different iterations with five epochs on the first cycle with different N_{uns}.

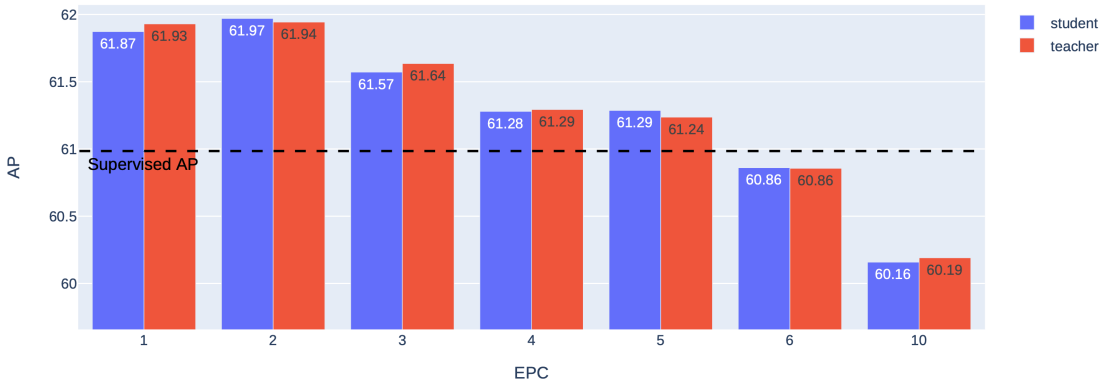


Figure 9: Results of median AP over ten iterations with two cycles comparing different EPCs.

based on self-supervised learning to perform the training. A self-supervised learning approach requires a prior pre-training on the entire unsupervised dataset, which increases the computational cost needed to train the model. In the Figure 10, we can see the amount of training time required to achieve the results obtained with the YouTube-GDD dataset. Resulting in our methodology being six times more efficient than UP-DETR with similar or better results, depending on the dataset used.

6. Discussion of results

After the results obtained in the previous section, it can be observed that the proposed methodology improves in all cases over supervised training and even reaches or surpasses the results of state-of-the-art implementations.

Compared to Unbiased Teacher, our methodology improves the UGR and Youtube-GDD datasets by 7.42 and 10.5 points in AP and 12.1 and 12.76 in AR, respectively, which is a significant improvement over a state-of-the-art framework in semi-supervised learning when applied to weapon detection.

Even though our methodology has some similarities with Unbiased Teacher, this is not reflected in the experimentation. This is, to our knowledge, due to several factors. The first one is the detection model they use, Faster R-CNN, an object detection model that, although it achieves good results with small and medium datasets, fails to learn more on large datasets. Contrary to architectures based on Transformers as DETR or Deformable DETR, this model fails to learn more on large datasets. On the other hand, they perform combined training with the supervised and unsupervised data, which can lead to overfit the supervised data if limited supervised data is presented, as in our datasets. In contrast to our methodology, which separates the training of these sets into phases. These differences, in addition to the advantage of our novelties, can lead to the difference that is reflected in experimentation.

Regarding UP-DETR, we improved 1.22 points in AP on the UGR dataset and an almost similar UP-DETR AP on the YouTube-GDD dataset, which is also an achievement, as it is not the same approach being a methodology based on self-supervised learning. This difference in our methodology between these datasets is probably because semi-supervised learning, rather than self-supervised learning, requires the unlabelled dataset to be similar to the supervised set. Since the Instagram dataset differs more from the YouTube-GDD dataset, resulting in less improvement than the UGR dataset. On the other hand, in self-supervised learning with fine-tuning, such as the one found in UP-DETR, a pre-training is performed on the unlabelled data, which increases the time needed for training by a factor of six times compared to our methodology. Then the model is fitted to the supervised data, which is more closely

Methodology	Unlabelled firearms	AP	AP50	AP75	AR
YOLOv5m [50]	No	51.46 ± 0.63	82.14 ± 0.97	53.28 ± 1.86	59.26 ± 0.57
YOLOv5l [50]	No	64.80 ± 0.95	89.60 ± 1.27	73.12 ± 2.38	66.32 ± 1.07
UP-DETR [6]	No	67.09 ± 0.42	88.61 ± 0.78	69.99 ± 0.77	80.08 ± 0.62
	Yes	70.10 ± 0.54	92.63 ± 0.23	73.63 ± 1.08	80.97 ± 1.21
DETReg [40]	No	68.52 ± 0.37	89.15 ± 0.31	72.65 ± 0.93	81.23 ± 0.08
	Yes	66.05 ± 0.40	88.09 ± 0.89	70.13± 1.31	79.36 ± 0.49
Unbiased Teacher [4]	No	59.75 ± 2.21	89.63 ± 0.81	62.51 ± 3.01	66.66 ± 1.12
	Yes	63.90 ± 1.25	92.53 ± 0.44	66.51 ± 2.22	68.34 ± 1.19
Conditioned Cooperative Training	No	69.46 ± 1.32	92.46 ± 0.41	75.54 ± 2.21	78.00 ± 0.68
	Yes	71.32 ± 1.00	92.88 ± 0.42	77.72 ± 1.71	80.44 ± 0.93

Table 3: Comparison of methodologies with the UGR dataset with the methodology's name, the use or non-use of D_{uns}, and the detection metrics during five iterations.

Methodology	Unlabelled firearms	AP	AP50	AP75	AR
YOLOv5m [50]	No	46.58 ± 2.12	78.70 ± 1.78	47.80 ± 6.38	46.50 ± 0.70
YOLOv5l [50]	No	55.28 ± 0.73	76.46 ± 1.01	58.00 ± 1.19	55.28 ± 0.53
UP-DETR [6]	No	52.87 ± 0.66	69.47 ± 0.31	56.51 ± 1.23	68.21 ± 0.28
	Yes	58.87 ± 0.60	74.05 ± 0.38	61.88 ± 0.97	72.13 ± 0.53
DETReg [40]	No	56.68 ± 0.51	73.77 ± 0.64	61.92 ± 0.69	74.65 ± 0.39
	Yes	53.12 ± 0.42	72.15 ± 0.93	59.19 ± 0.85	72.50 ± 0.39
Unbiased Teacher [4]	No	41.36 ± 2.32	71.81 ± 0.42	42.21 ± 4.02	53.24 ± 1.48
	Yes	48.06 ± 0.52	75.56 ± 0.88	51.54 ± 1.09	57.14 ± 0.88
Conditioned Cooperative Training	No	58.28 ± 0.84	77.70 ± 0.81	62.78 ± 1.66	69.56 ± 1.82
	Yes	58.56 ± 1.09	77.26 ± 0.22	63.22 ± 2.03	69.90 ± 2.13

Table 4: Comparison of methodologies with the YouTube-GDD dataset with the methodology's name, the use or non-use of D_{uns}, and detection metrics during five iterations.

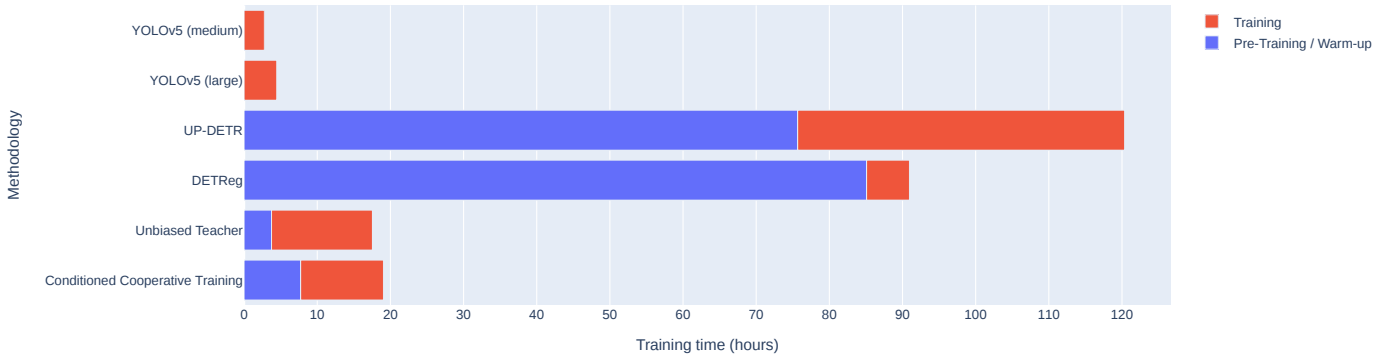


Figure 10: Training time in hours per methodology during pre-training or warm-up and training phases with YouTube-GDD dataset.

related to the set to be evaluated.

Analysing the results obtained by DETReg, we observe that its methodology does not produce an improvement when applying a pre-training of weapons. This fact may be due to the use of the selective search that they implemented and tested in their study with generic datasets, where they showed that it im-

proved other methodologies such as UP-DETR; however, this is not reflected when applied to weapons datasets. On the other hand, our methodology also improves the results obtained by DETReg, achieving a mean increase of 2.8 points in AP with the UGR dataset and 1.88 points in AP with the YouTube-GDD dataset. Since DETReg is based on a self-supervised learning system, it requires an expensive pre-training, which significantly increases the total training time. Nevertheless, DETReg is based on Deformable DETR instead of DETR, which results in a less expensive fine-tuning.

From the results, without D_{uns}, Deformable DETR achieves a significant improvement in both datasets compared to Faster R-CNN and DETR models used, respectively, by Unbiased Teacher and UP-DETR; proving that Deformable DETR provides better predictive performance in weapons detection.

In addition, an exciting conclusion observed is that, generally, when training a specific detection system, as in the case of weapon detection, pre-trained weights were used with large general datasets, such as ImageNet or COCO. However, it can be observed that by pre-training the self-supervised system UP-DETR, with a large set of images collected with images more related to the specific classes, a significant improvement is achieved, improving from 52.87 to 58.87 (+6.00) just by changing the initial weights.

In conclusion, the methodology proposed in this study, includes two new elements, a conditioning module for the iterative learning process to prevent bias during the training cycles and a threshold-finder method to select the best confidence threshold for each training cycle. We outperformed the state-of-the-art semi-supervised Unbiased Teacher methodology for handgun detection by up to 10.5 points, significantly improving the field of firearm detection. Furthermore, we have verified that our methodology achieves a higher prediction performance in some cases than other types of approach with less training time required, such as self-supervised learning, using the state-of-the-art UP-DETR and DETReg, and fully supervised learning, using YOLOv5.

7. Conclusions

In this study, we have developed a new semi-supervised learning methodology based on Conditioned Cooperative Training that allows controlled cooperation of two models using the learning of unlabelled dataset features. This proposal has introduced two novelties: a) the addition of a conditioning module to the iterative learning process between the teacher and student models and b) the implementation of a threshold-finder method, which selects the best confidence threshold for each training cycle.

A new unlabelled dataset collected from Instagram hashtags with 458,599 images was used to implement and compare our methodology with state-of-the-art semi- and self-supervised learning architectures. We achieved a mean AP of 71.32 in UGR and 58.56 in Youtube-GDD datasets, improving by up to 10.5 points against the state-of-the-art semi-supervised architectures. Through this improvement, we aim to advance the field of early threat detection through computer vision by detecting weapons in time to reduce the number of assaults with firearms.

Furthermore, with the results obtained, it has been proven that fine-tuning with pre-training of object-specific self-supervised learning achieves better predictive performance than pre-training weights of COCO or ImageNet. Thus, the presented approach is a new and better alternative to fine-tuning in small supervised sets with pre-trained weights from generic objects. As far as we know, this is the first study to demonstrate the improvement of this pre-training approach as a better alternative to specific object detection.

Despite the improvements achieved in this study, there are still challenges to overcome. The most significant challenge to resolve is the lack of progress in improving the models after several cycles; this problem is caused by the fact that it is impossible to train very uncertain images, as they have imprecise annotations. Possible solutions to this challenge could use traditional techniques to segment objects by their morphology and edge detection or search for regions of interest. A further possible improvement is the inclusion of detectors of violent actions or aggressive poses in our architecture. This addition will allow the system to obtain different types of information that will enable it to decide whether it is under serious threat due to the characteristics of the object, the pose detected, or the violent action.

8. Acknowledgements

This research is partially supported by DISARM project ('Automatic Detection of Armed Individuals' - Grant n. PDC2021-121197) and HORUS project ('Hybrid Object Recognition platform for Weapon Seek' - Grant n. PID2021-126359OB-I00) funded by MCIN/AEI/10.13039/501100011033 and by the "European Union NextGenerationEU/PRTR". This research was also supported by grants from NVIDIA and utilised NVIDIA A100 donated to our colleague Miguel A. Martinez-del-Amor.

References

[1] P. Goyal, M. Caron, B. Lefaudeux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin, P. Bojanowski, Self-supervised Pretraining of Visual Features in the Wild, arXiv e-prints (2021) arXiv:2103.01988arXiv:2103.01988.

[2] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable DETR: Deformable Transformers for End-to-End Object Detection, arXiv preprint arXiv:2010.04159 (2020).

[3] E. Arazo, D. Ortego, P. Albert, N. E. O'Connor, K. McGuinness, Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8.

[4] Y.-C. Liu, C.-Y. Ma, Z. He, C.-W. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, P. Vajda, Unbiased Teacher for Semi-Supervised Object Detection (2021). arXiv:2102.09480.

[5] A. Bar, X. Wang, V. Kantorov, C. J. Reed, R. Herzig, G. Chechik, A. Rohrbach, T. Darrell, A. Globerson, DETReg: Unsupervised Pretraining with Region Priors for Object Detection (2021). arXiv:2106.04550.

[6] Z. Dai, B. Cai, Y. Lin, J. Chen, UP-DETR: Unsupervised Pre-training for Object Detection with Transformers, in: Proceedings of the IEEE/CVF

conference on computer vision and pattern recognition, 2021, pp. 1601–1610.

[7] M. Grega, A. Matiolanski, P. Guzik, M. Leszczuk, Automated Detection of Firearms and Knives in a CCTV Image, Sensors 16 (1) (JAN 2016). doi:10.3390/s16010047.

[8] R. Olmos, S. Tabik, F. Herrera, Automatic handgun detection alarm in videos using deep learning, Neurocomputing 275 (2018) 66–72. doi:https://doi.org/10.1016/j.neucom.2017.05.012.

[9] M. T. Bhatti, M. G. Khan, M. Aslam, M. J. Fiaz, Weapon Detection in Real-Time CCTV Videos Using Deep Learning, IEEE Access 9 (2021) 34366–34382. doi:10.1109/ACCESS.2021.3059170.

[10] T. Lu, Y. Wang, Y. Zhang, J. Jiang, Z. Wang, Z. Xiong, Rethinking prior-guided face super-resolution: a new paradigm with facial component prior, IEEE Transactions on Neural Networks and Learning Systems (2022).

[11] Y. Wang, T. Lu, Y. Zhang, Z. Wang, J. Jiang, Z. Xiong, Faceformer: aggregating global and local representation for face hallucination, IEEE Transactions on Circuits and Systems for Video Technology (2022).

[12] T. Lu, Y. Wang, Y. Zhang, Y. Wang, L. Wei, Z. Wang, J. Jiang, Face hallucination via split-attention in split-attention network, in: Proceedings of the 29th ACM international conference on multimedia, 2021, pp. 5501–5509.

[13] N. Vallez, A. Velasco-Mata, O. Deniz, Deep autoencoder for false positive reduction in handgun detection, Neural Computing and Applications 33 (11) (2021) 5885–5895.

[14] J. Ruiz-Santaquiteria, A. Velasco-Mata, N. Vallez, G. Bueno, J. A. Álvarez García, O. Deniz, Handgun Detection Using Combined Human Pose and Weapon Appearance, IEEE Access 9 (2021) 123815–123826. doi:10.1109/ACCESS.2021.3110335.

[15] A. Lamas, S. Tabik, A. C. Montes, F. Pérez-Hernández, J. García, R. Olmos, F. Herrera, Human pose estimation for mitigating false negatives in weapon detection in video-surveillance, Neurocomputing (2022). doi:https://doi.org/10.1016/j.neucom.2021.12.059.

[16] F. Pérez-Hernández, S. Tabik, A. Lamas, R. Olmos, H. Fujita, F. Herrera, Object Detection Binary Classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance, Knowledge-Based Systems 194 (2020) 105590. doi:https://doi.org/10.1016/j.knosys.2020.105590.

[17] F. J. Rendón-Segador, J. A. Álvarez García, F. Enríquez, O. Deniz, ViolenceNet: Dense Multi-Head Self-Attention with Bidirectional Convolutional LSTM for Detecting Violence, Electronics 10 (13) (2021). doi:10.3390/electronics10131601.
URL https://www.mdpi.com/2079-9292/10/13/1601

[18] R. Olmos, S. Tabik, F. Perez-Hernandez, A. Lamas, F. Herrera, MULTICAST: MULTI Confirmation-level Alarm SysTem based on CNN and LSTM to mitigate false alarms for handgun detection in video-surveillance (2021). doi:10.48550/ARXIV.2104.11653.
URL https://arxiv.org/abs/2104.11653

[19] J. A. Álvarez-García, J. L. Salazar-González, O. Deniz, J. R. Santaquiteria-Alegre, Vision and Crowdsensing Technology for an Optimal Response in Security: Project results, in: 2021 IEEE International Mediterranean Conference on Communications and Networking (MediTCom), IEEE, 2021, pp. 82–87.

[20] J. L. Salazar-González, C. Zaccaro, J. A. Álvarez García, L. M. Soria-Morillo, F. Sancho, Real-time gun detection in CCTV: An open problem, Neural Networks 132 (2020) 297–308. doi:https://doi.org/10.1016/j.neunet.2020.09.013.

[21] L. Schmarje, M. Santarossa, S.-M. Schröder, R. Koch, A Survey on Semi-, Self- and Unsupervised Learning for Image Classification, IEEE Access 9 (2021) 82146–82168. doi:10.1109/ACCESS.2021.3084358.

[22] X. Qi, D. J. Foran, J. L. Nosher, I. Hacihaliloglu, Multi-Feature Semi-Supervised Learning for COVID-19 Diagnosis from Chest X-Ray Images, in: C. Lian, X. Cao, I. Rekik, X. Xu, P. Yan (Eds.), Machine Learning in Medical Imaging, Springer International Publishing, Cham, 2021, pp. 151–160.

[23] D. Hong, N. Yokoya, N. Ge, J. Chanussot, X. X. Zhu, Learnable manifold alignment (LeMA): A semi-supervised cross-modality learning framework for land cover and land use classification, ISPRS journal of photogrammetry and remote sensing 147 (2019) 193–205.

[24] H. Guo, Q. Shi, A. Marinoni, B. Du, L. Zhang, Deep building footprint update network: A semi-supervised method for updating existing building footprint from bi-temporal remote sensing images, Remote Sensing of Environment 264 (2021) 112589. doi:https://doi.org/10.1016/j.rse.2021.112589.

[25] X. Wu, D. Hong, J. Tian, R. Kiefl, R. Tao, A weakly-supervised deep network for DSM-aided vehicle detection, in: IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2019, pp. 1318–1321.

[26] X. Wu, W. Li, D. Hong, J. Tian, R. Tao, Q. Du, Vehicle detection of multi-source remote sensing data using active fine-tuning network, ISPRS Journal of Photogrammetry and Remote Sensing 167 (2020) 39–53.

[27] J.-C. Su, S. Maji, The Semi-Supervised iNaturalist-Aves Challenge at FGVC7 Workshop (2021). arXiv:2103.06937.

[28] K. Ohri, M. Kumar, Review on self-supervised image recognition using deep neural networks, Knowledge-Based Systems 224 (2021) 107090. doi:https://doi.org/10.1016/j.knosys.2021.107090.

[29] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum Contrast for Unsupervised Visual Representation Learning, arXiv preprint arXiv:1911.05722 (2019).

[30] X. Chen, H. Fan, R. Girshick, K. He, Improved Baselines with Momentum Contrastive Learning, arXiv preprint arXiv:2003.04297 (2020).

[31] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A Simple Framework for Contrastive Learning of Visual Representations (2020). arXiv:2002.05709.

[32] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al., Bootstrap your own latent-a new approach to self-supervised learning, Advances in neural information processing systems 33 (2020) 21271–21284.

[33] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, A. Joulin, Unsupervised learning of visual features by contrasting cluster assignments, Advances in Neural Information Processing Systems 33 (2020) 9912–9924.

[34] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, G. Hinton, Big Self-Supervised Models are Strong Semi-Supervised Learners (2020). arXiv:2006.10029.

[35] H. Pham, Z. Dai, Q. Xie, Q. V. Le, Meta pseudo labels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11557–11568.

[36] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, T. Pfister, A Simple Semi-Supervised Learning Framework for Object Detection, arXiv preprint arXiv:2005.04757 (2020).

[37] V. M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual domain adaptation: A survey of recent advances, IEEE signal processing magazine 32 (3) (2015) 53–69.

[38] L. Wang, K.-J. Yoon, Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[39] Z. Meng, J. Li, Y. Zhao, Y. Gong, Conditional teacher-student learning, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 6445–6449.

[40] A. Bar, X. Wang, V. Kantorov, C. J. Reed, R. Herzig, G. Chechik, A. Rohrbach, T. Darrell, A. Globerson, Detreg: Unsupervised pretraining with region priors for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14605–14615.

[41] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[42] C. Sammut, G. I. Webb, Encyclopedia of machine learning, Springer Science & Business Media, 2011.

[43] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 658–666.

[44] R. Olmos, S. Tabik, F. Herrera, Automatic handgun detection alarm in videos using deep learning, Neurocomputing 275 (2018) 66–72.

[45] G. Yongxiang, L. Xingbin, Q. Xiaolin, YouTube-GDD: A challenging gun detection dataset with rich contextual information, arXiv preprint arXiv:2203.04129 (2022).

[46] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[47] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

[48] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020).

[49] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Scaled-yolov4: Scaling cross stage partial network, in: Proceedings of the IEEE/cvf conference on computer vision and pattern recognition, 2021, pp. 13029–13038.

[50] G. Jocher, YOLOv5 by Ultralytics (5 2020). doi:10.5281/zenodo.3908559.
URL `https://github.com/ultralytics/yolov5`

[51] L. N. Smith, Cyclical learning rates for training neural networks, in: 2017 IEEE winter conference on applications of computer vision (WACV), IEEE, 2017, pp. 464–472.

[52] L. N. Smith, A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay, arXiv preprint arXiv:1803.09820 (2018).

[53] L. N. Smith, N. Topin, Super-convergence: Very fast training of neural networks using large learning rates, in: Artificial intelligence and machine learning for multi-domain operations applications, Vol. 11006, SPIE, 2019, pp. 369–386.

[54] J. Howard, S. Gugger, Fastai: a layered API for deep learning, Information 11 (2) (2020) 108.