2021 IEEE/CVF International Conference on Computer Vision (ICCV) | 978-1-6654-2812-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCV48922.2021.00473

Morphable Detector for Object Detection on Demand

Xiangyun Zhao[1], Xu Zou[2], and Ying Wu[1]

[1]Northwestern University
[2]Huazhong University of Science and Technology

Abstract

Many emerging applications of intelligent robots need to explore and understand new environments, where it is desirable to detect objects of novel classes on the fly with minimum online efforts. This is an object detection on demand (ODOD) task. It is challenging, because it is impossible to annotate a large number of data on the fly, and the embedded systems are usually unable to perform back-propagation which is essential for training. Most existing few-shot detection methods are confronted here as they need extra training. We propose a novel morphable detector (MD), that simply "morphs" some of its changeable parameters online estimated from the few samples, so as to detect novel classes without any extra training. The MD has two sets of parameters, one for the feature embedding and the other for class representation (called "prototypes"). Each class is associated with a hidden prototype to be learned by integrating the visual and semantic embeddings. The learning of the MD is based on the alternate learning of the feature embedding and the prototypes in an EM-like approach which allows the recovery of an unknown prototype from a few samples of a novel class. Once an MD is learned, it is able to use a few samples of a novel class to directly compute its prototype to fulfill the online morphing process. We have shown the superiority of the MD in Pascal [12], COCO [27] and FSOD [13] datasets.

1. Introduction

In applications, like robotics exploration and autonomous driving, the systems need to explore and understand new environments, where it is desirable to detect objects of novel classes on the fly with minimum online human supervision and interaction. This is an object detection on demand (ODOD) task. ODOD is very challenging because it is impossible to collect a large amount of data on the fly, and the computing resources are generally not powerful enough for computationally intensive and time-consuming training on board, not to mention that many embedded systems are unable to perform back-propagation which is essential for training. In embedded systems, the detection task is usually carried out on a computationally limited platform where the neural networks are locked after the system is built due to resource limits [19]. The prevailing few-shot detection (FSD) [20, 49, 47, 46, 48] methods are confronted here as they generally need to perform extra training for objects from novel classes.

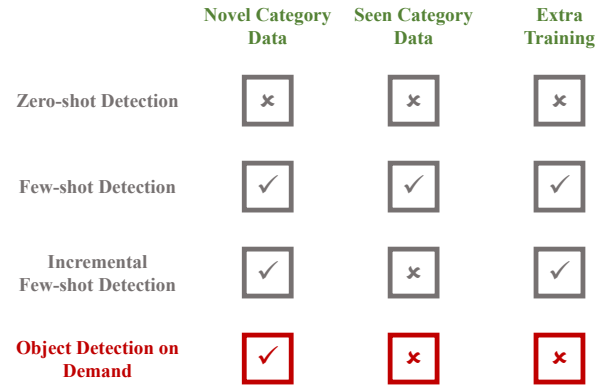

Figure 1: Comparison with other detection tasks. Different from other few-shot detection tasks, Object Detection on Demand requires no extra training.

To this end, we define Object Detection on Demand (ODOD) specifically as detecting the novel classes without extra training while preserving the existing knowledge, given (1) a detector offline trained using base class data, (2) no access to base class data (3) a few samples for novel classes. The ODOD can be regarded as a special few-shot detection task and the differences of ODOD from other detection tasks are listed in Fig 1 .

The prevailing few-shot detectors (FSD) aim to train a detector using base class data and further train it with a few samples from novel classes. However, extra training is unfeasible in the ODOD task. Furthermore, to keep the performance for the base classes, these FSDs have to use

978-1-6654-2812-5/21/$31.00 ©2021 IEEE
DOI 10.1109/ICCV48922.2021.00473

the base class data in extra training, otherwise, they suffer from catastrophic forgetting [29] - a significant performance degradation, when the past data are not available. Other few-shot detectors [13, 18] use a siamese network and take "query-target" pairs as input so as to detect all instances of the "target" object appearing in the "query" image. However, as the target representation is always changing during the training, the model learns less discriminative representations. As a result, the model's generalizability to unseen samples of the base classes is limited.

In this paper, we present a novel morphable detector (MD) that simply "morphs" some of its changeable parameters online estimated from the few samples, so as to detect novel classes without any extra training. Different from most existing object detectors, this novel MD has two sets of parameters, one for the feature embedding (i.e. the network parameters), and the other for class representation (called "prototypes") as illustrated in Fig 2. We view the MD for recognizing visual samples of different classes as if they live in a common space called feature space. Each class is associated with a prototype which is the targeted coordinate of each class in the feature space. Therefore, for each object proposal, the MD learns a feature vector whose similarity with prototypes is regarded as the foreground classification score. As it is hard to assign one prototype to the background, the MD directly regresses a background score from the visual features as shown in Fig 2. Once an MD is learned, it is able to use a few samples of a novel class to directly compute its prototype to fulfill the online morphing process (details are in 3.3).

The learning of the MD is based on the alternate learning of the feature embedding and the prototypes in an EM-like approach as shown in Fig 3. The prototype is regarded as a hidden variable to be learned by integrating the visual and semantic embeddings. In "E"-step, we fix the network parameters and update the prototypes by combining the current prototypes and the feature vectors of the training samples on the trained model (details are in Sec. 3.2.2). In "M"-step, the prototypes are fixed and the network is trained using the updated prototypes. The prototypes are initialized with semantic vectors which bring useful external information from textual data. But note that directly using semantic vectors as prototypes without the proposed EM-like algorithm still suffers from limited generalizability (to novel classes) because the external information does not directly examine visual appearances while the model concerns itself with recognizing visual features. Therefore, the joint learning of the feature embedding and prototypes allows better recovery of an unknown prototype from a few samples of a novel class. Our approach is different from the existing approach, such as RepNet [21] which learns representatives for each class from the visual data in an end-to-end training. The proposed MD learns the representatives (prototypes) by an EM-like approach where the visual and semantic information are integrated to improve the model's generalizability (to novel classes).

Overall, the contributions of this work are four-fold:

- We study a special few-shot detection task, Object Detection on Demand, which is rarely discussed in the literature and can not be solved by many existing Few-shot detection methods.
- We present a novel morphable detector (MD) which can be online morphed to detect the novel classes without extra training.
- We propose to learn the MD by joint visual and semantic embedding in an EM-like approach.
- Extensive experiments are performed on different datasets to demonstrate the superiority of the MD over other methods.

2. Related Work

Zero / few Shot Learning Zero Shot Learning (ZSL) [10, 9, 35, 40, 1, 22, 23, 53] has been widely studied for image recognition. It aims to recognize unseen classes without training samples. People usually leverage semantic information [10, 9, 35, 40] or attributes representation [1, 22, 23] for ZSL. Few Shot Learning aims to recognize a class with a few annotated samples. People try to address this problem by metric learning based approaches [45, 42, 33, 43, 16] or meta-learning based approaches [36, 50, 31, 32]. Different from them, in this work, we focus a more challenging object detection task.

Object Detection tasks Most of existing detection methods [4, 5, 25, 28, 26, 37, 38, 39, 52, 51, 41, 15, 14] focus on general object detection task where each category has large number of annotated data. However, when the labeled data are scarce or not available, the models can overfit or fail to generalize. So people start to focus on zero-shot / few-shot detection [24, 2, 55, 20, 46, 47, 49, 48, 6] tasks where no example or a few examples for novel category are given. But the models can suffer from catastrophic forgetting [29] when the past data are not available. People start to work on incremental few-shot detection [34] task. Different from the above tasks, we focus on a more challenging task, Object Detection on Demand, where only a few samples for novel categories are given and no extra training is required.

Non-morphable Detector Recent works [55, 20, 46, 47, 49, 48, 6] have made significant progresses on few-shot detection tasks. They aim to leverage fully labeled seen category data to train a base model and adapt this model to novel classes using a meta feature learner(i.e. extra training). However, extra training is unfeasible for emerging applications of robots. Different from them, we propose to train a morphable detector that can be online morphed to detect novel categories without any extra training.

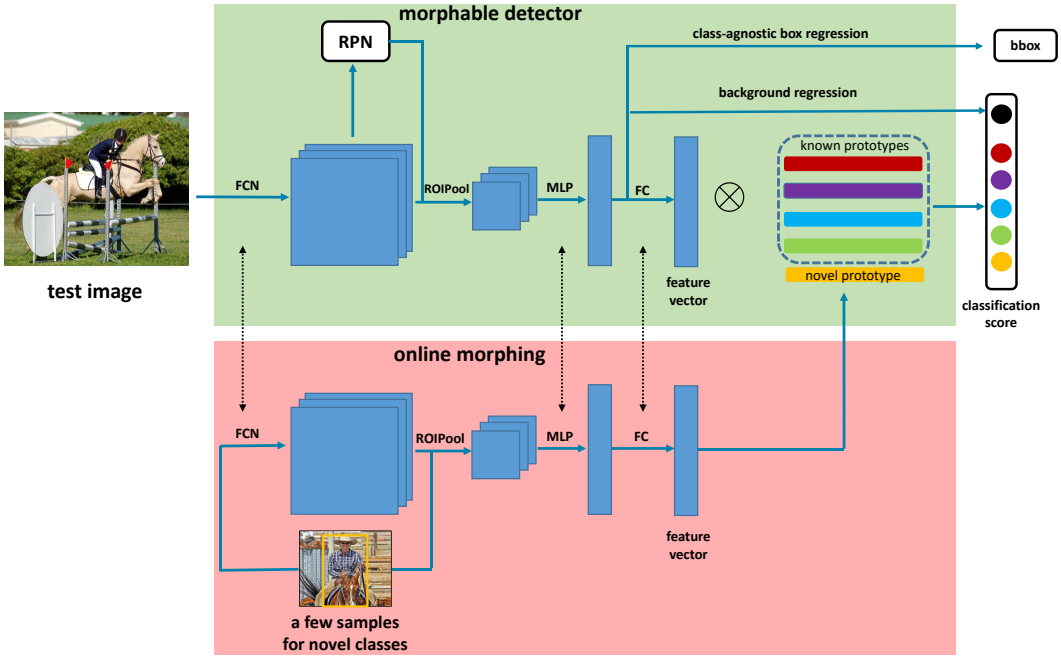

Figure 2: The proposed morphable detector (MD) structure. Given a trained MD, it is able to compute the representations (prototypes) for the novel classes using a few samples from novel classes (see sec. 3.3). Given a test image and proposals generated by RPN, the MD outputs the feature vectors, box regression, and background score for each proposal. The similarity between the feature vectors with the prototypes associated with each novel class is used to estimate the novel class posterior probability (see sec. 3.3).

Morphable Detector One basic morphable detector is a zero-shot detector [24, 54, 2] which can detect the novel categories by leveraging semantic information without any annotated examples. However, as the semantic information does not directly examine the visual appearances, zero-shot detectors have limited generalization capability and the overall performance is far from satisfactory. Another morphable detector is some few-shot detector which takes "query-target" pairs as input to detect all instances of the "target" object appearing in the "query" image. But this model learns less discriminative representations as the target representations are always changing in training. RepNet [21] propose to learn a few representatives for each category from the visual data. But learning from visual data is not enough for the model generalization. Different from them, we propose to leverage external semantic information [30] and present an EM-like approach to integrate the visual and semantic embeddings.

3. Morphable Detector

In this paper, we present a novel Morphable Detector (MD) which can be online morphed to detect the novel classes without extra training. As illustrated in Fig 3, we propose to learn the prototypes and network parameters alternately in an EM-like approach, with the other fixed in each iteration. Fig 2 illustrates how the morphable detector(MD) is morphed to detect the novel classes given a few samples from novel classes. Once the MD is trained, the MD only needs to forward a few samples of a novel class through the network to compute its prototype (details are in sec. 3.3) to detect the novel classes.

3.1. Basic Morphable Detector

We have base class set $\mathcal{C}_{base}$ and novel class set $\mathcal{C}_{\text{novel}}$, in which $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{novel} = \phi$. We denote the base class dataset as D_{base} which consists of the training images and the corresponding box annotations. The MD framework applies to a variety of CNN-based detectors [41, 7, 25, 5]. Here we instantiate the framework with Faster R-CNN(FRCNN) [41] because it is a simple and widely used framework. The MD uses Region Proposal Network (RPN) [41] to generate proposals and ROI pooling to extract the proposal features as illustrated in Fig. 2. The MD has two sets of parameters: the network parameters and the class representation (called "prototypes"). We denote the prototypes for base and novel classes as $\mathcal{P}_{\text{base}}$ and $\mathcal{P}_{\text{novel}}$ respectively. $\mathcal{P}_{\text{base}}$ is learned by joint visual and semantic embeddings in the EM-like approach. Once the MD is trained, $\mathcal{P}_{\text{novel}}$ can be computed by forwarding the samples from novel classes through the trained network. Specifically, a trained MD's parameters consist of the network parameter Θ and prototypes $\mathcal{P}_{\text{base}}$. Once $\mathcal{P}_{\text{novel}}$ is computed, the MD can be online morphed

to a new detector whose parameters consist of Θ, $\mathcal{P}_{\text{base}}$ and $\mathcal{P}_{\text{novel}}$. As a result, the new detector can detect novel classes.

3.2. Training of Morphable Detector

In the Morphable Detector (MD), each class is associated with one prototype. Suppose we have prototypes for base classes $\mathcal{P}_{\text{base}} = \{p_j\}$ where j indicates the class. MD generates the proposals $\{x_i, y_i\} \in \text{ROI}$ by the RPN [41]. The MD learns prototypes for the base classes and a feature space where the feature vector of a given sample is expected to be close to the corresponding prototype while far away from prototypes of other classes. The objective is to maximize the likelihood:

$$\sum_{x_i, y_i \in \text{ROI}, y_i > 0} P(y_i|p_i)P(p_i|x_i) + \sum_{x_i, y_i \in \text{ROI}, y_i = 0} P(y_i|x_i) \quad (1)$$

where $P(p_i|x_i)$ or $P(y_i|x_i)$ is determined by the network output and $P(y_i|p_i)$ is determined by the prototype associated with each class.

To maximize the above likelihood, we regard the prototype as a hidden variable and propose to learn it by integrating the visual and semantic embedding. The feature embedding and the prototypes are alternately learned in an EM-like approach where the prototypes are initialized with semantic vectors in the initial training and recursively updated over iterations. In "E"-step, the network parameters Θ are fixed so the $P(p_i|x_i)$ is a constant in Eq 1. It aims to update the prototypes for next iterative training. So, the "E" step takes provided base class data D_{base}, the trained model $\mathcal{N}^t$ and the current prototypes $\mathcal{P}^t_{base}$ as inputs, then outputs updated prototypes $\mathcal{P}^{t+1}_{\text{base}}$.

$$\mathcal{P}^{t+1}_{\text{base}} = E(D_{\text{base}}, \mathcal{N}^t, \mathcal{P}^t_{base}). \quad (2)$$

In "M" step, with the prototypes fixed, $P(y_i|p_i)$ is a constant. The optimization is essentially equivalent to the maximum-likelihood estimation of the network parameters. Therefore, the "M" step takes the training data D_{base} and the prototypes $\mathcal{P}^{t+1}_{\text{base}}$ for base classes as input, and outputs a newly trained network model $\mathcal{N}^{t+1}$.

$$\mathcal{N}^{t+1} = M(D_{\text{base}}, \mathcal{P}^{t+1}_{\text{base}}). \quad (3)$$

The model usually converges after several iterations.

3.2.1 Network Training

Given the extracted proposals $\{x_i, y_i\} \in \text{ROI}$, the deep visual features for each proposal x_i are extracted as $\phi(x_i)$. As it is hard to assign a prototype to the background, the MD directly regresses a background score from the visual features. $\phi(x_i)$ is forwarded through two separate fully connected layers to obtain the background score $b_i \in \mathbb{R}^1$ and the feature vector $f_i \in \mathbb{R}^d$, where d is the dimension of the feature vector. The network is trained with a prototype-based contrastive loss which consists of two terms: the foreground loss and background loss. The foreground loss encourages the feature vector of the proposal $\{x_i, y_i \in \text{ROI}\}$ to be close to the corresponding prototype while far away from other prototypes if the proposal belongs to the foreground (i.e. $y_i > 0$). So, the foreground loss is defined as

$$L_{FG} = \sum_{y_i > 0} -\log\left(\frac{\exp(f_i \cdot p_{y_i})}{\exp(b_i) + \sum_{p_m \in \mathcal{P}_{\text{base}}} \exp(f_i \cdot p_m)}\right), \quad (4)$$

where p_{y_i} is the prototype corresponding to class y_i. When the proposal belongs to the background, the contrastive loss encourages the background score to be high. The background loss is defined as

$$L_{BG} = \sum_{y_i = 0} -\log\left(\frac{\exp(b_i)}{\exp(b_i) + \sum_{p_m \in \mathcal{P}_{\text{base}}} \exp(f_i \cdot p_m)}\right). \quad (5)$$

The overall loss is sum of the foreground loss, background loss, and a class-agnostic bounding box regression loss [41]

$$L = L_{\text{BG}} + L_{\text{FG}} + L_{\text{bbox}}. \quad (6)$$

3.2.2 Prototype update

We denote one ground truth box for class j as $x_i \in g_j$ and the feature vectors of it on the network $\mathcal{N}$ is $\mathcal{N}(x_i)$. Then we compute the mean feature vector v_j of all samples from each class j as

$$v_j = \frac{1}{|g_j|} \sum_{x_i \in \text{g}_\text{j}} \mathcal{N}^t(x_i). \quad (7)$$

Then we use the mean feature vector v_j to compute the new prototype p^{t+1}_j by fusing it with the current prototypes p^t_j (associated with class j) by weighted element-wise sum,

$$p^{t+1}_j = (1 - \lambda)v_j + \lambda p^t_j, \quad (8)$$

where λ is a constant between 0 and 1. Note that before the element-wise sum, both prototypes and mean feature vectors are normalized.

3.3. Online Morphing

The online morphing is to compute the new prototypes for the novel classes as shown in Fig 2. Suppose we have a ground truth box $x_i \in g_j$ from novel class j, and forward the samples through the network to get the feature vectors $\mathcal{N}(x_i)$. The mean feature vector of all samples belonging to a novel class j is used as the new prototype for that class,

$$p_j = \frac{1}{|g_j|} \sum_{x_i \in \text{g}_\text{j}} \mathcal{N}(x_i). \quad (9)$$

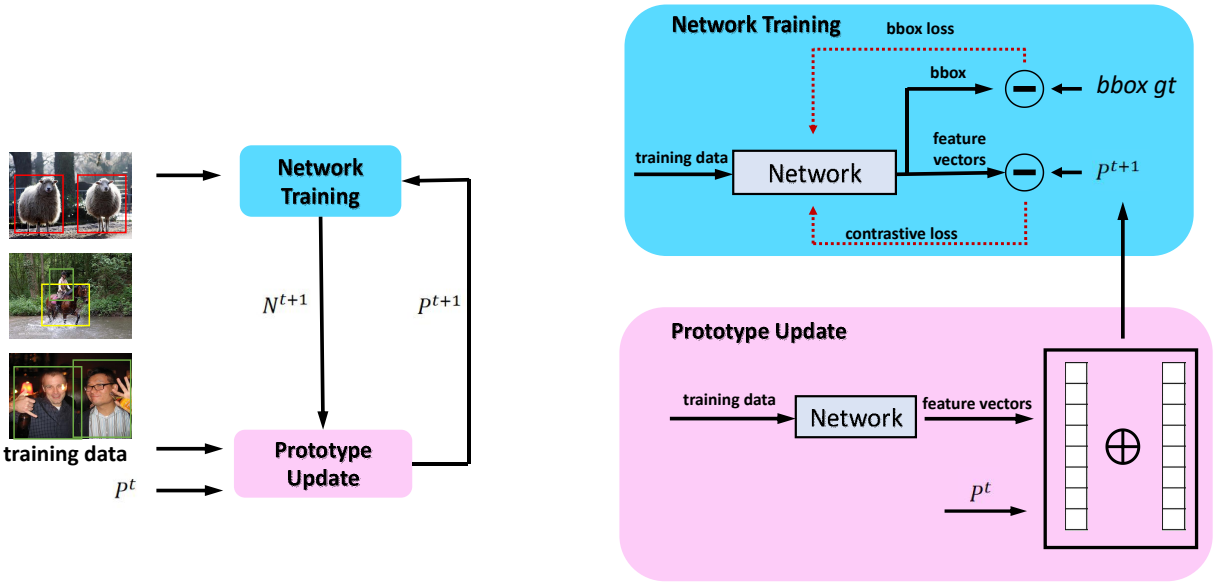

Figure 3: Training framework (EM-like approach). The learning of the MD is based on the alternate learning of the feature embedding and the prototypes in an EM-like approach. In "E" step, with the network fixed, we compute the mean feature vector for each base class on the feature space to update the prototypes associated with that class (see sec. 3.2.2). In "M" step, with the prototypes fixed, we use the prototypes computed in "E" step to train the network (see sec. 3.2.1).

Method	Extra training	AP	$AP_{0.5}$
FRCNN [41]	✓	-	23.0
LSTD [6]	✓	-	24.2
FSOD method [13]	✗	-	27.5
Visual (ImageNet)	✗	10.1	16.3
Visual (FRCNN)	✗	15.5	22.6
MD (concat)	✗	17.3	29.9
MD ($\lambda = 0$)	✗	21.5	36.2
MD ($\lambda = 0.3$)	✗	21.3	35.9
MD ($\lambda = 0.7$)	✗	21.6	36.3
MD (iter1)	✗	18.2	31.2
MD (iter2)	✗	21.9	36.7
MD	✗	**22.2**	**37.1**

Table 1: Comparison of our method with different baselines and different variants on FSOD Dataset

Now, we have the $\mathcal{P}_{\text{novel}} = \{p_j\}$ where j is the class index for a novel class, so the novel class can be detected. Given a test image, the RPN first generates the proposals $x_i \in \text{ROI}$, and get the bbox score b_i and feature vector f_i. The class posterior probability for class j is,

$$\frac{\exp(f_i \cdot p_j)}{\exp(b_i) + \sum_{p_m \in \mathcal{P}_{\text{base}} \cup \mathcal{P}_{\text{novel}}} \exp(f_i \cdot p_m)}. \quad (10)$$

Then, the detected boxes are obtained by setting a threshold for the class score like other detectors [41].

Method	Split 1		Split 2		Ave
	AP	$AP_{0.5}$	AP	$AP_{0.5}$	$AP_{0.5}$
OSOD [18]	-	-	-	-	22.0
MD (iter1)	20.2	32.9	21.1	32.6	32.8
MD	**21.5**	**33.0**	**24.9**	**36.1**	**34.6**

Table 2: One-shot detection performance comparison on the first two splits of COCO dataset for the novel classes.

4. Experiments

To evaluate our morphable detector (MD) on Object Detection on Demand (ODOD), we begin with an evaluation on a challenging large-scale dataset FSOD [13] which benchmarks the performance of detectors on few-shot detection setting. Then we evaluate on two widely used datasets MS COCO [27] and Pascal VOC datasets [12], and compare against state-of-the-arts on few-shot detection setting. Finally, as a by-product, we compare against state-of-the-arts on zero-shot detection setting.

4.1. Experiments on FSOD dataset

Dataset and implementation Few Shot Object Detection (FSOD) [13] dataset is proposed to evaluate the detector which is trained using base class data and evaluated on the novel classes. This dataset contains 1000 classes with 800/200 split for training and test set respectively. There is no overlap between the training and test classes. There are

Method	Ours (1-shot)	FSView [48] (1-shot)	LSTD [6] (10-shot)	MetaYOLO [20] (10-shot)	MetaDet [47] (10-shot)	MetaRCNN [49] (10-shot)	TFA w/fc [46] (10-shot)	TFA-w/cos [46] (10-shot)
AP	9.7	4.5	3.2	5.6	7.1	8.7	**10.0**	**10.0**
$AP_{0.5}$	15.0	12.4	8.1	12.3	14.6	**19.1**	-	-
$AP_{0.75}$	**9.9**	2.2	2.1	4.6	6.1	6.6	9.2	9.3

Table 3: Performance comparison with state-of-the-arts on COCO dataset for novel classes in split 3. The best one and the second-best one are highlighted in **bold** and underlined respectively.

Method	Ours	FSView [48]	LSTD [6]	MetaYOLO [20]	MetaDet [47]	MetaRCNN [49]	TFA w/fc [46]	TFA w/cos [46]
Split 1	**53.2**	24.2	8.4	14.8	18.9	19.9	22.9	25.3
Split 2	**41.6**	21.6	11.4	15.7	21.8	10.4	16.9	18.3
Split 3	**38.6**	21.2	12.6	21.3	20.6	14.3	15.7	17.9

Table 4: Performance comparison with state-of-the-arts PASCAL VOC dataset for novel classes in three splits.

52350 images with 147489 annotated boxes in the training set and 14152 images with 35102 annotated boxes in the test set. We use the Region Proposal Network and class-agnostic regression from Faster-RCNN in our model. Following the training strategies in [13], we use ResNet-50 as our backbone and pretrain the model on the COCO dataset. Then we train the model using the base class data which contains 800 classes and test on the test set which contains 200 novel classes. We randomly select 5 samples for each novel class as known samples for the novel classes as [13]. We train the model with batch size 4 for 50k iterations with a learning rate 0.002 and another 20k iterations with a learning rate 0.0002. All models are evaluated using standard $AP_{0.5}$ and AP. The dimension size of the feature embedding and semantic vectors [1] is 200.

Comparison with baselines We first compare against different baselines which use different prototype initializations. Then, we evaluate the MD model after each iteration.

- **Visual (ImageNet/FRCNN)**. In our MD, the prototypes are initialized using semantic vectors. In the experiments, we compare against MD variants using visual features for prototype initialization. To get the visual features for each class, we forward the base class ground truth samples through a trained Faster-RCNN model or ImageNet pre-trained model to get the visual features whose dimension is 1024 in our experiment. Then we use the mean feature vectors for each class as the prototype for that class to train the MD.
- **Iterative results**. We evaluate the MD model's performance after each iteration.

Table 1 summarizes the comparisons against the baselines. Directly using ImageNet [8] or FRCNN [41] features as prototypes does not work well. The results show that the model learned only using visual features can not be well generalized to novel classes. The semantic information learned from text data provides useful information about the relationship between different classes. So, the MD (iter 1) which uses semantic vectors as prototypes obtains significant improvements over the model only using visual features. This verifies that semantic vectors can help improve the model's generalizability to novel classes. Note that the semantic information does not examine the visual appearances. As a result, the overall performance is still limited. To overcome this limitation, our MD is learned by integrating the visual and semantic embeddings. The results show that the model's performance can be significantly improved by the joint visual and semantic embedding. We empirically find that the performance can not be further improved after 2-3 iterations so we set the number of iterations to be 3.

Ablation study To study the best way to combine the mean feature vector and the prototypes to compute the new prototypes, we compare "element-wise sum" and "concatenate". After the initial training, we concatenate or element-wise sum the mean feature vectors of the ground truth samples and the prototypes (i.e., the first item and second item in Eq. 8). The experimental results show that the element-wise sum is a better combination way, so the MD uses element-wise sum in the remaining experiments. Then, we compare the MDs using different λ in Eq. 8, and we find that $\lambda = 0.5$ works the best.

Comparison with state-of-the-arts We compare against three state-of-the-arts: FSOD method [13], FRCNN [41], LSTD [6]. FRCNN and LSTD results are reimplemented by [13]. Our MD obtains $AP_{0.5}$ 10 points improvements over FSOD and 13 points improvement over LSTD. Unlike FSOD method [13] which takes 'query-target' pairs as input to train the model, our morphable detector learns more discriminative features and leverages useful external textual in-

[1] We use the extracted semantic vectors from https://github.com/agnusmaximus/Word2Bits

formation. Therefore, our detector has better generalizability to novel classes. To verify that the MD can be well generalized to novel classes, we randomly selected 50 classes from the test set and visualize the feature vectors by t-SNE tool [44]. Fig 5 shows that objects from most novel classes are clustered together on the learned feature space.

Computation time It takes around 0.04 seconds in ResNet 50 and 0.09 seconds in ResNet 101 to add a new class using a single RTX 3090 GPU.

4.2. Experiments on Pascal and COCO

Datasets and implementation On Pascal VOC [11], VOC 07 and 12 train/val sets are used for training and VOC 2007 test set is used for testing. In order to fairly compare with the state-of-the-art methods [20, 6, 49], we follow [20] to use the same novel splits. On MS COCO [27], we follow [25] and train the model using the union of 80k train images and a 35k subset of val images (trainval35k [3]), and report the testing performed on a 5k subset of val images (minival). We group the 80 classes in COCO dataset into 5 different semantic clusters and randomly select two classes from each cluster as novel classes(i.e. 10 classes in total). We randomly select two splits using this strategy. In the split 1, we select "*bicycle*", "*car*", "*dog*", "*sheep*", "*frisbee*", "*surfboard*", "*pizza*", "*laptop*", "*microwave*" and "*refrigerator*" as the novel classes. In the split 2, we select "*car*", "*train*", "*boat*", " *dog*", " *horse*", "*skateboard*", "*sandwich*", "*pizza*", "*keyboard*" and "*microwave*" as the novel classes. To fairly compare with most few-shot detectors [20, 6, 49, 47, 46, 48], we also perform experiments using the same split as them (i.e., using 20 Pascal VOC classes as novel classes). Following these few-shot detectors, we ignore the novel classes annotations in training and randomly select one example as a given example for each novel class in testing. So, we perform one-shot detection experiments. We use ResNet-101 [17] as the backbone and train the model with batch size 4 for 50k iterations and 160k iterations for Pascal VOC and COCO separately. The initial learning rate is 0.002 and it is decreased to 0.0002 after 40k and 120k iterations for Pascal and COCO. We use the same embedding size and semantic vectors used in FSOD experiments.

4.3. Comparison on novel classes

Table 2 summarizes the comparisons on COCO dataset for split 1 and 2. Over iterations, the MD's performance consistently improves on the two splits. This verifies the effectiveness of the proposed EM-like approach. We also use the average performance to compare against one-shot-detector [18] which perform their experiments on another four random splits. Same with [13], OSOD [18] takes "query-target" pairs as input. So, our MD has better generalizability to the novel classes than [18]. Table 3 summarizes the comparison against several state-of-the-arts [20, 6, 49, 47, 46, 48] on COCO dataset for split 3. Among them, only FSView [48] reported their performance on one-shot detection settings. For others, they reported their 10-shot performance in their paper. Our method outperforms FSView by a large margin in the 1-shot detection setting and outperforms most of the others even though we only use 1-shot data. More importantly, as all these methods need extra training, they can not be deployed on the embedded systems as our method. Note that the performance may vary for different splits. The reason is that as the prototypes are initialized with semantic vectors, the relationship between base and novel classes can influence the MD's performance. In the first two splits, the base and novel classes are split based on semantic clusters of the classes, so the results of them can be obviously better than those of split 3. Table 4 summarizes the comparison against the state-of-the-arts on three splits of the Pascal VOC dataset. Our MD outperforms state-of-the-arts by a large margin on the three splits. Our MD leverages semantic and visual information to help generalize the trained model to novel classes.

Method	Split 1		Split 2		Split 3		Ave
	AP	$AP_{0.5}$	AP	$AP_{0.5}$	AP	$AP_{0.5}$	$AP_{0.5}$
FRCNN [41]	37.3	59.9	36.9	58.7	37.0	59.1	59.2
OSOD [18]	-	-	-	-	-	-	40.9
MD (iter1)	37.5	60.6	**37.2**	**59.3**	37.2	59.0	59.6
MD	**37.8**	**60.7**	36.9	59.0	**37.5**	**59.2**	**59.7**

Table 5: Performance comparison on the COCO dataset for base classes.

4.4. Comparison on base classes

Table 5 and 6 summarize the comparison against our baselines and state-of-the-arts on Pascal and COCO datasets for base classes. Our MD obtains obviously a bit better performance on the base classes over iterations. This verifies the proposed EM-like algorithm can help improve the model's generalizability to unseen samples of base classes. Our MD performs much better than OSOD [18] which takes "query-target" pairs as input. This verifies that OSOD [18] learns much less discriminative features for base classes. Compared with state-of-the-arts few-shot detectors, our MD performs the best for the base classes. The reason is that these FSD models take a small number of base class data to further train the model so these models can easily overfit to the small data. Compared with FRCNN [41], our model can still outperform it. This shows the advantage of our MD over FRCNN on the base classes. Note that FRCNN can not be generalized to novel classes without extra training.

4.5. Comparison with zero shot detectors

As a by-product, we also perform the MD under zero-shot detection setting. In training, we use the semantic vec-

Figure 4: Some qualitative results of our proposed Morphable Detector on FSOD test set.

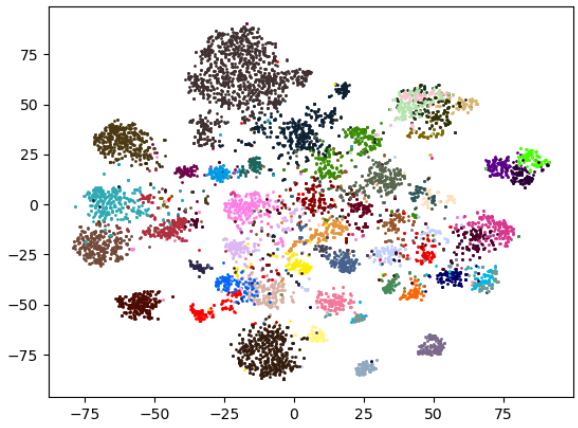

Figure 5: t-SNE visualization of feature embeddings of objects from randomly selected 50 novel classes in FSOD test set on the learned MD.

Method	Split 1	Split 2	Split 3	Ave
MetaRCNN [49]	64.8	-	-	64.8
TFA w/cos [46]	79.2	-	-	79.1
OSOD [18]	-	-	-	60.1
MD (iter1)	80.2	81.6	78.4	80.1
MD	**80.7**	**82.1**	**79.2**	**80.7**

Table 6: Performance comparison on the PASCAL VOC dataset for base classes.

tors for the base classes as prototypes to train the MD. In testing, we use the semantic vectors for the novel classes as novel prototypes. Table 7 summarizes the comparison against state-of-the-art zero shot detectors using standard evaluation metrics recall@100 and $AP_{0.5}$. Our MD obtains very impressive results on the first two splits. The performance on the split 3 is not as good as the other two splits because the relationship between base and novel classes on split 3 does not help much as the other splits. Our MD obtains an average of 39.7 Recall@100 which is better than other zero-shot detectors.

Method	Recall@100	$AP_{0.5}$
SB [2]	22.4	0.7
DSES [2]	27.2	0.54
TD [24]	34.3	-
DELO [55]	33.5	7.6
MD (split 1)	44.8	9.4
MD (split 2)	**47.2**	**9.8**
MD (split 3)	27.0	4.2

Table 7: Comparison under zero shot detection setting on COCO dataset for novel classes.

5. Conclusion

In this paper, we focus on a very challenging task: object detection on demand (ODOD) task. The prevailing FSD methods can not solve this problem well as ODOD requires no extra training. We propose a novel morphable detector (MD), that simply "morphs" some of its changeable parameters online estimated from the few samples, so as to detect novel categories without any extra training. The learning of the MD is based on the alternative learning of the feature embedding and the prototypes in an EM-like approach which allows better recovery of an unknown prototype from a few samples of a novel category. Extensive experiments are performed to demonstrate the superiority of the MD.

Acknowledgements We thank Pengbo Zhao for the suggestions on the writing. This work was supported in part by National Science Foundation grant IIS-1619078, IIS-1815561, and IIS-2007613.

References

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1425–1438, 2015. 2

[2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 384–400, 2018. 2, 3, 8

[3] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2874–2883, 2016. 7

[4] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European conference on computer vision*, pages 354–370. Springer, 2016. 2

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2, 3

[6] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. Lstd: A low-shot transfer detector for object detection. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2, 5, 6, 7

[7] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[9] Mohamed Elhoseiny, Babak Saleh, and Ahmed Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2584–2591, 2013. 2

[10] Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. Link the head to the" beak": Zero shot learning from noisy text description at part precision. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6288–6297. IEEE, 2017. 2

[11] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 7

[12] 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. 1, 5

[13] 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. 1, 2, 5, 6, 7

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[16] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[18] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems*, pages 2725–2734, 2019. 2, 5, 7, 8

[19] Xiaotang Jiang, Huan Wang, Yiliu Chen, Ziqi Wu, Lichuan Wang, Bin Zou, Yafeng Yang, Zongyang Cui, Yu Cai, Tianhang Yu, Chengfei Lv, and Zhihua Wu. Mnn: A universal and efficient inference engine. In *MLSys*, 2020. 1

[20] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8420–8429, 2019. 1, 2, 6, 7

[21] Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogerio Feris, Raja Giryes, and Alex M Bronstein. Repmet: Representative-based metric learning for classification and few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2019. 2, 3

[22] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009. 2

[23] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, 2013. 2

[24] Zhihui Li, Lina Yao, Xiaoqin Zhang, Xianzhi Wang, Salil Kanhere, and Huaxiang Zhang. Zero-shot object detection with textual descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8690–8697, 2019. 2, 3, 8

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2, 3, 7

[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[27] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 5, 7

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2

[29] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 2

[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013. 3

[31] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017. 2

[32] Tsendsuren Munkhdalai, Xingdi Yuan, Soroush Mehri, and Adam Trischler. Rapid adaptation with conditionally shifted neurons. In *International Conference on Machine Learning*, pages 3664–3673. PMLR, 2018. 2

[33] Boris N Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018. 2

[34] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13846–13855, 2020. 2

[35] Ruizhi Qiao, Lingqiao Liu, Chunhua Shen, and Anton Van Den Hengel. Less is more: zero-shot learning from online textual documents with noise suppression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2249–2257, 2016. 2

[36] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016. 2

[37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 2

[38] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 2

[39] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[40] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58, 2016. 2

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2, 3, 4, 5, 6, 7

[42] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017. 2

[43] Eleni Triantafillou, Richard S Zemel, and Raquel Urtasun. Few-shot learning through an information retrieval lens. In *NIPS*, 2017. 2

[44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 7

[45] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016. 2

[46] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020. 1, 2, 6, 7, 8

[47] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9925–9934, 2019. 1, 2, 6, 7

[48] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *European Conference on Computer Vision*, pages 192–210. Springer, 2020. 1, 2, 6, 7

[49] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 9577–9586, 2019. 1, 2, 6, 7, 8

[50] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*, pages 2365–2374, 2018. 2

[51] Xiangyun Zhao, Shuang Liang, and Yichen Wei. Pseudo mask augmented object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4061–4070, 2018. 2

[52] Xiangyun Zhao, Samuel Schulter, Gaurav Sharma, Yi-Hsuan Tsai, Manmohan Chandraker, and Ying Wu. Object detection with a unified label space from multiple datasets. In *European Conference on Computer Vision*, pages 178–193. Springer, 2020. 2

[53] Xiangyun Zhao, Yi Yang, Feng Zhou, Xiao Tan, Yuchen Yuan, Yingze Bao, and Ying Wu. Recognizing part attributes with insufficient data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 350–360, 2019. 2

[54] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 3

[55] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Don't even look once: Synthesizing features for zero-shot detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11693–11702, 2020. 2, 8