FS-DETR: FEW-SHOT DETECTION TRANSFORMER WITH PROMPTING AND WITHOUT RE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper is on Few-Shot Object Detection (FSOD), where given a few templates (examples) depicting a *novel* class (not seen during training), the goal is to detect all of its occurrences within a set of images. From a practical perspective, an FSOD system must fulfil the following *desiderata*: (a) it must be used as is, without requiring *any* fine-tuning at test time, (b) it must be able to process an arbitrary number of novel objects concurrently while supporting an arbitrary number of examples from each class and (c) it must achieve accuracy comparable to a closed system. While there are (relatively) few systems that support (a), to our knowledge, there is no system supporting (b) and (c). In this work, we make the following contributions: We introduce, for the first time, a simple, yet powerful, few-shot detection transformer (FS-DETR) that can address both desiderata (a) and (b). Our system builds upon the DETR framework, extending it based on two key ideas: (1) feed the provided visual templates of the novel classes as visual prompts during test time, and (2) "stamp" these prompts with pseudo-class embeddings, which are then predicted at the output of the decoder. Importantly, we show that our system is not only more flexible than existing methods, but also, making a step towards satisfying desideratum (c), it is more accurate, matching and outperforming the current state-of-the-art on the most well-established benchmarks (PASCAL VOC & MSCOCO) for FSOD. Code will be made available.

1 INTRODUCTION

Thanks to the advent of deep learning, object detection has witnessed tremendous progress over the last years. However, the standard setting of training and testing on a closed set of classes has specific important limitations. Firstly, it's unfeasible to annotate all objects of relevance present in-the-wild, thus, current systems are trained only on a small subset. It does not seem straightforward to significantly scale up this figure. Secondly, human perception operates mostly under the open set recognition/detection setting. Humans can detect/track new unseen objects on the fly, typically using a single template, without requiring any "re-training" or "fine-tuning" of their "detection" skills, arguably a consequence of the prior representation learned, an aspect we sought to exploit here too. Finally, important applications in robotics, where agents may interact with previously unseen objects, might require their subsequent detection on the fly without any re-training. Few-Shot Object Detection (FSOD) refers to the problem of detecting a novel class not seen during training and, hence, can potentially address many of the aforementioned challenges.

There are still important *desiderata* that current FSOD system must address in order to be practical and flexible to use: (a) They must be used as is, not requiring *any* re-training (*e.g.* fine-tuning) at test time. However, many existing state-of-the-art FSOD systems (*e.g.* Sun et al. (2021); Wu et al. (2021); Qiao et al. (2021)) rely on re-training with the few available examples of the unseen classes. While such systems are still useful, the requirement for re-training makes them significantly more difficult to deploy on the fly and in real-time or on devices with limited capabilities for training. (b) They must be able to handle an arbitrary number of novel objects (and moreover an arbitrary number of examples per novel class) simultaneously during test time, in a single forward pass without requiring batching. This is akin to how closed systems work, which are able to detect multiple objects concurrently. However, to our knowledge there is no FSOD system possessing this property without requiring re-training. (c) They must attain classification accuracy that is comparable to that of closed

systems. However, existing FSOD systems are far from achieving such high accuracy, especially for difficult datasets like MSCOCO.

This work aims to significantly advance the state-of-the-art in all three above-mentioned challenges. To this end, and building upon the DETR Carion et al. (2020) framework, we propose a system, called Few-Shot DETR (FS-DETR), capable of detecting multiple novel classes at once, supporting a variable number of examples per class, and importantly, without any extra re-training. In our system, the visual template(s) from the new class(es) are used, during test time, in two ways: (1) in FS-DETR's encoder to filter the backbone's image features via cross-attention, and more importantly, (2) as visual prompts in FS-DETR's decoder, "stamped" with special pseudo-class encodings and prepended to the learnable object queries. The pseudo-class encodings are used as pseudo-classes which a classification head attached to the object queries is trained to predict via a Cross-Entropy loss. Finally, the output of the decoder are the predicted pseudo-classes and regressed bounding boxes.

In summary, **our main contributions** are:

1. We propose the very first, to the best of our knowledge, Few-Shot DEtection TRansformer (FS-DETR) capable of detecting multiple novel objects at once, supporting a variable number of samples per class, and without requiring any fine-tuning.
2. We show that all these features can be enabled by extending DETR based on two key ideas: (1) feed the provided visual templates of novel classes as visual prompts during test time, and (2) "stamp" these prompts with pseudo-class embeddings, which are then predicted at the output of the decoder along with bounding boxes.
3. We also propose a simple and efficient yet powerful pipeline consisting of unsupervised pre-training followed by prompt-like base class training.
4. In addition to being more flexible, our system matches and outperforms state-of-the-art results on the standard FSOD setting on PASCAL VOC and MSCOCO. Specifically, FS-DETR outperforms the not re-trained method of Han et al. (2021) (ICCV21) and most re-training based methods on extreme few-shot settings ($k = 1, 2$), while being competitive for more shots.

2 Related work

DEtection TRansformer (DETR) approaches: After revolutionizing NLP Vaswani et al. (2017); Raffel et al. (2019), Transformer-based architectures have started making significant impact in computer vision problems Dosovitskiy et al. (2020); Liu et al. (2021b). In object detection, methods are typically grouped into two-stage (proposal-based) Ren et al. (2015); He et al. (2017); Cai & Vasconcelos (2018) and single-stage (proposal-free)Lin et al. (2017); Liu et al. (2016); Tian et al. (2019b); Zhou et al. (2019); Law & Deng (2018) methods. In this field, a recent breakthrough is the DEtection TRansformer (DETR) Carion et al. (2020), which is a single-stage approach that treats the task as a direct set prediction without requiring hand-crafted components, like non-maximum suppression or anchor generation. Specifically, DETR is trained in an end-to-end manner using a set loss function which performs bipartite matching between the predicted and the ground-truth bounding boxes. Because DETR has slow training convergence, several methods have been proposed to improve it Meng et al. (2021); Zhu et al. (2021b); Dai et al. (2021). Conditional DETR Meng et al. (2021) learns a conditional spatial query from the decoder embeddings that are used in the decoder for cross-attention with the image features. Deformable DETR Zhu et al. (2021b) proposes deformable attention in which attention is performed only over a small set of key sampling points around a reference point. Unsupervised pre-training of DETR Dai et al. (2021) (UP-DETR) improves its convergence, where randomly cropped patches are summed to the object queries and the model is then trained to detect them in the original image.

While our approach is agnostic to the exact variant of DETR, due to its fast training convergence, we opted to use Conditional DETR as the model that we build our FS-DETR approach upon. Beyond this, the above mentioned works are on closed set recognition and while UP-DETR's unsupervised pre-training could be potentially used for few-shot detection, the experimental setting presented in their work doesn't match the standard settings for few-shot detection and no code is provided for its training. We re-implemented UP-DETR Dai et al. (2021) for few-shot detection and found that our method outperforms it. This is expected as their goal is unsupervised pre-training and not FSOD.

Few Shot Object Detection (FSOD) methods can be categorised into *re-training based* and *without re-training* methods. Re-training based methods assume that during test time, but before deployment, the provided samples of the novel categories can be used to fine-tune the model. This setting is restrictive as it requires training before deployment. Instead, without re-training methods can be directly deployed on the fly for the detection of novel examples.

Re-training based approaches can be divided into meta-learning and fine-tuning approaches. Meta-learning based approaches attempt to transfer knowledge from the base classes to the novel classes through meta-learning Finn et al. (2017); Gidaris & Komodakis (2019). Meta R-CNN Yan et al. (2019) introduces a Predictor-head Re-modelling Network which uses examples of novel classes to meta-learn corresponding class-attentive vectors that are used to modulate (via attention) RoI features in order to detect the novel objects. MetaDet Wang et al. (2019) proposes a weight prediction meta-model that can predict the parameters of class-specific layers using the examples of the novel classes. TIP Li & Li (2021) proposes to augment the standard FSOD pipeline with consistency regularization using the predictions obtained by standard image augmentations. FSIW Xiao & Marlet (2020) proposes to tackle both FSOD and few-shot viewpoint estimation through a proposed feature aggregation module and meta-training on a balanced dataset. Fine-tuning based methods follow the standard pre-train and fine-tune pipeline. TFA Wang et al. (2020) proposes fine-tuning the final classification layer of a Faster R-CNN model (first trained on base classes), with a balanced subset containing also the examples of the novel classes. More recently, SRR-FSD Zhu et al. (2021a) proposed to construct a semantic space using word embeddings, and then train a FSOD by projecting and aligning object visual features with their corresponding text embeddings. CME Li et al. (2021) proposes to learn a feature embedding space where the margins between novel classes are maximised. Retentive R-CNN Fan et al. (2021) addresses the problem of learning a FSOD without catastrophic forgetting (*i.e*. without compromising base class accuracy). FSCE Sun et al. (2021) aims to decrease instance similarity between objects belonging to different categories by adding a secondary branch to the primary RoI head, which is trained via supervised contrastive learning. The method of Zhang & Wang (2021) proposes a hallucinator network to generate examples which can help the classifier learn a better decision boundary for the novel classes. FSOD-UP Wu et al. (2021) proposes to construct universal prototypes capturing invariant object characteristics which, via fine-tuning, are adapted to the novel categories. DeFRCN Qiao et al. (2021) is a fine-tuning based method which proposes to perform stop-gradient between the RPN and the backbone, and scale-gradient between RCNN and the backbone. Currently, DeFRCN represents the state-of-the-art of FSOD re-trained methods.

Without re-training approaches are primarily based on metric learning Vinyals et al. (2016); Snell et al. (2017). A standard approach is Hsieh et al. (2019), which uses cross-attention between the backbone's and the query's features to refine the proposal generation, then re-uses the query to re-weight the RoI features channel-wise (in a squeeze-and-excitation manner) for novel class classification. A similar approach for proposal generation is described in Fan et al. (2020), where the squeeze-and-excitation module is replaced with a multi-relation network. Finally, QA-FewDet extends Hsieh et al. (2019); Fan et al. (2020) by modelling class-class, class-proposal and proposal-proposal relationships using various GCNs and, to our knowledge, represents the state-of-the-art FSOD method without re-training. We show that the proposed FS-DETR outperforms it by a large margin.

3 METHOD

Given a dataset where each image is annotated with a set of bounding boxes representing the instantiations of C known base classes, our goal is to train a model capable of localizing objects belonging to novel classes, *i.e*. unseen during training, using up to k examples per novel class. In practice, we partition the available datasets into two disjoint sets, one containing C_{novel} classes for testing, and another with C_{base} classes for training (*i.e*. $C = C_{novel} \cup C_{base}$ and $C_{novel} \cap C_{base} = \emptyset$).

3.1 OVERVIEW OF FS-DETR

We build the proposed Few-Shot DEtection TRansformer (FS-DETR) upon DETR's architecture [1]. FS-DETR's architecture consists of: (1) the CNN backbone used to extract visual features from the

[1]We note that, in practice, due to its superior convergence properties, we used the Conditional DETR as the basis of our implementation but for simplicity of exposition we will use the original DETR architecture.

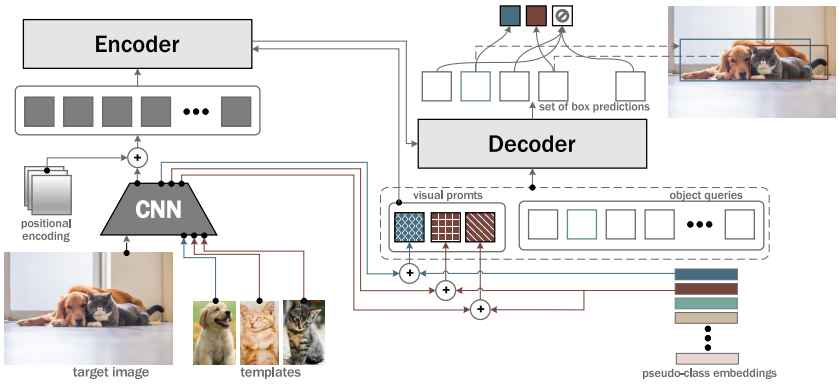

Figure 1: In the proposed FS-DETR, the available templates are provided as additional *visual prompts* to the system in order to condition and control the output. To train and test the system, these prompts are "stamped" with *pseudo-class embeddings* (see Sec. 3.2) which are predicted at the output of the decoder along with bounding boxes (note, that there is *no correlation* between actual classes and pseudo-classes, e.g. the cat could be of either class: "blue" or "red" as there is no preferred order). FS-DETR naturally supports $k-$shot detection, as the model can process multiple examples per class at once. Templates belonging to the same class will share the same pseudo-class embedding. Red and blue colors denote the different pseudo-classes associated to the input templates.

target image and the templates, (2) a transformer encoder that performs self-attention on the image tokens and cross-attention between the templates and the image tokens, and (3) a transformer decoder that processes object queries and templates to make predictions for pseudo-classes (see also below) and bounding boxes. Contrary to the related works of Fan et al. (2020); Han et al. (2021), our system processes an arbitrary number of templates (*i.e.* new classes) jointly, in a single forward pass, *i.e.* without requiring batching, significantly improving the efficiency of the process.

Key contributions: DETR re-formulates object detection as a set prediction problem, making object predictions by "tuning" a set of N learnable queries $\mathbf{O} \in \mathbb{R}^{N \times d}$ to the image features through cross-attention. The queries $\mathbf{O}$ are used as prompts in DETR for closed-set object detection. To accommodate for open-set FSOD, we propose to provide novel classes' templates as additional *visual prompts* to the system in order to condition and control the detector's output. To train the system, we also propose to "stamp" these prompts with *pseudo-class embeddings*, which are then predicted by the decoder along with bounding boxes. The proposed FS-DETR is depicted in Fig. 1. Compared to Carion et al. (2020), we highlight key differences in our mathematical formulation in red.

3.2 FS-DETR

The following subsections detail FS-DETR's architecture and main components.

Template encoding: Let $\mathbf{T}_{i,j} \in \mathbb{R}^{H_p \times W_p \times 3}, i \in \{1, \dots, m\}, j \in \{1, \dots, k\}$ be the template images of the available classes (sampled from C_{base} during training) where m is the number of classes at the current training iteration (m can vary), and k is the number of examples per class (*i.e.* k-shot detection; k can also vary). A CNN backbone (*e.g.* ResNet-50) generates template features $\mathbf{X} = \texttt{CNN}(\mathbf{T}),\ \mathbf{X} \in \mathbb{R}^{mk \times d}$ using either average or attention pooling (see Sec. 5).

Pseudo-class embeddings: We propose to dynamically and randomly associate, at each training iteration, the k templates in $\mathbf{X}$ belonging to the i-th class (for that iteration) with a pseudo-class represented by a pseudo-class embedding $\mathbf{c_i^s} \in \mathbb{R}^d$, which are added to the templates as follows:

$$\mathbf{X}^s = \mathbf{X} + \mathbf{C}^s, \tag{1}$$

where $\mathbf{C}^s \in \mathbb{R}^{mk \times d}$ contains the pseudo-class embeddings for all templates at the current iteration. The pseudo-class embeddings are initialised from a normal distribution and learned during training. They are not determined by the ground-truth categories and are class-agnostic. During each inference step, we arbitrarily associate to a template (belonging to some class) the i-th embedding as described by Eq. 1. The goal is to predict the pseudo-class i. Note that the actual class information is not used. As the assigned embedding changes at every iteration, there is no correlation between the actual classes and the learned embeddings. See also Fig. 1 that exemplifies this process. In the proposed

FS-DETR, each decoded object query $\mathbf{o}_i$ in $\mathbf{O}$ will attempt to predict a pseudo-class using a classifier. Pseudo-class embeddings add a signature to each template allowing the network to track the template within and dissociate it from the rest of the templates belonging to a different class. The pseudo-class embeddings are a key contribution of our approach. The method cannot be trained without the pseudo-class embeddings (*i.e.* it won't converge). As transformers are permutation invariant, it's not possible to predict the pseudo-class without such embeddings.

Templates as visual prompts: We propose to provide the templates $\mathbf{X}^s$ as *visual prompts* to the system by prepending them to the sequence of object queries fed to the decoder:

$$\mathbf{O}' = [\mathbf{X}^s \;\; \mathbf{O}], \;\; \mathbf{O}' \in \mathbb{R}^{(mk+N)\times d}. \tag{2}$$

As shown below, the templates will induce pseudo-class related information into the object queries via attention. This can be interpreted as a new form of training-aware soft-prompting Liu et al. (2021a).

FS-DETR encoder: Given a target image $\mathbf{I} \in \mathbb{R}^{H'\times W'\times 3}$, the same CNN backbone used for template feature extraction first generates image features $\mathbf{Z} = \texttt{CNN}(\mathbf{I}), \;\; \mathbf{Z} \in \mathbb{R}^{S\times d}, \;\; S = H \times W$, which are enriched with positional information through positional encodings $\mathbf{Z} \leftarrow \mathbf{Z} + \mathbf{P}_s, \;\; \mathbf{P}_s \in \mathbb{R}^{S\times d}$. The features $\mathbf{Z}$ are then processed by FS-DETR's encoder layers in order to be enriched with global contextual information. The $l-$th encoding layer processes the output features of the previous layer $\mathbf{Z}^{l-1}$ using a series of Multi-Head Self-Attention (MHSA), Layer Normalization (LN), and MLP layers (typical in Vaswani et al. (2017) and Carion et al. (2020)), as well as a newly proposed Multi-Head Cross-Attention (MHCA) layer as follows [2]:

$$\mathbf{Z}' = \text{MHSA}(\text{LN}(\mathbf{Z}^{l-1})) + \mathbf{Z}^{l-1}, \tag{3}$$
$$\mathbf{Z}'' = \text{MHCA}(\text{LN}(\mathbf{Z}'), \mathbf{X}^s) + \mathbf{Z}', \tag{4}$$
$$\mathbf{Z}^l = \text{MLP}(\text{LN}(\mathbf{Z}'')) + \mathbf{Z}''. \tag{5}$$

The purpose of the MHCA layer above is to filter and highlight early on, before decoding, the image tokens of interest. We have found that such a layer noticeably increases few-shot accuracy (see also Section 5). FS-DETR's encoder is implemented by stacking $L = 6$ blocks, each following Eq. (3)-(5). As image tokens are permutation invariant, we followed Carion et al. (2020) and used a fixed positional encoding. For the templates, pseudo-class embeddings serve as positional encodings.

FS-DETR decoder: FS-DETR's decoder accepts as input the concatenated templates and learnable object queries $\mathbf{O}'$ which are transformed by the decoder's layers through self-attention and cross-attention layers in order to be eventually used for pseudo-class prediction and bounding box regression. The $l-$th decoding layer processes the output features of the previous layer $\mathbf{V}^{l-1}$ as follows:

$$\mathbf{V}' = \text{MHSA}(\text{LN}(\mathbf{V}^{l-1}) + \mathbf{O}') + \mathbf{V}^{l-1}, \tag{6}$$
$$\mathbf{V}'' = \text{MHCA}(\text{LN}(\mathbf{V}') + \mathbf{O}', \mathbf{Z}^l) + \mathbf{V}', \tag{7}$$
$$\mathbf{V}^l = \text{MLP}(\text{LN}(\mathbf{V}'')) + \mathbf{V}'', \tag{8}$$

where $\mathbf{V}^0 = [\mathbf{X}^s \;\; \texttt{zeros}(N, d)]$. Notably, different MLPs are used to process the decoder's features $\mathbf{V} = [\mathbf{V}_{\mathbf{X}^s} \;\; \mathbf{V}_{\mathbf{O}}]$ corresponding to the templates $\mathbf{V}_{\mathbf{X}^s}$ and the object queries $\mathbf{V}_{\mathbf{O}}$:

$$\texttt{MLP}(\mathbf{V}) = [\texttt{MLP}(\mathbf{V}_{\mathbf{X}^s}) \;\; \texttt{MLP}(\mathbf{V}_{\mathbf{O}})]. \tag{9}$$

FS-DETR's decoder consists of $L = 6$ layers implemented using Eqs. (6)-(9).

FS-DETR training and loss functions: For each base class that exists in the target image, we create a template for that class by randomly sampling and cropping an object from that category using a different image (containing an object of the same class) from the train set. After applying image augmentation, the cropped object/template is passed through the CNN backbone of FS-DETR. For each target image and template i (depicted in that image), the ground truth is $y_i = (c_i^s, b_i)$, where c_i^s is the target pseudo-class label (up to m classes in total) and $b_i \in [0, 1]^4$ are the normalised bounding box coordinates. To calculate the loss for training FS-DETR, only the N transformed object queries $\mathbf{V}_{\mathbf{O}}^L \in \mathcal{R}^{N\times d}$ from the output of the last decoding layer are used for pseudo-class classification and bounding box regression (*i.e.* $\mathbf{V}_{\mathbf{X}^s}^L$ is not used). To this end, pseudo-class and

[2] We follow DETR's notation where $\mathbf{O}'$ is added to $LN(\mathbf{V}^{l-1})$ and then projected to form the query Q and key K for self-attention. Here, the first layer $\mathbf{V}^{l-1}$ is initialised as $[X_s, zeros]$ while in DETR with $zeros$.

bounding box prediction heads are used to produce a set of N predictions $\{\hat{y}_i\}_{i=1}^N$ consisting of the pseudo-class probabilities $\hat{p}_i(c^s)$ and the bounding box coordinates $\hat{b}_i$. The heads are implemented using a 3-layer MLP and ReLU activations. Similarly to Carion et al. (2020), we used an additional special pseudo-class $\varnothing$ to denote tokens without valid object predictions. Note that as the training is done in a class-agnostic way via mapping of the base class templates to pseudo-classes (the actual class information is discarded) the model is capable to generalise to the unseen novel categories.

Following Carion et al. (2020), bipartite matching is used to find an optimal permutation $\{\hat{y}_{\sigma i}\}_{i=1}^N$. Finally, the loss is:

$$L = \sum_{i=1}^{N} \lambda_1 L_{\text{CE}}(c_i^s, \hat{p}_{\sigma(i)}(c^s)) + \lambda_2 ||b_i - \hat{b}_{\sigma(i)}||_1 + \lambda_3 \texttt{IoU}(b_i, \hat{b}_{\sigma(i)}), \tag{10}$$

where $\texttt{IoU}$ is the GIoU loss of Rezatofighi et al. (2019) and λ_i are the loss weights.

Pre-training: Transformers are generally *more data hungry* than CNNs due to their lack of inductive bias Dosovitskiy et al. (2020). Therefore, building representations that generalise well to unseen data, and prevent overfitting within the DETR framework, requires larger amounts of data. To this end, we used images from ImageNet-100 Tian et al. (2019a) and to some extent MSCOCO, for unsupervised pre-training where the classes and the bounding boxes are generated on-the-fly using an object proposal system, *without using any labels*. Our pre-training is detailed in the appendix.

4 EXPERIMENTS

Datasets: Experiments presented in this work were all conducted using PASCAL VOC Everingham et al. (2010; 2015) and MSCOCO Lin et al. (2014) datasets. Moreover, ImageNet100 Tian et al. (2019a), consisting of ∼125K images and 100 categories, is used (without labels) to pre-train our object detector. PASCAL VOC and MSCOCO are used to train and evaluate few-shot experiments. Following previous works Kang et al. (2019); Wang et al. (2020); Han et al. (2021), we evaluate the proposed method on PASCAL VOC 2007+2012 and MSCOCO 2014, using the same data splits provided by Kang et al. (2019); Wang et al. (2020). Specifically, PASCAL VOC is randomly divided into three different splits, each consisting of 15 base and 5 novel classes; as is common practice, training is done on the PASCAL VOC 2007+2012 train/val sets, and evaluation on the PASCAL VOC 2007 test set. Similarly, MSCOCO is split into base and novel categories, where the 20 overlapping categories with PASCAL VOC are considered novel, while the remainder are the base categories; following recent convention Kang et al. (2019); Wang et al. (2020); Han et al. (2021), 5k samples from the validation set are held out for testing, while the remaining samples from both train and validation sets are used for training.

Evaluation setting: There are currently two widely-used FSOD evaluation protocols. The first focuses exclusively on novel classes while disregarding base class performance, thus not monitoring base class catastrophic forgetting. The second, more comprehensive protocol, often called generalised few-shot object detection (G-FSOD), considers both base and novel classes. The choice of protocol and, hence, results interpretation, bears special importance for re-training based methods, as generalizability to base classes might be compromised. Without re-training methods, as FS-DETR, adhere to the second protocol (G-FSOD) by default, as base class catastrophic forgetting is not applicable.

Baselines: Existing FSOD methods can be broadly categorised into: re-training based, and without re-training. The latter can handle few-shot detection on the fly at deployment, while re-training based FSOD methods generally tend to perform better. Re-training based methods can be further subdivided into "meta-learning" and "fine-tuning" approaches. *"Re-training based: meta-learning"* approaches include: FSRW Kang et al. (2019), MetaDet Wang et al. (2019), Meta R-CNN Yan et al. (2019), Xiao *et al*. Xiao & Marlet (2020), DCNET Hu et al. (2021), TIP Li & Li (2021) and QA-FewDet Han et al. (2021). *"Re-training based: fine-tuning"* methods include: TFA Wang et al. (2020), MPSR Wu et al. (2020), Fan *et al*. Fan et al. (2020), CME Li et al. (2021), SRR-FSD Zhu et al. (2021a), Zhang *et al*. Zhang & Wang (2021) and DeFRCN Qiao et al. (2021), *"Without re-training"* methods include: UP-DETR Dai et al. (2021), Fan *et al*. Fan et al. (2020) and QA-FewDet Han et al. (2021). Note that these two last methods can also be re-trained, offering improved accuracy.

Table 1: FSOD performance (AP50) on the PASCAL VOC dataset. Results with † are from Han et al. (2021) while those with ‡ were produced by us. Results with * disregard performance on base classes Qiao et al. (2021). Our approach outperforms all without re-training methods. Moreover, it provides competitive results compared with other re-training based methods for $k = 3, 5, 10$, and even outperforms them for $k = 1, 2$, *i.e.* extreme few-shot settings.

Method / Shot	Venue	Backbone	Novel Set 1					Novel Set 2					Novel Set 3				
			1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
Re-training based methods (meta-learning or fine-tuning)																	
FSRW* Kang et al. (2019)	ICCV'19	YOLOv2	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
MetaDet* Wang et al. (2019)	ICCV'19	VGG16	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
Meta R-CNN* Yan et al. (2019)	ICCV'19	RN-101	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
TFA w/cos Wang et al. (2020)	ICML'20	RN-101	25.3	36.4	42.1	47.9	52.8	18.3	27.5	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6
TFA w/cos* Wang et al. (2020)	ICML'20	RN-101	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
Xiao *et al*. Xiao & Marlet (2020)	ECCV'20	RN-101	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
MPSR* Wu et al. (2020)	ECCV'20	RN-101	41.7	42.5	51.4	55.2	61.8	24.4	29.3	39.2	39.9	47.8	35.6	41.8	42.3	48.0	49.7
Fan *et al.*† Fan et al. (2020)	CVPR'20	RN-101	37.8	43.6	51.6	56.5	58.6	22.5	30.6	40.7	43.1	47.6	31.0	37.9	43.7	51.3	49.8
DCNET Hu et al. (2021)	CVPR'21	RN-101	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7
TIP Li & Li (2021)	CVPR'21	RN-101	27.7	36.5	43.3	50.2	59.6	22.7	30.1	33.8	40.9	46.9	21.7	30.6	38.1	44.5	50.9
CME Li et al. (2021)	CVPR'21	RN-101	41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5
SRR-FSD Zhu et al. (2021a)	CVPR'21	RN-101	47.8	50.5	51.3	55.2	56.8	32.5	35.3	39.1	40.8	43.8	40.1	41.5	44.3	46.9	46.4
Zhang *et al.** Zhang & Wang (2021)	CVPR'21	RN-101	47.0	44.9	46.5	54.7	54.7	26.3	31.8	37.4	37.4	41.2	40.4	42.1	43.3	51.4	49.6
QA-FewDet Han et al. (2021)	ICCV'21	RN-101	42.4	51.9	55.7	62.6	63.4	25.9	37.8	46.6	48.9	51.1	35.2	42.9	47.8	54.8	53.5
DeFRCN Qiao et al. (2021)	ICCV'21	RN-101	40.2	53.6	58.2	63.6	66.5	29.5	39.7	43.4	48.1	52.8	35.0	38.3	52.9	57.7	60.8
DeFRCN* Qiao et al. (2021)	ICCV'21	RN-101	53.6	57.5	61.5	64.1	60.8	30.1	38.1	47.0	53.3	47.9	48.4	50.9	52.3	54.9	57.4
Without re-training methods																	
Fan *et al.*† Fan et al. (2020)	CVPR'20	RN-101	32.4	22.1	23.1	31.7	35.7	14.8	18.1	24.4	18.6	19.5	25.8	20.9	23.9	27.8	29.0
UP-DETR‡ Dai et al. (2021)	ICCV'21	DETR-R50	38.2	40.4	44.5	45.8	46.0	20.0	23.6	25.8	28.0	33.9	34.1	35.3	37.0	40.1	40.3
QA-FewDet Han et al. (2021)	ICCV'21	RN-101	41.0	33.2	35.3	47.5	52.0	23.5	29.4	37.9	35.9	37.1	33.2	29.4	37.6	39.8	41.5
FS-DETR (Ours)	this work	DETR-R50	**45.0**	**48.5**	**51.5**	**52.7**	**56.1**	**37.3**	**41.3**	**43.4**	**46.6**	**49.0**	**43.8**	**47.1**	**50.6**	**52.1**	**56.9**

4.1 RESULTS ON PASCAL VOC

Table 1 summarises our results and compares them with the current state-of-the-art on PASCAL VOC. Experiments for k-shot detection were conducted for three data splits, where k was set to $1, 2, 3, 4, 5, 10$ and AP50 values are reported. Note that Table 1 is split into two sections: Methods at the top require an additional few-shot re-training stage, while those at the bottom, including our method, do not require any re-training. Here, it can be appreciated that our approach outperforms all without re-training methods by a large margin, improving the current state-of-the-art Han et al. (2021); Dai et al. (2021) in any shot and all split experiments by up to 17.8 AP50 points, and, in most cases, by at least $\sim$10 AP50 points. Moreover, and contrary to Han et al. (2021), our method can process multiple novel classes in a single forward pass. Finally, we re-implemented UP-DETR Dai et al. (2021) for few-shot detection on PASCAL VOC (since there is no publicly available implementation for few-shot detection or results). Our method largely outperforms it, perhaps unsurprisingly, as the latter was not developed for few-shot detection, but for unsupervised pre-training.

Importantly, the proposed method provides competitive results or even outperforms re-training based methods (meta-learned or fine-tuned). Specifically for $k = 3, 5, 10$, our method provides accuracy which is on par with that of the most accurate methods (*e.g.* Li & Li (2021); Li et al. (2021); Zhu et al. (2021a); Qiao et al. (2021)). However, for $k = 1, 2$, *i.e.* on extreme few-shot settings, our method outperforms all re-training based methods but DeFRCN* Qiao et al. (2021). However, DeFRCN* is fine-tuned to optimise novel class performance only. The same method, DeFRCN, when optimised for both base and novel classes (G-FSOD setting), achieves a more muted performance and still below FS-DETR (ours) for extreme few-shot settings. Qualitative visualizations in appendix.

4.2 RESULTS ON MSCOCO

Table 2 shows evaluation results for FS-DETR and all competing state-of-the-art methods on MSCOCO. Similarly to above, Table 2 is split into methods requiring re-training at the top and those that do not require re-training at the bottom. There, it can be appreciated that FS-DETR outperforms all comparable state-of-the-art methods Han et al. (2021); Fan et al. (2020) by up to 3.1 AP50 points (1-shot) and, in most cases, by at least $\sim$ 1.1 AP50 points. In our experiments UP-DETR failed to converge on MSCOCO, hence, results are not included Table 2. We speculate that this might be due to UP-DETR's partitioning the input queries by the number query patches, therefore, limiting the number of tokens query patches interact with. This appears to be too restrictive for MSCOCO. Moreover, and in line with results observed on PASCAL VOC, FS-DETR achieves competitive

results to those of re-trained based methods on MSCOCO, a far more challenging dataset. FS-DETR outperforms all re-training based methods for $k = 1, 2, 3$, with the exception of DeFRCN* Qiao et al. (2021), while performing comparably for $k = 5, 10$. Similar to observations from Sec. 4.1, DeFRCN optimised for both base and novel classes performs below FS-DETR for extreme few-shot settings.

Table 2: FSOD performance on the MSCOCO dataset. Results with † are from Han et al. (2021). Results with * disregard performance on base classes Qiao et al. (2021). Our method consistently outperforms the state-of-the-art methods in most of the shots and metrics.

Method	1-shot			2-shot			3-shot			5-shot			10-shot		
	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
Re-trained methods (meta-learning or fine-tuning)															
FSRW* Kang et al. (2019)	–	–	–	–	–	–	–	–	–	–	–	–	5.6	12.3	4.6
MetaDet* Wang et al. (2019)	–	–	–	–	–	–	–	–	–	–	–	–	7.1	14.6	6.1
Meta R-CNN* Yan et al. (2019)	–	–	–	–	–	–	–	–	–	–	–	–	8.7	19.1	6.6
TFA w/cos Wang et al. (2020)	1.9	3.8	1.7	3.9	7.8	3.6	5.1	9.9	4.8	7.0	13.3	6.5	9.1	17.1	8.8
TFA w/cos*† Wang et al. (2020)	3.4	5.8	3.8	4.6	8.3	4.8	6.6	12.1	6.5	8.3	15.3	8.0	10.0	19.1	9.3
Xiao *et al.*† Xiao & Marlet (2020)	3.2	8.9	1.4	4.9	13.3	2.3	6.7	18.6	2.9	8.1	20.1	4.4	10.7	25.6	6.5
MPSR*† Wu et al. (2020)	2.3	4.1	2.3	3.5	6.3	3.4	5.2	9.5	5.1	6.7	12.6	6.4	9.8	17.9	9.7
Fan *et al.*† Fan et al. (2020)	4.2	9.1	3.0	5.6	14.0	3.9	6.6	15.9	4.9	8.0	18.5	6.3	9.6	20.7	7.7
DCNET Hu et al. (2021)	-	-	-	-	-	-	-	-	-	-	-	-	12.8	23.4	11.2
TIP Li & Li (2021)	-	-	-	-	-	-	-	-	-	-	-	-	16.3	33.2	14.1
CME Hu et al. (2021)	-	-	-	-	-	-	-	-	-	-	-	-	15.1	24.6	16.4
SRR-FSD Zhu et al. (2021a)	-	-	-	-	-	-	-	-	-	-	-	-	11.3	23.0	9.8
Zhang *et al*. Zhang & Wang (2021)	4.4	7.5	4.9	5.6	9.9	5.9	7.2	13.3	7.4	-	-	-	-	-	-
QA-FewDet Han et al. (2021)	4.9	10.3	4.4	7.6	16.1	6.2	8.4	18.0	7.3	9.7	20.3	8.6	11.6	23.9	9.8
DeFRCN Qiao et al. (2021)	4.8	-	-	8.5	-	-	10.7	-	-	13.6	-	-	16.8	-	-
DeFRCN* Qiao et al. (2021)	9.3	-	-	12.9	-	-	14.8	-	-	16.1	-	-	18.5	-	-
Methods without re-training															
Fan *et al.*† Fan et al. (2020)	4.0	8.5	3.5	5.4	11.6	4.6	5.9	12.5	5.0	6.9	14.3	6.0	7.6	15.4	6.8
QA-FewDet Han et al. (2021)	5.1	10.5	4.5	7.8	16.4	6.6	8.6	17.7	7.5	9.5	19.3	8.5	10.2	20.4	9.0
FS-DETR (Ours)	**7.0**	**13.6**	**7.5**	**8.9**	**17.5**	**9.0**	**9.8**	**18.5**	**9.8**	**10.7**	**20.5**	**10.8**	**11.1**	**21.6**	**11.0**

Table 3: FSOD performance (AP50) on the PASCAL VOC dataset Novel Set 1 for various template construction configurations. † - result produced using bounding-box jittering for the patch extraction.

Resolution	Pool. type	Novel Set 1				
		1	2	3	5	10
128	global.avg.	42.9	46.0	49.4	50.5	54.0
128	attn.	45.0	48.5	51.5	52.7	56.1
128†	attn.	39.0	42.8	44.6	46.4	50.3
96	attn.	43.2	45.7	49.0	50.1	52.9
192	attn.	45.1	48.3	51.0	52.9	57.0

5 Ablation studies

Herein, we ablate different variations and components of our method, analysing the impact of different design choices. Unless otherwise specified, we report results on Novel Set 1 on PASCAL VOC. For more details and discussions see appendix.

Design of template encoder: An important component of our system is the extraction of discriminative prompts from the novel classes' templates. To this end, we re-use FS-DETR's input image CNN encoder. However, to focus on the most important components we used attention-based pooling instead of simple global average pooling. In Table 3 we report the impact of: (a) resolution, (b) augmentation level, and (c) pooling type. As the results show, increasing the resolution from 128 to 192px yields no additional gains. This suggests that, at least for the datasets in question, fine grained details are not quintessential for the identification of the targeted novel class and higher level concepts suffice. While spatial augmentation generally helps (*i.e*. for object recognition), we found that adding noise to the ground truth bounding box of the template at train time leads to lower accuracy. This makes the problem for the object detector too hard, and impedes convergence. Finally, attentive pooling can further boost the performance compared with a simpler global average pooling.

Pre-training: Many FSOD systems use pre-trained backbones on ImageNet for classification. Deviating from this, we pre-train our system in an unsupervised manner on ImageNet images and parts of MSCOCO without using the labels. We note that this is especially important for transformer based architectures which were shown to be more prone to over-fitting due to the lack of inductive bias Dosovitskiy et al. (2020). As the results from Table 4 show, unsupervised pre-training, can significantly boost the performance, preventing over-fitting toward the base classes and improving overall discriminative capacity. To reduce over-fitting the pre-training loss on ImageNet data is applied during supervised training every 8th iteration. Additional details and experiments in appendix.

Table 4: Few-shot object detection performance (AP50) on the PASCAL VOC dataset on the Novel Set 1 for models with and without pre-training.

Pre-training	Novel Set 1				
	1	2	3	5	10
	19.0	21.1	23.3	24.0	24.6
✓	45.0	48.5	51.5	52.7	56.1

Auxiliary losses: We explored the impact of using additional auxiliary losses applied to the object queries, an L_2 feature loss and a contrastive loss, where the positive pairs are formed by taking the input templates with all the object query tokens assigned to it by the Hungarian assignment algorithm. We did not observe any further gains from the additional losses, suggesting that the pseudo-classification loss alone suffices for guiding the network.

Impact on individual components: Herein, we analyse the accuracy improvement obtained by two components of FS-DETR namely the MHCA layer in FS-DETR's encoder (see Eq. 5), and the type-specific MLPs (TS-MLP) in FS-DETR's decoder (see Eq. 9). As Table 5 shows, while our system, without both components, provides satisfactory results, unsurprisingly, the addition of TS-MLP further boosts the accuracy. This is expected as the information carried by the object queries and template tokens is semantically different, so ideally they should be transformed using different functions. Finally, the MHCA within the encoder injects template-related information early on to filter or highlight certain areas of the image, and also helps increase the accuracy.

Table 5: Impact of various components on the few-shot object detection performance (AP50) on the PASCAL VOC dataset (Novel Set 1).

Approach	Novel Set 1				
	1	2	3	5	10
FS-DETR w/o TS-MLP	42.2	46.9	48.3	49.2	51.6
FS-DETR w/o MHCA of Eq. 4	38.1	40.6	41.7	42.2	45.6
FS-DETR	45.0	48.5	51.5	52.7	56.1

6 CONCLUSIONS

In this work we propose FS-DETR, a novel transformer based few-shot architecture, that is simple yet powerful, while also being very flexible and easy to train. FS-DETR outperforms all previously proposed methods, thus achieving a new state-of-the-art. In addition to the outstanding results and discussions presented in Sec. 4, the proposed method can simultaneously predict arbitrary number of classes, using variable-shots per class, in a single forward pass. These results, in combinations with the methods formulation, clearly demonstrate not only its performance improvements but also its high flexibility. Therefore, FS-DETR can uniquely satisfy the outlined FSOD system *desiderata* (a) and (b), while at the same time making big improvements toward satisfying (c).

ETHICS STATEMENT

Due to the vast computational resources required, training NNs is energy intensive, hence, potentially detrimental to the environment. As our approach requires no re-training for novel classes, the power

and compute requirements of adjusting to new classes is reduced. Due to its algorithmic nature, we foresee no direct negative applications for our approach. However, similarly to most data-driven systems, bias from the training data can potentially affect the fairness of the model. As such, we suggest to take this aspect into consideration when deploying the models into real-world scenarios. For a more detailed description of the impact and limitations please refer to the appendix.

REPRODUCIBILITY STATEMENT

The basis of the implementation used for all experiments conducted and detailed in this work make use the Conditional DETR Meng et al. (2021) library (which itself is based on the DETR Carion et al. (2020) library). Key algorithmic differences are detailed and highlighted in Section 3. Additional implementation details, including data preprocessing steps as well as pre-training and training hyper-parameters, are shared in Appendix A and B. To further assist efficient reproduciblily of our work, the complete code base (training and inference) will be made publicly available.

REFERENCES

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020.

Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-DETR: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Mark Everingham, S. M. Eslami, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-RPN and multi-relation detector. In *CVPR*, 2020.

Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *CVPR*, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Spyros Gidaris and Nikos Komodakis. Generating classification weights with GNN denoising autoencoders for few-shot learning. In *CVPR*, 2019.

Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *ICCV*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *NuerIPS*, 2019.

Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *CVPR*, 2021.

Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, 2019.

Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, 2018.

Aoxue Li and Zhenguo Li. Transformation invariant few-shot object detection. In *CVPR*, 2021.

Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *CVPR*, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *ECCV*, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021a.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional DETR for fast training convergence. In *ICCV*, 2021.

Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1): 145–151, 1999.

Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. DeFRCN: Decoupled Faster R-CNN for few-shot object detection. In *ICCV*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *NeurIPS*, 2015.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 2017.

Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. FSCE: Few-shot object detection via contrastive proposal encoding. In *CVPR*, 2021.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019a.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *ICCV*, 2019b.

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *NeurIPS*, 29, 2016.

Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML*, 2020.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *ICCV*, 2019.

Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Universal-prototype enhancing for few-shot object detection. In *ICCV*, 2021.

Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *ECCV*, 2020.

Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 2020.

Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta R-CNN: Towards general solver for instance-level low-shot learning. In *ICCV*, 2019.

Weilin Zhang and Yu-Xiong Wang. Hallucination improves few-shot object detection. In *CVPR*, 2021.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

Chenchen Zhu, Fangyi Chen, Uzair Ahmed, Zhiqiang Shen, and Marios Savvides. Semantic relation reasoning for shot-stable few-shot object detection. In *CVPR*, 2021a.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021b.

A IMPLEMENTATIONS DETAILS

FS-DETR extends Conditional DETR Meng et al. (2021) (see Section 3), and was pre-trained and trained on a single node with 8 P40 GPUs. Following Dai et al. (2021), the ResNet50 He et al. (2016) backbone is initialized from SwAV Caron et al. (2020) and kept frozen. Pre-training makes use of ImageNet-100 without labels, with object proposals detection as a pretext task.

Pre-training hyper-parameters were set to: Batch size of 32 per GPU, AdamW optimizer Loshchilov & Hutter (2017) with a learning rate of 10^{-4}, frozen backbone CNN, path dropout of 0.1, training for 60 epochs with the learning rate decreased by factor of 10 after 40 epochs. When using larger images for pre-training (*i.e*. containing complex scenes) the batch size is decreased to 2.

Training hyper-parameters were set to: Batch size of 2 per GPU, SGD with momentum (0.9) Qian (1999) with the learning rate initially set to $5e^{-1}$, path dropout of 0.1, training for 30 epochs with the learning rate decreased by a factor of 10 after 20 epochs (and respectively 15 for COCO). Augmentation followed DETR: input images were resized such that the short axis is 480 at least and 800 pixels at most, and the long side is, at most, 1333 pixels, and randomly cropped with 0.5 probability.

Patch augmentation hyper-parameters. The templates are cropped tightly based on the bounding box and then rescaled to a 128×128px image. During training we apply the following augmentations: color jittering, with 0.8 probability and 0.4 intensity, random gray scale (0.2 probability) and Gaussian blur with a probability of 0.5.

B PRE-TRAINING PROCESS

Transformer based architectures are known to generally be more data-hungry than their homologous CNNs Dosovitskiy et al. (2020); Carion et al. (2020). To alleviate this, we introduce a label-free pre-training step that closely mimics the training stage.

More specifically, at train time, for any given input image, we crop a set of patches according to the object proposals produced by Selective Search Uijlings et al. (2013) [3]. Each of these patches represents an object (belonging to some class) and can be mapped to a pseudo-class, by associating it to a different pseudo-class embedding. Note, that random patches can be used too, but the former leads to faster convergence. The goal of the network is to predict the location of these patches (*i.e.* object templates). To make the task harder, the patches (templates) are augmented using a set of random transformations before being passed to the backbone. Finally, the network is trained using a regression (for the bounding boxes) and a classification loss. As opposed to the supervised training stage, the classification loss is reduced to a binary classification problem: object/no object. The model is then trained using the hyper-parameters described in Section A while the ResNet based backone is initialised from a model pre-trained on Imagenet without supervision (SwAV Caron et al. (2020)). The process is illustrated in Fig. 2.

Pre-training dataset For our DETR pre-training, we used the images belonging to the base classes from COCO (60 classes in total) and ImageNet-100 (a subset of ImageNet introduced in Tian et al. (2019a)). We note the following: firstly, there is no overlap between COCO base classes and VOC and COCO novel classes. Secondly, ImageNet-100 contains classes that can be matched to 7 out of 20 VOC classes (bird, cat, dog, boat, car, motorcycle and chair). Specifically, split-1 of VOC novel classes contains 2/5 classes (bird and motorbike) that overlap with ImageNet-100, split-2 0/5 and split-3 3/5 (boat, cat and motorbike). Please note that NONE of the labels in ImageNet-100 (or COCO) are used at any stage of the pre-training. While we agree that the underlying data distribution, even for unsupervised learning is important, judging from the results from Tables 1 and 2 the gains in absolute terms offered by our approach are consistent across all 3 sets (note that split-2 has no overlap at all).

We note that, recent state-of-the-art methods (*e.g.* Fan et al Fan et al. (2020), QA-FewShot Han et al. (2021), DeFRCN Qiao et al. (2021)) make use of a backbone pre-trained with full supervision on the entire ImageNet, same which includes all VOC/COCO novel classes. In this regard, we trained FS-DETR initialized from a backbone pre-trained on the entirety of Imagenet for classification using full supervision (*e.g.* same as Fan et al. (2020); Han et al. (2021); Qiao et al. (2021)). Preliminary results shown in Tab. 6 (which could likely be improved from hyper-parameter optimization) indicate an overall improvement of approx. 1.5%. This highlights that the pre-training data used in the proposed work doesn't offer any advantage over prior art approaches that use fully supervised pre-trained backbones. Further to this, DeFRCN Qiao et al. (2021) experimented with using a backbone pre-trained on ImageNet without labels (SwAV weights - same as ours) which resulted in substantially degraded performance of approx. 5.0%.

Table 6: Impact of different initialisation of backbone on the PASCAL VOC dataset (Novel Set 1).

Approach	Novel Set 1				
	1	2	3	5	10
FS-DETR (Swav)	45.0	48.5	51.5	52.7	56.1
FS-DETR (ImageNet)	47.1	49.9	52.5	53.8	57.0

[3] Selective Search is a **training-free** generic region proposal algorithm that computes a hierarchical grouping of image regions based on color, texture, size and shape, and hence, has no notion of object classes.

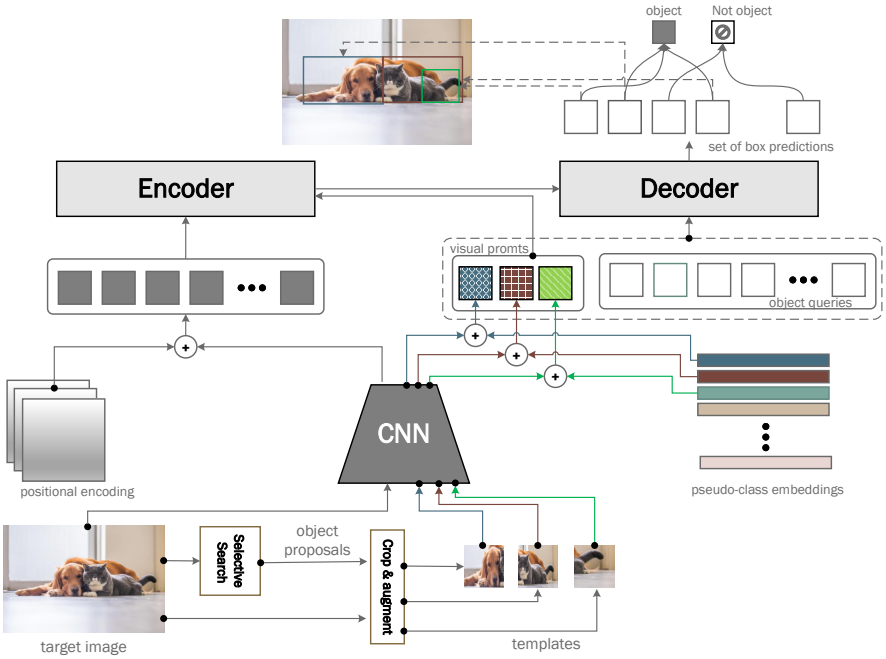

Figure 2: FS-DETR pre-training stage. The pre-training process largely mimics the training stage, with a few notable differences: (1) no annotations are used, (2) the target bounding boxes are proposed by selective search or sampled randomly, (3) the templates are sampled from the target image itself and (4) only two classes are defined - object and no object.

Figure 3: Novel class 1-shot detection examples with FS-DETR. First three columns depict success cases, while the right-most column failures. Green and red boxes indicate novel and base classes, respectively. Note that in the top-left image two novel classes are detected simultaneously.

C QUALITATIVE EVALUATION

Fig. 3 shows 1-shot detection examples of FS-DETR, with success cases shown on the first three columns, and fail cases on the right-most column. The image on top-left of the figure, illustrates an important and unique property of FS-DETR: Two novel classes coexist in a single image, and FS-DETR is able to successfully detect both of them at the same time.

Fig. 4 shows the effect of varying the 1-shot template used during novel class detection. There, smaller images refer to the templates used for 1-shot detection on the paired larger image. From the left-most two pairs of columns, it can be appreciated that even under large template visual variability, FS-DETR proves to be extremely robust, with detections hardly affected by the template change. The right-most illustrates a failure case, where the sofa fails to be detected.

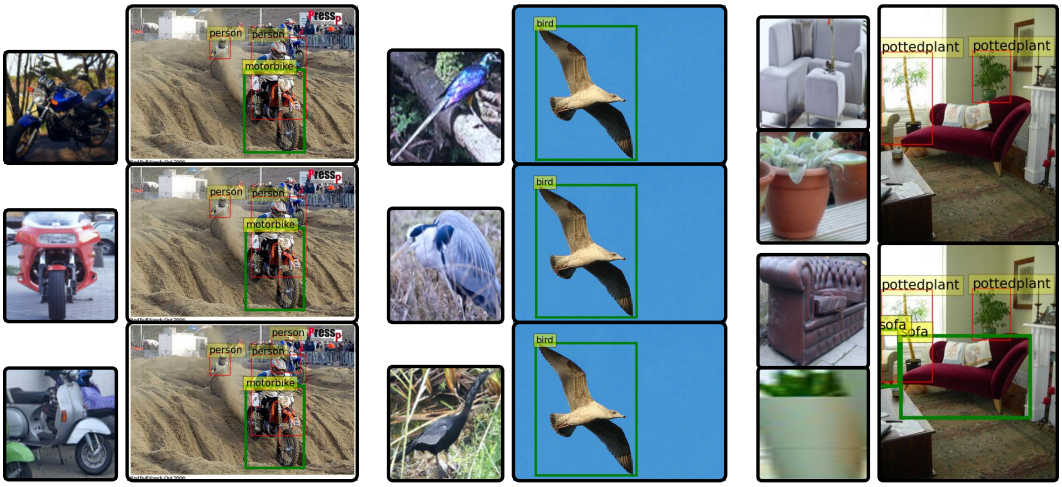

Figure 4: Effect of different 1-shot template on detection with FS-DETR. Small images indicate the template used to detect the objects on the larger images. The left-most two pairs of columns illustrate the robustness to template change, while the right-most column pair illustrates a failure case.

Additionally, in Fig. 5 we visualise the attention weights between the visual prompts and the encoded image features. Notice that our network learns to attend to parts of the target image that are semantically similar to the provided templates that are present in the target image.

D Discussion, challenges and limitations

Herein we offer a pertinent discussion on some things we tried but didn't work, defining some of the limitations and challenges that arise within the proposed framework and more so in general for FSOD using images within DETR framework.

D.1 Few-shot object detection objective ambiguity

A general limitation of few shot object recognition systems, trained and/or tested using one or more visual examples is the ill-definess of what represents a class. For example, presenting a template depicting a dog could require identifying the class "dog", "bulldog"(*i.e*. find dogs of a given bred), "a white dog" etc. While as the number of examples increases the ambiguity decreases, the problem is not fully solvable within the visual domain. A natural solution to this problem could be provided by constructing the templates using natural language. While an interesting solution, this goes beyond the scope of this work.

That being said, to some extent, our approach alleviates parts of this problem: As our model has to distinguish locally within the set of provided positive (present in the image) and negative (not present) templates, it can use them to semantically ground the notion of a class, effectively defining the semantic hierarchy. For example, if all templates are representing different apple varieties, the model is expected to differentiate between these varieties instead of detecting *any* apple.

D.2 Challenges within the DETR framework

Despite it's remarkable success and appealing formulation that removes the need of an explicit object proposal component or post-processing step (*i.e*. NMS), in the context of few-shot detection some of this advantages pose additional challenges, some of which we detail bellow. We believe this aspects could represent potentially interesting future exploration directions.

Semantic misalignment Traditional object detection systems, such as Ren et al. (2015); Redmon et al. (2016); He et al. (2017) preserve an exact feature alignment between the regressed bounding box and the semantic information (*i.e*. the ROI pooling extracts features at the location given by the proposal). DETR derived approaches however construct their representation gradually by adapting a set of object queries via self-attention and cross-attention with the encoded features. As each object

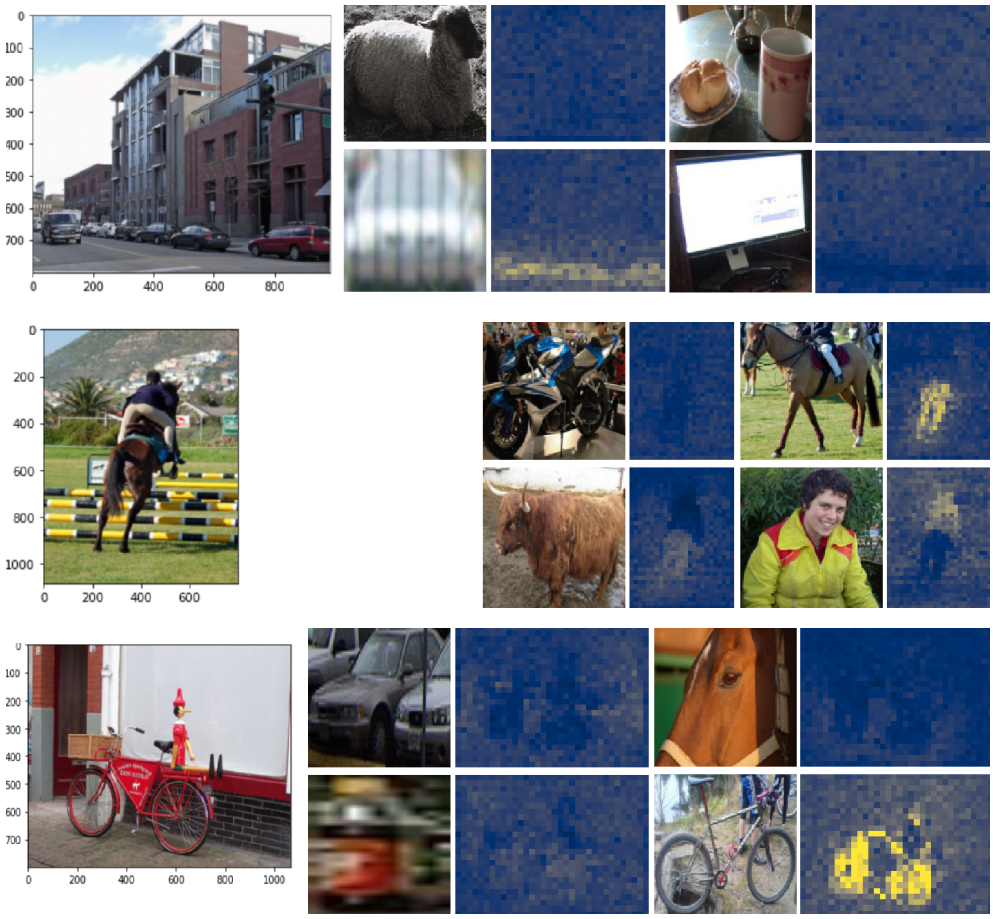

Figure 5: Attention weights between the visual prompts (templates) and the encoded image features for three randomly sampled target images (left column) from VOC Pascal dataset. Notice that the network learns to attend to the parts of the image that are semantically close to the presented templates. For each target image (left column), we show the attention weights generated by four templates. We observe that for the target image of the first row, only the car template generates attention of high magnitude at several locations corresponding to the location of the cars in the target image. Similarly, for the target image of the second row only the horse and the person templates fire at the corresponding locations in the target image as expected. Similar conclusions can be drawn for the target image of the last row.

query operates (attends) to the entire image, as opposed to the local ROI, the query can encode information outside of the predicted bounding box. Thus, we can get to cases where the class may be correct although the bounding box contains mostly objects of an incorrect category.

Therefore, when we tried to use an external supervised classifier, applied to the image region cropped based on the predicted bounding box, surprisingly we noticed a deterioration of the performance. Upon visual inspection we observed a manifestation of the above mentioned phenomena, where the model was able to predict the correct class despite the fact that the predicted bounding box contained predominantly content of a different class, while the external supervised classifier was unable to.

Reduced proposal diversity A key characteristic of DETR systems is the removal of an a) external object proposal generator and b) implicit Non Maximum Suppression (NMS). Upon close inspection of our system we noticed that as we advance within the transformer based decoder, the bounding boxes are pruned via self-attention. By the end, despite having 100-300 object queries, most will point to a very small set of distinct regions of the image, lacking the diversity present in more traditional systems, such as in Fast RCNN. The consequence of this is a higher likelihood of missing unseen

classes in limited data scenarios, making the pre-training even more so important to train the built-in object proposals system.