Image Synthesis Based Neural Network Pretraining for Object Detection

Leonardo Blanger
Institute of Mathematics and Statistics
University of São Paulo
São Paulo, Brazil
Email: lblanger@ime.usp.br

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

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

***Abstract*—This work proposes an initialization strategy for Object Detection models based on pretraining using synthesized samples. In order to demonstrate the feasibility of this initialization strategy, we design a simple synthesis pipeline using existing generative models and a recently proposed unsupervised segmentation technique. This pipeline effectively managed to generate an infinite stream of useful synthesized annotated detection samples, and it can be trained using only classification level images, without requiring expensive forms of supervision like bounding boxes or masks. We experimentally demonstrate that this sample synthesis based initialization allows us to take advantage of existing classification datasets to reduce the need for the costlier real labeled detection images. We managed to achieve comparable results in Object Detection tasks using less than 10% of the initial bounding box labeled images. The code to reproduce these experiments will be publicly released.**

I. Introduction

Object detection techniques have considerably evolved in the past few years [1], [2], [3], [4], [5]. Since the adoption of CNN based architectures [1], [2], the state of the art performance on standard detection benchmarks such as Pascal VOC [6] and MS-COCO [7] has consistently increased [8], [9]. However, these Deep Learning based detection models are known to require huge amounts of labeled data in order to achieve their best performance. Therefore, despite their successes on big standardized datasets, it is still challenging to deploy recent detection models on domains with low availability of labeled data.

Acquiring labeled data for object detection is costly. One needs to identify not only object categories, but also rectangular, axis-aligned bounding boxes [6], making the labeling effort harder, more time consuming, and more prone to mistakes and human biases than in the classification case.

In contrast, acquiring labeled data for image classification is much cheaper and faster, and public datasets for classification [10] tend to be orders of magnitude bigger than their detection counterparts. In this perspective, we wonder if it is possible to transfer knowledge from other domains that require more high level and less costly supervision, such as image classification, into the object detection task, in a way that alleviates the need for detection supervision.

In parallel, Deep Learning techniques have also allowed recent improvements in the field of Generative Image Models [11], [12], [13], [14], with the most representative techniques being currently instances of Generative Adversarial Networks (GANs) [15], Variational Auto Encoders (VAEs) [16], and Autoregressive Models [17]. On some domains, recent versions of these techniques can even generate images close to indistinguishable from real ones [12], [13].

Leveraging the knowledge encoded by generative models about the data distribution to improve downstream tasks is an important machine learning goal. Some works have explored the use of generative models for data synthesis in the classification task [18], [19], [20], [21], [22], [23], [24]. However, due to the current limitations of generative models, it is difficult to generate labeled images suitable for training object detection models. The few existing attempts require some low level form of supervision like bounding boxes [25], [26], mask annotations [27], [28], [29], or key points for training [25], [27]. This prevents image synthesis from being a viable option for expanding already small object detection datasets.

In this work, we propose a pretraining strategy that provides a better initialization for deep learning based detection models, thus achieving comparable results with less need for labeled data. This pretraining is performed on synthesized images that are created with the help of generative models trained only with classification level annotations. In order to demonstrate the feasibility of this initialization strategy, we design a simple synthesis pipeline by combining already existing "of-the-shelf" methods. To the best of our knowledge, this is the first attempt at synthesizing detection samples without requiring bounding box annotations for training.

The contributions of this work are, in order of importance: (1) we propose an initialization strategy for Object Detection models based on pretraining on synthesized samples, and experimentally show that such initialization allows comparable results with only a fraction of real labeled detection data in a series of domains; and (2), we demonstrate that it is possible to perform such synthesis **without bounding boxes** or other expensive supervision, by designing a simple synthesis

Project conducted when Leonardo Blanger was a research intern at the University of Münster.

Manuscript 2052 submitted to 2020 25th International Conference on Pattern Recognition (ICPR). Received July 13, 2020.

pipeline using already existing techniques, that can be trained with only classification level images.

We provide a review of related works (Section II), followed by a description of the synthesis pipeline we adopted (Section III) and a series of experiments to investigate the advantages of this initialization strategy on different scenarios (Section IV). We conclude with a discussion about the advantages and limitations of our method (Section V).

II. Related Work

Approaches to expand existing datasets by artificially generating additional samples have played an important role in Computer Vision so far. Traditionally, these approaches fall under the class of Data Augmentation techniques [30]. Most applications of data augmentation on computer vision consist of simple, manually designed, label preserving image transformations, such as random color and geometric manipulations [30]. More recently, however, some works attempted to learn data augmentation strategies directly from data, either by learning augmentation policies from sets of previously designed simple image operations [31], [32], [33], or by learning to synthesize novel samples from scratch [18], [19], [20], [21], [23], [22], [24], [34], [35], [36], [37], [38].

Many works attempted to apply generative image models (mostly GANs) to synthesize samples to expand image classification datasets [18], [19], [20], [21], and this constitutes a promising research direction, specially on domains like medical imaging [23], where labeled data is costly to acquire, or situations with underrepresented classes [22], [24].

Some works tried to apply GAN based synthesis for data augmentation on tasks beyond classification, such as detection and segmentation on medical [34], [35], [36] and aerial images [37], as well as eye gaze and hand pose estimation [38]. However, these approaches require costly forms of supervision in order to train the GANs, like segmentation masks [34], [35], [37], paired images from different domains [36], or manually designed simulators to aid the generation process [38].

The main challenge of sample synthesis for object detection is the requirement to generate granular forms of supervision, like bounding boxes, along with the images. Current generative models perform very well on single, centralized and full image objects [11], [12], [13], [14], but struggle to generate images with multiple objects of varying scales and geometric configurations interacting in complex scenes, while at the same time generating coherent bounding box labels. Existing attempts are either constrained to simplified artificial domains [39], or require costly forms of supervision like bounding boxes [25], [26], segmentation masks [27], [28], [29], or key points [25], [27] for training.

A more feasible approach would be to start from a real background image, and then generate objects coherently in it, as done in [40] and [41], although these methods still require images labeled with bounding boxes and segmentation masks for training, respectively. An even simpler approach would be to generate objects independently, and then placing them on regions of real images. This strategy allows the recycling of already pretrained generic generative models. However, in order to properly crop the object from their generated frame, some form of segmentation is necessary, which up until recently would also require mask level annotations for training.

Similar approaches are taken by [42], that uses a segmentation network trained with mask annotations to synthesize samples for Instance Detection, a problem in which the goal is to detect individual instances within object classes, and by [43], that trains a traffic sign detection model on a synthesized dataset, built by pasting predefined sign templates onto background images. The successful results in [42] and [43] suggest that, if done in an unsupervised way, sample synthesis would be a viable option for generic Object Detection as well.

In this work, we show that it is now possible to use existing generative image models to perform sample synthesis for Object Detection. In particular, we show that a recently proposed unsupervised segmentation technique [44] can be combined with traditional generative image models to synthesize an infinite stream of artificial labeled detection samples, which can then be used to pretrain Object Detection models.

As existing generative models can be trained with classification data only, being capable of combining them to synthesize detection samples allows us to effectively transfer the knowledge encoded in the larger and cheaper classification data, into the harder and more supervision hungry Object Detection problem.

We are avoiding to classify our method as "Image Augmentation", as we are not augmenting existing detection datasets, but rather transferring knowledge acquired from other datasets with a higher level form of supervision into the object detection task. In this perspective, our approach of pretraining on synthesized samples draws some inspiration from recent advances in Self-Supervised pretraining, in which a model is first trained using large bodies of unlabeled data on pretext-tasks like predicting rotations [45] or patch relationships [46], colorization [47], or inpainting [48], and then fine-tuned or used as feature extractors for downstream tasks.

III. Synthesis Pipeline

We highlight here that our main contribution is in demonstrating that sample synthesis based initialization allows us to take advantage of big amounts of classification images in order to reduce the need for real labeled detection data. We do not claim this pipeline to be the best approach to perform this synthesis, we merely tried to combine already existing techniques in a way that does not require bounding boxes or other expensive supervision, in order to show that this form of synthesis is possible. Further improvements on unsupervised segmentation and generative models in general could uncover better ways to perform this task.

That being said, the synthesis pipeline we used can be structured as a sequence of tasks. Figure 1 presents an overview of the whole pipeline. This section describes each of these steps.

A. Gathering Image Samples

The first step consists in gathering images of a given object class. These are traditional centralized simple object images,

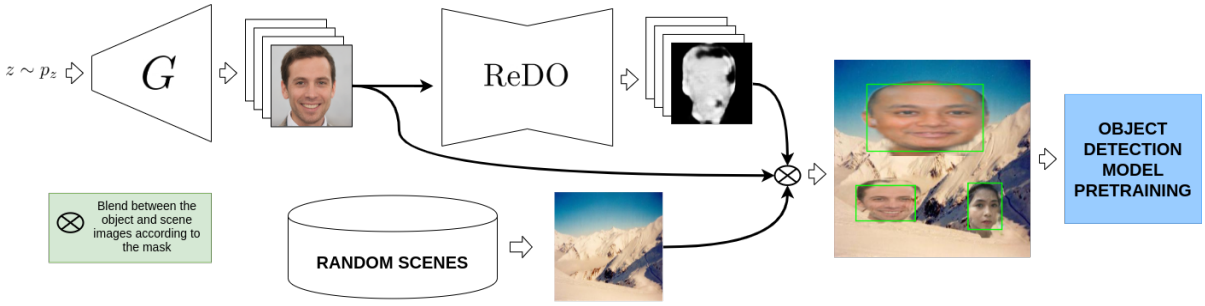


Fig. 1: Overview of our Object Detection sample synthesis. In this case, G is a StyleGAN face generator [12].

as used for classification tasks, and can be either real classification images, or fake images produced by traditional generative models, that are trained on such classification level data. In the Ablations section, we demonstrate that both approaches achieve equivalent results.

B. Unsupervised RoI Segmentation

The next step is to segment the object region of interest in these gathered images. As we aim to exploit image level classes only, we are restricted to segmentation techniques that can be trained without mask annotations, ruling out most of the deep learning based segmentation literature [49], [50]. In this work, we opted for the recent *Object Segmentation by Redrawing* (ReDO) method proposed by Chen et al. [44], that is trained with only classification level annotations.

The ReDO framework consists of a network with a branch responsible for segmenting images into a set of disjoint objects + background, with another branch that generates fake objects on these segmented regions. This network is trained adversarially against a discriminator that tries to tell real images apart from the ones that were segmented and redrawn, following the GAN paradigm [15].

Unsupervised segmentation is still far from achieving the same results as traditional supervised approaches, and ReDO is a considerably recent technique, with a proven applicability on only a few domains. Therefore, our synthesized samples are of noticeable low quality. Nonetheless, we experimentally show how a large amount of them can significantly improve detection performance when limited labeled data is available.

We chose the ReDO method as it fits nicely in a pipelined structure. Its end result is not concerned with generating object images along with masks as is done in [51] (although this can be achieved using the redrawing branch), but rather with taking images as input and segmenting them. This allows us to use it in combination with generative models from the previous step, specifically optimized for image quality.

C. Object Detection Sample Synthesis

Once we have classification images of a given object class paired with their segmentation masks, we can create detection samples by placing the segmented object regions on top of random (real or generated) scenes, at varying number and at random positions and scales. From the masks, we can easily extract the bounding box parameters.

At this point, we advocate for using fake generated images instead of real ones in the first stage. This way, once the models from the first two stages are trained, we can effectively generate an infinite stream of synthesized images with bounding box annotations without needing to load the classification data into memory again, and in practice, never repeat the same object instance twice. We also point to the fact that the only supervision needed for the previous steps is at the classification level, which allows us to potentially exploit much larger datasets than what is common for detection.

D. Object Detection Pretraining

Finally, we can use this infinite stream of synthesized samples to pretrain object detection models. We demonstrate in Section IV that initializing a detection model using this pretraining strategy allows us to achieve comparable results with a significantly smaller amount of real labeled detection images. We also provide some justification for this choice on how to use the synthesized samples in Section IV-C.

IV. Experiments

A. Datasets

We conducted a series of experiments on three, single class domains, in order to demonstrate the major benefits of our method. For the main experiments, we adopted the approach of generating fake image samples for the first stage. We considered the following object classes:

- **Faces.** We used the *Face Detection Data Set and Benchmark* (FDDB) [52], that contains 5171 faces across 2845 images that are divided in 10 folds. We used the first five folds for training, the next two for validation and the last three for testing, making for a split of 1449/581/815 images. We converted the face ellipsis to rectangular bounding boxes[1]. To generate fake samples, we used StyleGAN faces [12], with weights from the FFHQ dataset [12], and the ReDO segmentation branch with weights from the *Labeled Faces in the Wild* dataset (LFW) [53], [54], provided by [44][2].
- **Birds.** We used the *Caltech-UCSD Birds-200-2011* (CUB) dataset [55], that contains 11788 images, each

[1]We used the script provided in github.com/ankanbansal/fddb-for-yolo
[2]github.com/mickaelChen/ReDO

Fig. 2: Synthesized detection samples on faces (left), birds (middle) and cars (right). The composition is not perfect. By zooming in, we can see some irregular contours and unrealistic transparencies. These mostly come from the challenging task of unsupervised segmentation. Although not having human-level realism, we show that a high number of these easily created samples can significantly alleviate the need for labeled data. Best viewed in color.

with a single bird of varying scale. We used a DM-GAN [56] generator to generate the fake birds and a ReDO [44] segmentation branch provided by [44], both of them trained on the bounding box crops of the CUB dataset itself. As neither the DM-GAN [56] nor the ReDO segmentation model [44] were trained following the official CUB train/test splits [55], in order to prevent leakage of test information on the first two stages of the pipeline, we only used the intersection between the test sets from [24] and [44] as our test set, totaling 443 images. We additionally picked 1000 images from the remaining ones as validation, making for a 10345/1000/443 split.

- **Cars.** We used the Stanford Cars dataset [57], which contains 16185 images, each with a single car of varying scale. We picked 1000 images from the official train split to use as validation, making for a 7144/1000/8041 split. To generate fake samples, we used StyleGAN cars [12], with weights from the LSUN Cars dataset [58]. As [44] did not provide weights for cars, we trained a full ReDO model on the fake images just mentioned using the same model configuration as for the faces.

For all domains, we applied random color and geometric augmentations to the fake images, with the geometric operations being mirrored on the respective masks. For simplicity, we used the negative set of the INRIA Person dataset [59] as random background scenes for all three domains. Figure 2 shows some examples of synthesized detection images.

Note that the synthesized samples are far from realistic (faces in the sky and huge birds on top of bicycles do not happen in normal situations). Nonetheless, we experimentally show that a large amount of them are still useful for reducing the need for labeled data, and with this simple formulation, we do not need any complicated hand designed simulators or model of object context, that would probably depend on yet more expensive supervision.

We chose these three simple domains due to them being at a level of difficulty the ReDO segmentation [44] could handle. Despite being now a realistic possibility, unsupervised segmentation is still a very challenging field. Additionally, we opted for these generative models [12], [56], and by extent, their training data, due to their popularity, and their available pretrained versions for these domains.

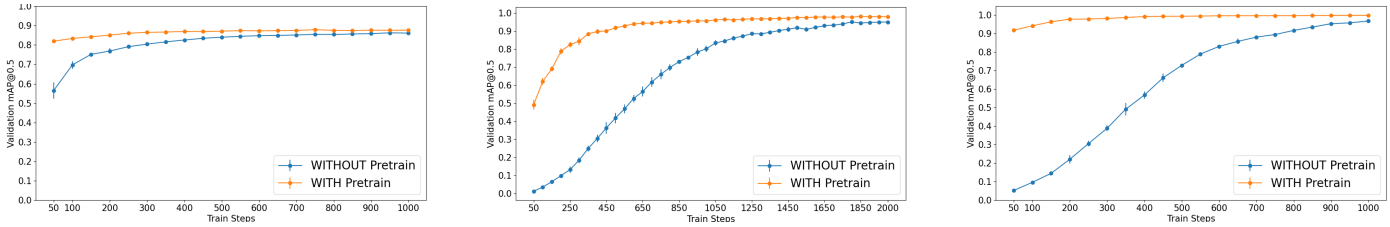
B. Main Results

We used an SSD detector [5] with an ImageNet [10] pretrained MobileNet backbone [60], frequently adopted for low data domains. For all experiments, we used a batch size of 32, and Adam optimizer [61] with learning rate 10^{-4} and $\beta = (0.9, 0.999)$. We report results in terms of mAP at 0.5 IoU, following the Pascal VOC metric [6]. The code to reproduce these experiments will be publicly released.

The models that include our pretraining strategy were first trained on synthesized samples for 1000 steps for faces and cars, and 2000 steps for birds, as we observed the validation results plateau by these iterations. We checkpointed the models every 50 steps, and use the one with the best validation mAP as the final pretrained model. The models without pretraining have traditional initialization: ImageNet [10] classification weights for the backbone and uniform Glorot/Xavier initialization [62] on the detection specific layers.

First, we trained a model using all the real data available for each domain, with and without our pretraining strategy, again for 1000 steps for faces and cars, and 2000 steps for birds. Figure 3 shows the results on the validation set at every 50 training steps. As we can see, these numbers of steps are enough to saturate the results even in this case when using all the data. We also observe that models that were pretrained achieve a better final validation error, with a faster convergence and more stable behaviour across different runs (shorter deviation bars) for all three domains.

Next, we trained a series of models on varying amounts of real data with and without our pretraining strategy, and report the results of the final models on the test set. For each number of images, we sampled them at random from the whole dataset, and used the same set of images for both the model with and without pretraining. We trained all models during the same number of steps as before. Results for the final model



(a) Faces (b) Birds (c) Cars

Fig. 3: Results on the validation sets in mAP@0.5 at every 50 steps when training with all the data. Each point is the average with deviation bar from three independent runs with the same training configurations.

	# real samples	without pretraining	with pretraining
faces	pretrain only (0%)	–	75.42% ± 1.91%
	100 (∼7%)	71.87% ± 0.66%	80.42% ± 0.69%
	800 (∼55%)	83.28% ± 0.53%	86.15% ± 0.50%
	1449 (100%)	85.83% ± 0.48%	87.34% ± 0.16%
birds	pretrain only (0%)	–	26.25% ± 0.95%
	1000 (∼10%)	41.08% ± 0.89%	93.67% ± 0.62%
	5000 (∼48%)	91.99% ± 0.96%	97.07% ± 0.30%
	10345 (100%)	93.24% ± 0.42%	97.32% ± 0.27%
cars	pretrain only (0%)	–	41.18% ± 2.22%
	500 (∼7%)	46.25% ± 0.56%	98.70% ± 0.13%
	3000 (∼42%)	95.50% ± 0.59%	99.65% ± 0.10%
	7144 (100%)	96.41% ± 0.60%	99.73% ± 0.04%

TABLE I: Test set performance in mAP@0.5 of models trained with a few representative numbers of real samples. Each value is the average with deviation from three independent runs with the same training configurations.

	# fake samples	mixed data	pretr.+finetune
faces 100 real samples (∼7%)	100 (1×)	76.84% ± 0.48%	76.55% ± 0.27%
	200 (2×)	78.09% ± 0.18%	77.23% ± 1.26%
	400 (4×)	78.93% ± 0.75%	78.25% ± 0.86%
	800 (8×)	79.41% ± 0.36%	78.85% ± 0.44%
	inf. stream	–	**80.42% ± 0.69%**
birds 1000 real samples (∼10%)	1000 (1×)	81.18% ± 3.14%	80.30% ± 2.30%
	2000 (2×)	83.36% ± 1.72%	84.00% ± 1.33%
	4000 (4×)	83.05% ± 0.75%	91.09% ± 0.66%
	8000 (8×)	82.40% ± 1.26%	93.03% ± 0.20%
	inf. stream	–	**93.67% ± 0.62%**
cars 500 real samples (∼7%)	500 (1×)	88.60% ± 0.96%	87.04% ± 2.22%
	1000 (2×)	94.25% ± 1.24%	92.28% ± 2.04%
	2000 (4×)	96.82% ± 0.73%	97.28% ± 0.16%
	4000 (8×)	97.50% ± 0.29%	98.33% ± 0.10%
	inf. stream	–	**98.70% ± 0.13%**

TABLE II: Test set performance in mAP@0.5 of models trained with varying amounts of synthesized samples, using a mixed data training session vs our pretraining strategy. Results are averages with deviations from three independent runs with the same training configurations.

versions are shown in Figure 4. Table I presents results for some representative numbers of samples.

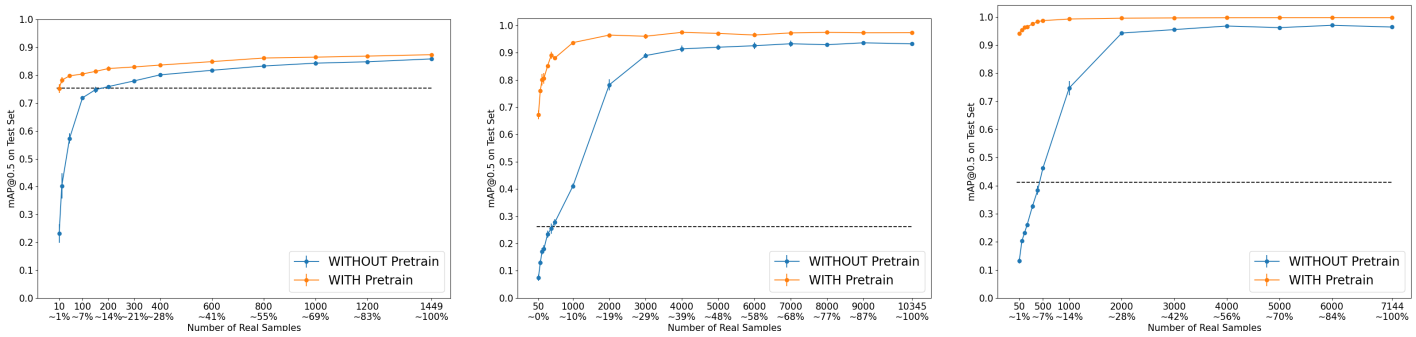
When using all the data, the pretrained models achieve a slightly but consistently better result. Additionally, the advantage of pretraining is more noticeable on the low data regime. As can be seen, the models that were previously pretrained achieve comparable results to the full-data non-pretrained ones when using less than 10% of the data (Table I), therefore supporting our hypothesis that synthesized samples reduce the need for real labeled data.

C. Ablation Studies

Importance of finetuning on real data. We evaluated these models on the test set after pretraining and before finetuning on real data. Results are shown in Figure 4 (horizontal dashed lines) and Table I (first row of each block). As can be seen, the pretraining initialization alone performs poorly, and in the case of birds and cars, it does not even breaks the 50% mark. This is evidence that the observed results do not come solely from the pretraining initialization, and that finetuning on real samples is still necessary. We believe this is the case due to the difference between the real and synthesized distributions. Although the pretraining is able to extract useful knowledge from the synthesized samples, like the overall structure and frequent textures of objects, it is not enough to completely bridge the gap with the real world data.

Advantage of pretraining. Next, to show the advantage of pretraining instead of simply training on mixed real and synthesized data, we trained a few models using a small proportion of real samples in each dataset, using a single mixed data train session vs our pretraining + finetuning strategy. For fairness, and to compensate for eventual "warm-up" effects in the pretraining case, we trained the mixed data models for the sum of the number of iterations in the pretraining and finetuning: 2000 steps for faces and cars and 4000 steps for birds. As we can not use an infinite stream of synthesized samples in the mixed data case, otherwise the effect of real data would disappear, we evaluated different proportions of synthesized samples. For each proportion, we used the same synthesized samples for both the pretraining and the mixed data cases. Results are shown in Table II.

For the faces, the mixed data performed slightly better than pretraining. For birds, the mixed data scores better with less synthesized samples, and gets worse as more are used, while the pretrained version performs increasingly better. For cars, the mixed data starts better, and improves slowly, being surpassed by the pretrained version. We believe the behaviour here depends on the quality of synthesized samples (faces and



(a) Faces (b) Birds (c) Cars

Fig. 4: Results on the test set in mAP@0.5 for models trained with varying amounts of real data, with and without our pretraining strategy. Each point is the average with deviation bar from three independent runs with the same training configurations. The dashed horizontal line represents the average test set mAP of the pretrained model before finetuning it on real data.

	# real samples	real cls. images	GAN generated images
faces	100 (7%)	79.51% ± 0.63%	80.42% ± 0.69%
birds	1000 (10%)	92.45% ± 0.48%	93.67% ± 0.62%
cars	500 (7%)	98.66% ± 0.13%	98.70% ± 0.13%

TABLE III: Test set results of models trained using real vs GAN generated classification samples. Results are averages with deviations from three independent runs with the same training configuration.

	# real samples	w/o pretrain	w/ pretrain Naive Pasting	w/ pretrain ReDO Seg.
faces	100 (7%)	71.87%	77.28% ± 0.51%	**80.42% ± 0.69%**
birds	1000 (10%)	41.08%	92.01% ± 0.84%	**93.67% ± 0.62%**
cars	500 (7%)	46.25%	62.16% ± 1.23%	**98.70% ± 0.13%**

TABLE IV: Test set results of models trained without pretraining vs pretrained using the naive pasting approach vs pretrained using ReDO [44] segmented samples. Results are averages with deviations from three independent runs with the same training configuration.

cars were significantly better segmented than birds [44]).

Nonetheless, for all three datasets, both versions performed either worse or on par with the infinite stream pretraining. The infinite stream also has practical advantages. Not having to specify the amount of synthesized samples, and not having to worry about sample proportions in batch loading reduces the need for hyper-parameter tuning and thus simplify our setup.

Difference between real and fake classification images. We opted for using GAN generated classification samples in the first stage instead of real ones, as this has the conceptual advantage of allowing an infinite stream of samples that never repeat an object instance. In this perspective, it is natural to ask whether real samples could achieve better results. To that end, we evaluated this same set of models, but replacing the GAN generation step with direct sampling of real images. Results are shown in Table III. As we can see, there is no noticeable decrease in the results for using fake images.

Importance of proper object segmentation. Finally, to show the importance of properly segmenting the objects, we trained a set of models using our pretraining strategy, but with samples synthesized without the ReDO segmentation step. For this, we naively pasted the complete generated fake images on top of the scenes, and used the fake image frame as its bounding box. All other training parameters stay the same. Results are shown in Table IV.

Pretraining with naive pasting performs better than training from scratch, although worse than if pretrained with the segmentation step. This gives evidence that useful knowledge is indeed being transfered to the detection task, but not as precise as when the objects are segmented and the bounding boxes are more accurate. For birds, where the segmentation is the least realistic among the three domains, the gap is considerably small. As for cars, the gap is of 36.54%. We hypothesize this might have happened due to the more uniform backgrounds pasted along with the fake cars (frequently roads), that may have confused the model into misunderstanding which parts of the bounding box region are discriminative factors for cars.

D. Experiments on the WIDER dataset

In order to evaluate our strategy on a more realistic scenario, we performed experiments on the challenging WIDER dataset [63], a state of the art benchmark for face detection. We used the RetinaFace architecture [64], designed specifically for face detection. In addition to the traditional detection outputs, RetinaFace has a key-point estimation head, trained on facial-landmarks that the authors labeled on the WIDER dataset, and a mesh decoder branch trained to predict 3D facial information in a self-supervised way [64]. By the time of this writing, the RetinaFace architecture achieved state of the art results on the hard partition of the WIDER benchmark.

We pretrained a RetinaFace model with MobileNet backbone [60] on our synthesized face samples, turning off the facial landmark output. Next, we finetuned the model on WIDER. Due to computational constraints, we only trained for half the number of epochs (125 instead of 250) and half

		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%
500 real samples	w/o pretrain	67.25%	64.25%	52.75%
	w/ pretrain	66.54%	64.08%	53.09%
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%

TABLE V: Validation performance on the WIDER dataset [63] of RetinaFace [64] models, using different amounts of samples. We used the Pytorch implementation provided in `github.com/biubug6/Pytorch_Retinaface`.

the image size (320 instead of 640). We considered different numbers of real samples. Results are shown in Table V.

Our pretraining strategy improves the results significantly when few real images are available. With 100 images, pretraining gives a boost of more than 10% across all three partitions. This presents further evidence of the potential of our pretraining strategy in little data scenarios. However, the gap quickly diminishes as more images are used, with the pretrained model achieving slightly inferior results when the whole dataset is used, which might indicate that when enough data is available, initializing the model by training it on a related but different data distribution of the same task could potentially harm the results. More investigation is needed to identify the limits of our approach when using big datasets.

V. Discussion

Taking advantage of the knowledge encoded by generative models about the data distribution to improve downstream tasks is an important machine learning goal. At the same time, effectively performing object detection sample synthesis without detection supervision was and still is a challenging task. This work tries to combine existing techniques to overcome these challenges.

Recent advances in unsupervised segmentation [44] have made the above mentioned tasks considerably easier. Although still far from generating perfect segmentation masks, we found it to be the most viable option to perform sample synthesis without needing costly forms of supervision. Besides ReDO, unsupervised segmentation can theoretically be accomplished by "Copy-Pasting" GANs [39]. However, these are still mostly limited to simplified artificial datasets [39].

A second approach that could potentially replace segmentation is to adopt generative models that produce paired objects and masks, as done by the LR-GAN architecture [51]. This has the downside of not allowing traditional "of-the-shelf" generators, and we found the LR-GAN to be more difficult to train than ReDO. Additionally, the final sample quality was visually inferior to that of specialized generators + segmentation. We believe further advances in unsupervised segmentation and generative models could uncover ways to perform sample synthesis for many computer vision tasks.

VI. Conclusion

This work proposes an initialization strategy for Object Detection models, based on pretraining on synthesized samples. We demonstrate that this initialization allows comparable results using only a fraction ($\sim$10%) of the original real labeled data. In order to show that it is possible to perform such synthesis, we design a simple synthesis pipeline that is capable of synthesizing an infinite stream of labeled detection samples, by combining already existing techniques that do not require bounding boxes or other expensive forms of supervision for training. In effect, we managed to take advantage of the bigger and cheaper classification data to reduce the need for the more expensive detection data.

Acknowledgment

Project funded by the São Paulo Research Foundation (FAPESP) under grants 2017/25835-9, 2015/22308-2, 2018/00390-7 and 2019/17312-1.

References

[1] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *Int. Conf. on Learning Representations*, 2014.

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

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conf. on computer vision*. Springer, 2016, pp. 21–37.

[6] 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," *Int. Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conf. on computer vision*. Springer, 2014.

[8] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[9] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, 2019.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255.

[11] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Int. Conf. on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=B1xsqj09Fm

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

[13] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," *arXiv preprint arXiv:1912.04958*, 2019.

[14] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 837–14 847.

[15] 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.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd Int. Conf. on Learning Representations*, 2014.

[17] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *33nd Int. Conf. on Machine Learning*, 2016.

[18] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2018. [Online]. Available: https://openreview.net/forum?id=S1Auv-WRZ

[19] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "Bagan: Data augmentation with balancing gan," *arXiv preprint arXiv:1803.09655*, 2018.

[20] S. Yamaguchi, S. Kanai, and T. Eda, "Effective data augmentation with multi-domain learning gans," *arXiv preprint arXiv:1912.11597*, 2019.

[21] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *preprint arXiv:1712.04621*, 2017.

[22] J. R. T. Pinetz, J. Ruisz, and D. Soukup, "Actual impact of GAN augmentation on CNN classification performance," in *8th Int. Conf. on Pattern Recognition Applications and Methods*, 2019, pp. 15–23.

[23] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

[24] X. Zhu, Y. Liu, J. Li, T. Wan, and Z. Qin, "Emotion classification with data augmentation using generative adversarial networks," in *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*. Springer, 2018.

[25] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Advances in Neural Information Processing Systems*, 2016, pp. 217–225.

[26] T. Hinz, S. Heinrich, and S. Wermter, "Generating multiple objects at spatially distinct locations," in *7th Int. Conf. on Learning Representations*, 2019.

[27] S. Reed, A. van den Oord, N. Kalchbrenner, V. Bapst, M. Botvinick, and N. De Freitas, "Generating interpretable images with controllable structure," 2016.

[28] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *IEEE Conf. on computer vision and pattern recognition*, 2018, pp. 8798–8807.

[29] M. O. Turkoglu, W. Thong, L. Spreeuwers, and B. Kicanaoglu, "A layer-based sequential framework for scene generation with gans," in *AAAI Conf. on Artificial Intelligence*, vol. 33, 2019.

[30] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, p. 60, 2019.

[31] A. J. Ratner, H. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, "Learning to compose domain-specific transformations for data augmentation," in *Advances in Neural Information Processing Systems*, 2017.

[32] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

[33] B. Zoph, E. D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens, and Q. V. Le, "Learning data augmentation strategies for object detection," *arXiv preprint arXiv:1906.11172*, 2019.

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

[35] C. Bowles, L. Chen, R. Guerrero, P. Bentley, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert, "Gan augmentation: Augmenting training data using generative adversarial networks," *arXiv preprint arXiv:1810.10863*, 2018.

[36] 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*, no. 1, 2019.

[37] S. Milz, T. Rudiger, and S. Suss, "Aerial ganeration: Towards realistic data augmentation using conditional gans," in *European Conf. on Computer Vision*, 2018, pp. 0–0.

[38] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *IEEE C. on Computer Vision and Pattern Recognition*, 2017.

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

[40] S. Hong, X. Yan, T. S. Huang, and H. Lee, "Learning hierarchical semantic image manipulation through structured representations," in *Advances in Neural Information Processing Systems*, 2018.

[41] H. Park, Y. Yoo, and N. Kwak, "MC-GAN: Multi-conditional generative adversarial network for image synthesis," in *British Machine Vision Conference*, 2018, p. 76.

[42] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1301–1310.

[43] L. T. Torres, T. M. Paixão, R. F. Berriel, A. F. De Souza, C. Badue, N. Sebe, and T. Oliveira-Santos, "Effortless deep training for traffic sign detection using templates and arbitrary natural images," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2019.

[44] M. Chen, T. Artières, and L. Denoyer, "Unsupervised object segmentation by redrawing," in *Advances in Neural Information Processing Systems*, 2019, pp. 12 705–12 716.

[45] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *6th Int. Conf. on Learning Representations*, 2018.

[46] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Int. C. on Computer Vision*, 2015, pp. 1422–1430.

[47] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *European Conf. on Computer Vision*. Springer, 2016, pp. 649–666.

[48] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[49] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[50] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE Int. Conf. on computer vision*, 2017.

[51] J. Yang, A. Kannan, D. Batra, and D. Parikh, "Lr-gan: Layered recursive generative adversarial networks for image generation," *ICLR*, 2017.

[52] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[53] 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," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[54] G. B. H. E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2014-003, May 2014.

[55] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[56] M. Zhu, P. Pan, W. Chen, and Y. Yang, "Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2019.

[57] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th Int. IEEE Workshop on 3D Representation and Recognition*, Sydney, Australia, 2013.

[58] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[59] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, 2005.

[60] 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," *arXiv preprint arXiv:1704.04861*, 2017.

[61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. on Learning Representations*, 2014.

[62] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th Int. Conf. on Artificial Intelligence and Statistics*, 2010.

[63] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[64] 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.