Reducing the need for bounding box annotations in Object Detection using Image Classification data

Leonardo Blanger
Neuralmind AI
Institute of Mathematics and Statistics
University of São Paulo, Brazil
leonardoblanger@gmail.com

Nina S. T. Hirata
Institute of Mathematics and Statistics
University of São Paulo, Brazil
nina@ime.usp.br

Xiaoyi Jiang
Faculty of Mathematics
and Computer Science
University of Münster, Germany
xjiang@uni-muenster.de

***Abstract*—We address the problem of training Object Detection models using significantly less bounding box annotated images. For that, we take advantage of cheaper and more abundant image classification data. Our proposal consists in automatically generating artificial detection samples, with no need of expensive detection level supervision, using images with classification labels only. We also detail a pretraining initialization strategy for detection architectures using these artificially synthesized samples, before finetuning on real detection data, and experimentally show how this consistently leads to more data efficient models. With the proposed approach, we were able to effectively use only classification data to improve results on the harder and more supervision hungry object detection problem. We achieve results equivalent to those of the full data scenario using only a small fraction of the original detection data for Face, Bird, and Car detection.**

I. Introduction

Training Object Detection architectures requires large amounts of images labeled with bounding boxes. When compared with the more traditional Image Classification task, labeling data for Object Detection is much slower and more expensive. Bounding boxes are also more dependent on human intervention, being more vulnerable to labeling mistakes and biases. Thus, it is of great interest to develop effective ways to train models with fewer labeled samples.

Several techniques can be employed to reduce the dependency on large annotated datasets. Transfer Learning from other tasks [1] (e.g. ImageNet classification) may lead to a good initialization before finetuning on labeled target data. Data Augmentation techniques can be used to generate additional samples by applying random, label preserving transformations to existing ones. Augmentation is effective, but limited in the sense that no object instances beyond those already present in the dataset will ever be seen. **Sample Synthesis** is a potentially more general approach, in which completely new instances are artificially created. Evidence for the effectiveness of Sample Synthesis has been reported for several Computer Vision tasks [2]–[8]. In this work, we turn our attention to Object Detection Sample Synthesis.

Some works have investigated the use of artificial samples for Object Detection [7], [8]. However, they usually depend on other expensive forms of supervision in order to generate samples. Additionally, they all focus only on how to create samples, while little attention has been given on how best to incorporate these samples during Object Detection training procedures.

In this work, we set out to investigate ways to generate and make use of artificial detection samples that require no expensive supervision. In particular, we propose taking advantage of existing cheaper image classification data, in such a way as to improve data efficiency on the Object Detection problem.

Our method consists in combining classification images with a generative unsupervised technique [9], to build a sample synthesis pipeline, capable of automatically generating an infinite stream of artificial samples with bounding box annotations. In order to avoid expensive supervision, all stages of this synthesis pipeline are trained using classification data only.

On the issue of how best to use these artificial samples, our main finding is that pretraining detection models with artificial samples before finetuning them on real images is very effective. This is in contrast to the simpler approach followed by related works [7], [8], of simply training with mixed real and artificial samples. We thoroughly demonstrate how this simple pretraining approach works as a powerful initialization strategy, resulting in a more data efficient training, which in turn, allows competitive detection results using only a small fraction of the original real labeled detection data.

The contributions of this work are the following: **(1)** We show that it is possible to automatically generate artificial labeled detection samples using a simple pipeline of already existing techniques, all of which can be trained with only classification level supervision. **(2)** We show how such artificial samples are a viable way to reduce the dependency of detection models on labeled data. And **(3)**, we propose using these samples as a pretraining initialization strategy for detection models, and experimentally show how this approach leads to more data efficient training.

The remainder of this paper is structured as follows. Section II discusses how existing works deal with the task of

sample synthesis, and justifies the choices we made in this project. Section III describes our method. Section IV presents a series of experiments we conducted in order to evaluate and better understand our method. Finally, Sections V and VI present, respectively, discussions and our concluding remarks.

II. Background

There are several options to automatically generate artificial labeled samples to train Computer Vision models. A straightforward form of Sample Synthesis is to use Computer Graphics [10] to render instances of objects paired with the respective labels. However, this approach requires heavy human intervention, as one needs access to some graphical model of the objects, which needs to be manually designed most of the time.

Another option is to use some generative model, such as GANs [11], to synthesize image samples. This approach is already well established in the context of Image Classification, as demonstrated for instance in [2], [3]. There are also a few works that attempted GAN based synthesis for Detection and Segmentation on medical [4], [5] and aerial [6] images. However, they all require expensive supervision like bounding boxes or masks for training these generative models, so they are not ideal as a means of reducing the dependency on annotated data.

In the case of Detection, a popular approach is to start from images of the objects of interest, and then crop and paste the object regions on top of random background scenes. For instance, [7] showed improved results on the Pascal VOC Detection dataset [12] simply training with additional artificial images that were created by placing cropped instances of the objects on top of background scenes. Another similar approach [8] was applied to the problem of Instance Detection, a form of Object Detection that involves discriminating between different instances of the same object class. The problem formulation in [8] allowed them to train a segmentation model using masks available for other instances of the same objects.

The major limitation of the above mentioned works is the need for segmentation mask annotations in order to crop the object regions. Although the intention in [7], [8] was not primarily to reduce the need for bounding boxes for general Object Detection, their results suggest that, if the samples could be generated without expensive supervision, it could be possible to perform detection without requiring as much expensive masks or bounding boxes. In this work, we demonstrate how it is possible to use a combination of already existing techniques to generate artificial detection samples, starting from only classification data.

Moreover, a question that is not addressed by existing works is how best to incorporate these artificial samples into the training of regular detection models. In this work, we propose using artificial samples as a form of pretraining initialization, before finetuning the model on real labeled data. We experimentally show how this strategy leads to a more data efficient training.

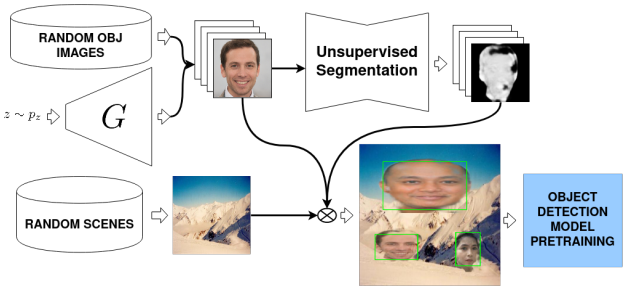

Fig. 1. Object Detection pretraining initialization based on Artificial Samples generated from (real or fake) classification images. Best viewed in color.

III. Proposed Method

This Section presents our proposed method. We first describe our Sample Synthesis pipeline (III-A), which we use to generate artificial samples starting from only classification data. Then, we describe our proposal for using these artificial samples as a pretraining initialization strategy (III-B). Next, we explain how we combine these two ideas together to train detection models using less bounding box annotations (III-C). Figure 1 illustrates the whole method.

A. Sample Synthesis

Traditionally, artificial detection samples are generated by cropping object instances from existing images, and pasting them on top of background scenes. Existing works [7], [8] do this by using mask annotations to crop the object instances. These masks are obtained from existing segmentation annotations [7], or extracted by a segmentation model previously trained on similar annotations [8]. In order to use only classification supervision, we turned our attention to Unsupervised Segmentation.

Unsupervised Segmentation: Some recent works proposed deep learning based unsupervised segmentation methods that rely only on classification level annotations. For instance, in [13], segmentation is performed through an iterative optimization method. In "Copy-Pasting" GANs [14], unsupervised segmentation could theoretically be achieved as a byproduct of the "object discovery" sub-task, although only results using simplified artificial contexts have been presented there. More notably, Unsupervised Segmentation by Redrawing (ReDO) [9] has demonstrated impressive results on a small set of real world objects (faces, flowers, and birds), using a GAN-inspired adversarial training dynamic which depends only on classification images.

We note that, by using any of these segmentation techniques, we could train a segmentation model for the objects of interest, without needing mask or bounding box annotated samples. Then, such model could be used to automatically segment objects from class annotated images, to be then inserted onto random background images.

We also note that we can use any regular generative image model, such as a GAN [11], to generate the object images, from which the object instances are segmented and then cropped. Standard GAN architectures are already trained on

classification style images, so this does not incur any additional supervision penalty on the synthesis pipeline. The advantage of using these "fake" object images, instead of existing real ones is that, in doing so, we can treat the synthesis pipeline as a single component, without having to "carry" a classification dataset around. Additionally, by using a generative model instead of a classification dataset, we can synthesize an infinite stream of artificial detection samples where no object instance will be seen more than once, independent of the size of the original classification dataset.

Detection samples synthesized this way will naturally lack real world realism since no coherence between the inserted object and the background image is enforced, as is the case in [7]. Despite the low quality image composition, we experimentally show how pretraining on these "cheap" but abundant samples is very effective for reducing the need for real labeled detection data.

B. Pretraining Initialization

As mentioned above, existing works suggest that artificial samples might help achieve better performing models. In our proposal, we address the question of how best to incorporate these artificial samples into the regular training process of detection models. One could consider a direct approach of simply mixing a certain proportion of these artificial samples with real training images, as done by [7], [8]. Here we investigate a different approach, in which we pretrain detection models on artificial samples before finetuning them on real data. In this regard, one might question whether using exclusively artificial samples to initialize a model could introduce or amplify some bias from the synthesis mechanism. In Section IV-D1, we show how this pretraining initialization strategy works significantly better than the traditional approach of training on mixed and artificial samples.

C. Complete Method

Figure 1 illustrates our overall strategy. In the top part of the figure, either a real object image or a GAN generated 'fake' image goes through an unsupervised segmentation step, which extracts a segmentation mask of the object. Then, in the bottom part, the segmented objects undergo some simple augmentation operations, and are inserted on randomly chosen background images, at random scales and positions. These augmentations are mirrored on the masks when applicable. The bounding box annotations can be automatically extracted from the masks. The masks are also used to blend the object regions with the background scenes, using a straightforward *alpha-blending*:

$$\text{image} = \text{object} \times \text{mask} + \text{background} \times (1 - \text{mask})$$

The resulting detection samples are used on our proposed pretraining initialization.

We highlight here that the main goal for this work is to identify how to use classification supervision to reduce the need for the more expensive detection supervision. We opted to pursue this objective through the idea of Sample Synthesis. We do not claim (or expect) this synthesis pipeline to be the optimal way to generate detection samples. Instead, we propose this pipeline as a way to demonstrate how it is possible to perform an effective sample synthesis using just a simple combination of already existing techniques, followed by a clever use of such samples during training. To the best of our knowledge, Object Detection Sample Synthesis has not been explored with the goal of reducing the need for annotations.

IV. Experiments

We evaluate our synthesis pipeline and pretraining initialization strategy by performing detection using three object classes: Faces, Birds and Cars. Our choice of objects and datasets was strongly guided by what we knew recent Unsupervised Segmentation techniques could handle. Nonetheless, the results obtained still provide evidence for the effectiveness of Sample Synthesis based Pretraining using classification data. We expect Sample Synthesis to become more widely applicable as unsupervised techniques naturally improve.

The code to replicate the experiments is available at `https://github.com/Leonardo-Blanger/synthesis_pretraining_object_detection`.

A. Datasets

To synthesize detection samples, we generate object images using GAN [11] models, then segment them using the unsupervised ReDO method [9]. We also apply a small set of random augmentations on these object images. We paste the segmented objects at random on top of background images that were sampled from the Pascal VOC dataset [12], as described in Section III-C.

Faces: The object images were generated using a StyleGAN trained on FFHQ [15]. Unsupervised segmentation was performed using a ReDO model trained on the LFW dataset [16], made available by [9]. We finetune and evaluate our detection models on real samples from the FDDB Faces dataset [17], with train/valid/test partitions equal to 1449/581/815.

Birds: Object images were generated using a DM-GAN [18], trained on the CUB-200-2011 dataset [19]. Unsupervised segmentation was performed using a ReDO model, also trained on CUB, and made available by [9]. We finetune and evaluate our detection models on real samples from the CUB dataset as well, with train/valid/test partitions equal to 10345/1000/443.

Note that we are finetuning/evaluating our detection models on the same dataset that was used to train the components of the sample synthesis pipeline. We point out that the DM-GAN [18] and ReDO [9] models were trained using different partitions of the CUB dataset. In order to avoid test leakage through the artificial samples, we use the intersection between the test sets of DM-GAN and Birds ReDO as our test set. Also note that, despite the CUB dataset having bounding box annotations, the ReDO method did not need them for training.

Cars: The object images were generated using StyleGAN [15] trained on LSUN Cars [20]. As [9] did not provide weights for cars, we trained our own ReDO instance with

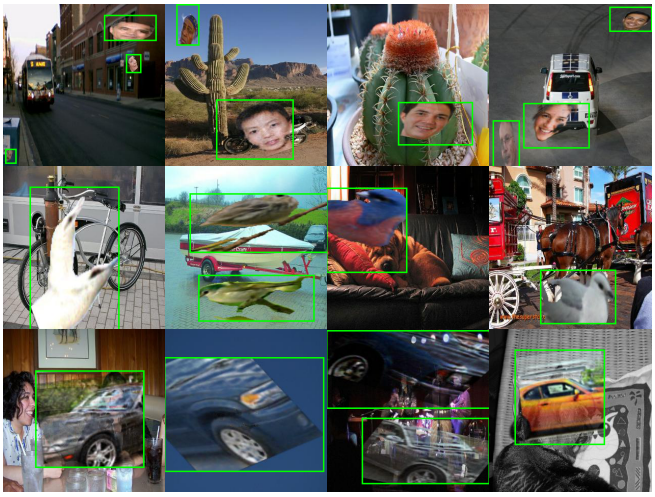

Fig. 2. Artificial detection samples on Faces, Birds, and Cars.

these GAN generated images. We finetuned and evaluated our detection models on the Stanford Cars dataset [21], with train/valid/test partitions equal to 7144/1000/8041.

Figure 2 shows instances of artificial detection images generated by our synthesis pipeline. As expected, these images are far from realistic, but as we demonstrate in Section IV-C, a large number of them can significantly reduce the need for expensive bounding boxes.

B. Training and Evaluation Methodologies

We performed experiments with a Single Shot Detector architecture (SSD) [22], trying both a MobileNet [23] and a ResNet50 [24] networks as backbone CNNs. This detection architecture was trained separately for each object category, both with and without the proposed pretraining initialization. We used the Adam optimizer [25] with 10^{-4} learning rate and the standard $\theta = (0.9, 0.999)$.

The models with pretraining initialization were first trained on the stream of artificial samples for 500 iterations for Faces and Cars, and 1000 iterations for Birds, as we noticed the loss stagnates after these quantities. The non-pretrained baseline models received standard initialization instead, with ImageNet weights for the backbone CNN and Glorot/Xavier initialization [26] for the detection heads.

Then, both the baselines and the pretrained models were finetuned on the real data, again for 500 iterations for Faces and Cars, and 1000 iterations for Birds. We evaluated the model on the validation set every 25 iterations, and choose the checkpoint with the best result as the final trained model. The final results are computed on the test subsets, and measured in AP@0.5 IOU, following [12].

C. Main Results

For each model configuration, training was repeated for distinct amounts of real samples, making sure that all model versions were trained for each considered amount with the same subset of real images. We repeated training three times for each model configuration, and report means and standard deviations.

TABLE I
NUMERICAL RESULTS FOR A FEW OF THE SSD MOBILENET MODELS FROM FIGURE 3 (TOP ROW).

	# real samples	no pretraining	with pretraining
Faces	pretrain only (0%)	–	56.03% ± 3.35%
	10 (∼1%)	27.73% ± 1.02%	72.21% ± 0.74%
	100 (∼7%)	58.73% ± 0.62%	77.86% ± 0.46%
	200 (∼14%)	65.63% ± 0.41%	80.34% ± 0.54%
	800 (∼55%)	76.78% ± 2.01%	83.68% ± 0.13%
	1449 (100%)	79.36% ± 1.22%	84.86% ± 0.80%
Birds	pretrain only (0%)	–	25.83% ± 5.21%
	50 (<1%)	26.69% ± 4.09%	77.95% ± 1.82%
	100 (∼1%)	44.30% ± 2.08%	81.68% ± 0.99%
	200 (∼2%)	53.42% ± 2.12%	87.35% ± 0.92%
	4000 (∼39%)	95.12% ± 1.06%	97.48% ± 0.97%
	10345 (100%)	95.83% ± 1.09%	97.78% ± 0.09%
Cars	pretrain only (0%)	–	30.51% ± 0.32%
	50 (∼1%)	43.28% ± 0.67%	98.83% ± 0.17%
	100 (∼1%)	52.06% ± 1.43%	99.05% ± 0.10%
	200 (∼3%)	58.71% ± 1.70%	99.34% ± 0.11%
	4000 (∼56%)	97.86% ± 0.95%	99.80% ± 0.03%
	7144 (100%)	98.42% ± 0.28%	99.83% ± 0.01%

Figure 3 shows AP values of the final trained models on the test sets as we increase the size of the subset of real images. Tables I and II detail numerical values for some subset sizes, for the MobileNet and ResNet50 backbones, respectively.

As we can see, the models that were initially pretrained on artificial samples achieved either comparable or superior results for all quantities of real samples and on both architectures. We notice that, depending on the object and architecture, the non-pretrained models can close the gap if enough real samples are used, especially with the ResNet50 backbone. But most importantly for our purposes, this advantage is significantly larger, and always present, when considering very few samples. These results support our hypothesis that initializing the models by pretraining them on artificial samples leads to a more data efficient training.

D. Ablation Experiments

1) Importance of Pretraining Initialization: The first, and most important ablation experiment, compared the proposed strategy of pretraining + finetuning with the common approach of training on real and artificial samples mixed together, as done for instance in [7]. For this, we can not use an infinite stream of artificial samples, as that would drown out any effect from the finite real data. Therefore, we fixed 100 real samples while varying the proportion of artificial ones, as the influence of pretraining is more noticeable on these low quantities of real samples.

We trained the models using our pretraining + finetuning strategy and another using the whole mixed set, for each proportion of artificial samples. For fairness, and to compensate

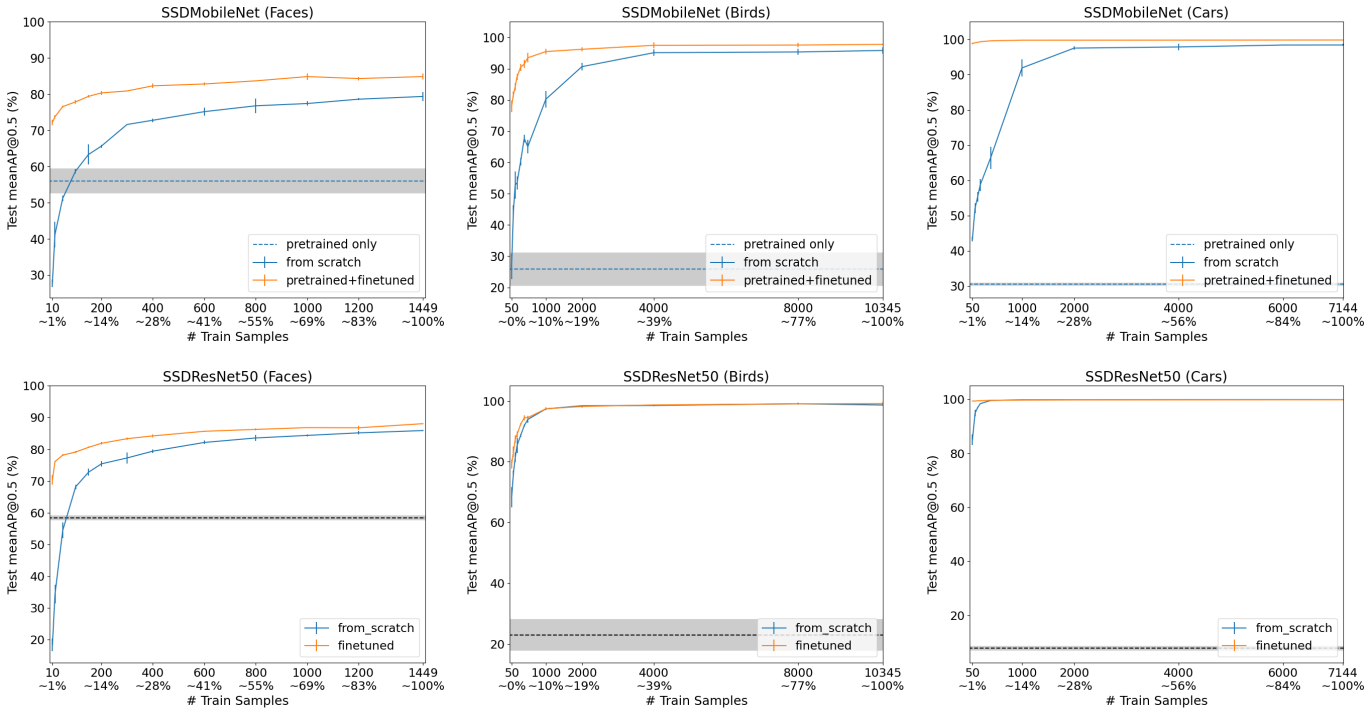

Fig. 3. Average Precision on the test sets for models with (orange) vs without (blue) pretraining on artificial samples, after being finetuned on varying numbers of real samples. Values are averages (± deviations) over three independent runs. The horizontal dashed line is the average (± deviation) AP of pretrained models right before finetuning. Top row: SSD detector with MobileNet backbone. Bottom row: SSD detector with ResNet50 backbone.

TABLE II
NUMERICAL RESULTS FOR A FEW OF THE SSD RESNET50 MODELS FROM FIGURE 3 (BOTTOM ROW).

	# real samples	no pretraining	with pretraining
Faces	pretrain only (0%)	–	58.40% ± 0.66%
	10 (∼1%)	18.33% ± 2.01%	70.38% ± 1.53%
	100 (∼7%)	68.15% ± 0.72%	79.14% ± 0.25%
	200 (∼14%)	75.40% ± 0.88%	81.88% ± 0.41%
	800 (∼55%)	83.57% ± 0.92%	86.26% ± 0.39%
	1449 (100%)	85.90% ± 0.13%	88.06% ± 0.11%
Birds	pretrain only (0%)	–	22.98% ± 5.09%
	50 (<1%)	68.28% ± 3.34%	79.20% ± 1.43%
	100 (∼1%)	76.35% ± 0.78%	83.30% ± 1.74%
	200 (∼2%)	85.36% ± 2.49%	89.01% ± 1.05%
	4000 (∼39%)	98.51% ± 0.25%	98.70% ± 0.07%
	10345 (100%)	98.67% ± 0.28%	99.14% ± 0.55%
Cars	pretrain only (0%)	–	7.82% ± 0.72%
	50 (∼1%)	85.11% ± 1.98%	99.35% ± 0.02%
	100 (∼1%)	94.92% ± 1.26%	99.36% ± 0.07%
	200 (∼3%)	98.39% ± 0.17%	99.46% ± 0.00%
	4000 (∼56%)	99.86% ± 0.01%	99.79% ± 0.02%
	7144 (100%)	99.87% ± 0.01%	99.82% ± 0.01%

for an eventual "warm-up" effect in the pretraining case, we trained the mixed data models for the sum of the number of iterations in the pretraining and finetuning: 1000 steps for Faces/Cars and 2000 steps for Birds. For each proportion of artificial samples, we used the same artificial and 100 real samples for both pretraining and mixed data cases. We again repeated the training of each model version three times and report means and standard deviations, measured an AP@0.5, following [12]. Results are shown in Tables III and IV.

In all cases, pretraining followed by finetuning gives better results than training on real and artificial data mixed together. But regardless of that, both options were always either matched or surpassed by pretraining on the infinite stream of artificial samples.

2) Importance of Unsupervised Segmentation: The next experiment aimed at evaluating the importance of properly segmenting the object instances before pasting them on the background scenes. That is, we tried to analyse the influence of the unsupervised ReDO segmentation step [9]. For this, we created another stream of artificial samples, without using the segmentation step, but instead pasting the whole generated image frames on the background images, and considering all of it to be the bounding boxes. We call this style of samples "Naive Pasting". Detection samples generated using this strategy are shown in Figure 4.

We trained a set of models following our pretraining initialization strategy, but using these naive artificial samples, and compared them against the models trained in the main experiments (Tables I and II). We again used 100 real samples for finetuning, and report results on the test sets. Again, we repeated these experiments three times for each model configuration. Results are presented in Table V.

TABLE III
MIXED REAL AND ARTIFICIAL SAMPLES VS PRETRAINING + FINETUNING, FOR VARYING PROPORTIONS OF ARTIFICIAL SAMPLES, FOR THE SSD MOBILENET ARCHITECTURE.

	# fake samples	mixed data	pretr. + finetune
Faces 100 real samples (∼7%)	100 (1×)	68.67% ± 1.32%	73.36% ± 1.00%
	200 (2×)	68.45% ± 1.87%	74.74% ± 0.16%
	400 (4×)	68.23% ± 0.86%	75.09% ± 0.37%
	800 (8×)	69.11% ± 1.52%	76.29% ± 0.19%
	inf. stream	–	**77.86% ± 0.46%**
Birds 100 real samples (∼1%)	100 (1×)	31.54% ± 4.15%	39.89% ± 1.98%
	200 (2×)	31.53% ± 4.78%	45.39% ± 1.95%
	400 (4×)	31.13% ± 2.69%	45.54% ± 2.11%
	800 (8×)	30.25% ± 2.47%	47.65% ± 1.53%
	inf. stream	–	**81.68% ± 0.99%**
Cars 100 real samples (∼1%)	100 (1×)	76.38% ± 1.01%	91.68% ± 1.87%
	200 (2×)	76.46% ± 2.87%	92.22% ± 1.14%
	400 (4×)	76.26% ± 1.18%	94.50% ± 0.79%
	800 (8×)	79.27% ± 2.13%	95.92% ± 0.35%
	inf. stream	–	**99.05% ± 0.10%**

TABLE IV
MIXED REAL AND ARTIFICIAL SAMPLES VS PRETRAINING + FINETUNING, FOR VARYING PROPORTIONS OF ARTIFICIAL SAMPLES, FOR THE SSD RESNET50 ARCHITECTURE.

	# fake samples	mixed data	pretr. + finetune
Faces 100 real samples (∼7%)	100 (1×)	72.96% ± 0.18%	77.06% ± 0.38%
	200 (2×)	72.59% ± 0.89%	77.56% ± 0.30%
	400 (4×)	73.19% ± 0.26%	77.82% ± 0.30%
	800 (8×)	72.19% ± 0.67%	**79.19% ± 0.30%**
	inf. stream	–	**79.14% ± 0.25%**
Birds 100 real samples (∼1%)	100 (1×)	57.94% ± 0.85%	62.29% ± 2.56%
	200 (2×)	59.49% ± 0.36%	62.52% ± 0.33%
	400 (4×)	58.06% ± 1.78%	66.32% ± 3.52%
	800 (8×)	60.15% ± 0.72%	65.24% ± 3.11%
	inf. stream	–	**83.30% ± 1.74%**
Cars 100 real samples (∼1%)	100 (1×)	96.91% ± 0.06%	98.73% ± 0.13%
	200 (2×)	96.93% ± 0.24%	98.88% ± 0.10%
	400 (4×)	96.79% ± 0.46%	99.08% ± 0.10%
	800 (8×)	96.83% ± 0.11%	98.93% ± 0.02%
	inf. stream	–	**99.36% ± 0.07%**

As we can see, the naive samples already lead to a significant improvement over the non-pretrained models. However, in all cases, they either match, or are outperformed by the models pretrained on samples generated with segmentation, with the largest gap of around 11%, happening for the Birds class on the MobileNet backbone.

These results show that the unsupervised segmentation step is very important for almost all of our cases, although the exact advantage varies widely across dataset and architecture. Further investigation is needed in order to understand which factors are more significant for the final results.

3) Importance of GAN generation: So far, we have opted to use GAN generated object images at the first stage of our synthesis pipeline instead of real ones. This allowed us to generate an infinite stream of artificial samples, where no object instance appears more than once, while also having the convenience of not requiring us to manipulate a classification dataset during pretraining. However, a natural question is whether we could achieve better results by synthesizing samples starting from real object images instead.

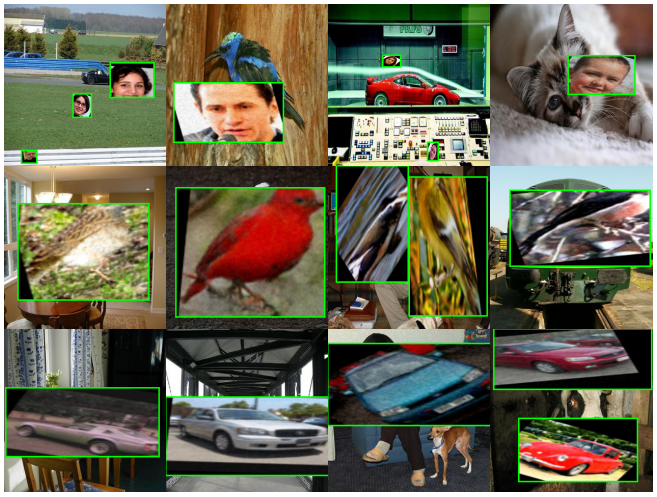

Fig. 4. Artificial detection samples generated without object segmentation (Naive Pasting).

TABLE V
TEST SET RESULTS FOR SSD MODELS WITH MOBILENET (TOP) AND RESNET50 (BOTTOM) BACKBONES, TRAINED WITHOUT PRETRAINING VS PRETRAINED WITH NAIVE SAMPLES VS PRETRAINED WITH OUR REGULAR SAMPLES. EACH MODEL WAS FINETUNED ON 100 REAL SAMPLES.

	without pretraining	pretrained w. naive pasting	pretrained w. segmentation
Faces	58.73% ± 0.62%	75.45% ± 0.41%	77.86% ± 0.46%
Birds	44.30% ± 2.08%	70.58% ± 0.83%	81.68% ± 0.99%
Cars	52.06% ± 1.43%	96.91% ± 1.39%	99.05% ± 0.10%

	without pretraining	pretrained w. naive pasting	pretrained w. segmentation
Faces	68.15% ± 0.72%	76.49% ± 0.18%	79.14% ± 0.25%
Birds	76.35% ± 0.78%	81.41% ± 0.16%	83.30% ± 1.74%
Cars	94.92% ± 1.26%	99.06% ± 0.10%	99.36% ± 0.07%

To answer this question, we created another stream of artificial samples, using object images from the datasets that were used to train the GANs used for the main experiments, namely the FFHQ faces [15], non test samples from the CUB-200-2011 dataset [19] for birds, and LSUN Cars [20]. Next, we trained a set of models following our pretraining strategy over this new stream of artificial samples, and compared them against the results from the main experiments. We again used 100 real samples for finetuning, and report results on the test sets. Once again, we repeated these experiments three times for each model configuration. Results are presented in Table VI.

As the results demonstrate, there is no apparent loss in

TABLE VI
TEST SET RESULTS FOR SSD MODELS WITH MOBILENET (TOP) AND RESNET50 (BOTTOM) BACKBONES, TRAINED WITHOUT PRETRAINING VS PRETRAINED ON ARTIFICIAL SAMPLES BASED ON REAL OBJECT IMAGES VS PRETRAINED ON ARTIFICIAL SAMPLES BASED ON GAN GENERATED IMAGES. EACH MODEL WAS FINETUNED ON 100 REAL SAMPLES.

	without pretraining	pretrained (real object images)	pretrained (GAN generated object images)
Faces	58.73% ± 0.62%	77.63% ± 0.15%	77.86% ± 0.46%
Birds	44.30% ± 2.08%	82.87% ± 1.64%	81.68% ± 0.99%
Cars	52.06% ± 1.43%	98.91% ± 0.15%	99.05% ± 0.10%

	without pretraining	pretrained (real object images)	pretrained (GAN generated object images)
Faces	68.15% ± 0.72%	78.67% ± 0.23%	79.14% ± 0.25%
Birds	76.35% ± 0.78%	84.87% ± 0.54%	83.30% ± 1.74%
Cars	94.92% ± 1.26%	99.22% ± 0.14%	99.36% ± 0.07%

detection quality by using artificial samples composed from GAN generated images. We expect that, as unsupervised segmentation techniques improve and generative image techniques become more data efficient, the advantage of these infinite stream formulation will become more evident, and therefore, artificial samples pretraining will be able to deal with even more extreme low data situations.

E. Experiments on WIDER Face

Finally, we also evaluated our strategy on a more challenging scenario, using a more advanced model. We performed experiments on the WIDER Face dataset [27], a state of the art benchmark for face detection, using the RetinaFace detector [28], a modern architecture designed specifically for face detection[1]. At the time of this writing, RetinaFace was achieving state of the art results on WIDER. Pretraining was done using artificial Face samples generated as described previously, and finetuning was done with varying quantities of real samples.

Results are shown in Table VII, and as can be seen, the models that were pretrained achieved superior results, with the advantage again being larger on very few real samples (even on the harder detection subset), with a gap of $\sim 10\%$ for 100 real samples. This further supports our hypothesis that pretraining reduces the need for bounding boxes.

We note, however, a consistent disadvantage of our pretraining initialization approach as we include all the real data available. Further investigation is needed in order to understand what influence and biases are brought in by artificial samples when real data is already abundant.

V. DISCUSSIONS

The experimental results have shown that artificial samples are a promising direction towards better sample efficient detection models. However, in the presented formulation, our pipeline still has some limitations.

[1]We used the implementation provided in github.com/biubug6/Pytorch_Retinaface

TABLE VII
WIDER [27] VALIDATION RESULTS FOR A RETINAFACE [28] WITH VS WITHOUT ARTIFICIAL SAMPLES PRETRAINING. (SINGLE RUN)

		Easy	Medium	Hard
50 real samples	w/o pretrain	40.76%	38.35%	31.22%
	w/ pretrain	49.71%	44.39%	37.00%
100 real samples	w/o pretrain	45.5%	42.57%	33.36%
	w/ pretrain	59.34%	56.40%	43.19%
250 real samples	w/o pretrain	58.65%	53.99%	45.39%
	w/ pretrain	63.21%	58.80%	49.41%
1000 real samples	w/o pretrain	70.02%	67.30%	56.88%
	w/ pretrain	72.65%	69.24%	58.35%
all real samples	w/o pretrain	83.14%	79.36%	69.65%
	w/ pretrain	82.15%	77.78%	67.26%

First of all, our method still requires a significant dataset of classification images from objects of interest, either to be used directly or to be used for training a GAN. We expect that, as generative image models improve in terms of sample efficiency, this requirement will eventually be relaxed. A recent improvement in this regard comes from [29].

Second, the fact that we have a "pipeline" of steps, with GAN based generation, unsupervised segmentation, and random pasting, can be a significant source of noise on the generated samples. Any development which succeeds in combining these steps into an end-to-end architecture can potentially improve the detection sample generation process.

Despite these limitations, the experiments have provided evidence that our pretraining initialization strategy is a promising way of taking advantage of artificial samples. We expect this strategy to benefit from any improvement on the above listed limitations.

VI. CONCLUSION

In this work, we showed how it is possible to generate artificial labeled detection samples starting from only classification level supervision, by using a simple combination of already existing techniques. Additionally, we proposed using these artificial samples to pretrain detection models before finetuning them on real labeled data. With this approach, using only a small fraction of the available real bounding box annotated data for finetuning, we obtained detection performance on par with those achieved by the models trained on the whole real data. Therefore, we effectively managed to take advantage of the cheap and abundant Classification data in order to achieve competitive results on the harder and more supervision hungry Detection problem.

As a final note, the performance gap between pretrained-only (the horizontal dashed lines in Figure 3) and pretrained + finetuned models indicates the importance of model finetuning with real data. We expect that, as generative image models such as GANs continue to improve, and it becomes possible

to generate increasingly realistic artificial samples, the need for real data will be further reduced.

ACKNOWLEDGMENT

This work was partly funded by the São Paulo Research Foundation (FAPESP) under grants 2017/25835-9 and 2015/22308-2, and by the EU Horizon 2020 RISE Project ULTRACEPT under Grant 778062. L. Blanger received support from FAPESP (grants 2018/00390-7 and 2019/17312-1).

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[2] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," *preprint arXiv:1711.04340*, 2017.

[3] S. Yamaguchi, S. Kanai, and T. Eda, "Effective Data Augmentation with Multi-Domain Learning GANs," in *AAAI Conference on Artificial Intelligence*, 2020.

[4] O. Bailo, D. Ham, and Y. Min Shin, "Red blood cell image generation for data augmentation using conditional generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[5] V. Sandfort, K. Yan, P. J. Pickhardt, and R. M. Summers, "Data Augmentation using Generative Adversarial Networks (CycleGAN) to improve Generalizability in CT Segmentation Tasks," *Scientific Reports*, 2019.

[6] S. Milz, T. Rudiger, and S. Suss, "Aerial GANeration: Towards Realistic Data Augmentation using Conditional GANs," in *European Conference on Computer Vision (ECCV)*, 2018.

[7] N. Dvornik, J. Mairal, and C. Schmid, "Modeling visual context is key to augmenting object detection datasets," in *European Conference on Computer Vision (ECCV)*, September 2018.

[8] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *IEEE International Conference on Computer Vision*, 2017.

[9] M. Chen, T. Artières, and L. Denoyer, "Unsupervised object segmentation by redrawing," in *Advances in Neural Information Processing Systems*, 2019.

[10] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On pre-trained image features and synthetic images for deep learning," in *European Conference on Computer Vision (ECCV)*, 2018.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[13] A. Kanezaki, "Unsupervised image segmentation by backpropagation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

[14] R. Arandjelović and A. Zisserman, "Object discovery with a copy-pasting GAN," *preprint arXiv:1905.11369*, 2019.

[15] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[16] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," U. of Massachusetts Amherst, Tech. Rep., 2007.

[17] V. Jain and E. Learned-Miller, "FDDB: A Benchmark for Face Detection in Unconstrained Settings," U. of Massachusetts, Amherst, Tech. Rep., 2010.

[18] M. Zhu, P. Pan, W. Chen, and Y. Yang, "DM-GAN: Dynamic Memory Generative Adversarial Networks for text-to-image Synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[19] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep., 2011.

[20] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a Large-Scale Image Dataset using Deep Learning with Humans in the Loop," *preprint arXiv:1506.03365*, 2015.

[21] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D Object Representations for Fine-Grained Categorization," in *International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," in *European Conference on Computer Vision*, 2016.

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *preprint arXiv:1704.04861*, 2017.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[27] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A Face Detection Benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[28] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *arXiv preprint arXiv:1905.00641*, 2019.

[29] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable Augmentation for Data-Efficient GAN Training," *Advances in Neural Information Processing Systems*, vol. 33, 2020.