SampleAhead: Online Classifier-Sampler Communication for Learning from Synthesized Data

Qi Chen, Weichao Qiu, Yi Zhang, Lingxi Xie and Alan Yuille

Department of Computer Science, The Johns Hopkins University

ABSTRACT

State-of-the-art techniques of artificial intelligence, in particular deep learning, are mostly data-driven. However, collecting and manually labeling a large scale dataset is both difficult and expensive. A promising alternative is to introduce synthesized training data, so that the dataset size can be significantly enlarged with little human labor. But, this raises an important problem in active vision: given an **infinite** data space, how to effectively sample a **finite** subset to train a visual classifier?

This paper presents an approach for learning from synthesized data effectively. The motivation is straightforward – increasing the probability of seeing difficult training data. We introduce a module named **SampleAhead** to formulate the learning process into an online communication between a classifier and a sampler, and update them iteratively. In each round, we adjust the sampling distribution according to the classification results, and train the classifier using the data sampled from the updated distribution. Experiments are performed by introducing synthesized images rendered from ShapeNet models to assist PASCAL3D+ classification. Our approach enjoys higher classification accuracy, especially in the scenario of a limited number of training samples. This demonstrates its efficiency in exploring the infinite data space.

CONTRIBUTION

This is the first work that effectively samples synthesized images from an infinite data space to train deep convolutional neural networks (DCNN). Previously, all methods collect training images according to predefined distribution, and training data is fixed throughout the training phase, i.e. for every epoch, DCNNs see the same data.

we insert a novel module named **SampleAhead**, which maintains a distribution over the sampling space. In each training iteration, the distribution is first updated according to the current recognition results, and then used to sample synthesized training data and optimize the vision system. Although being simple, our approach works well in a challenging vision task – joint object detection and pose estimation, especially when the recognition task is difficult (e.g., the number of azimuth bins is large). The advantage of our approach becomes more significant in the scenario of limited training time.

REFERENCES

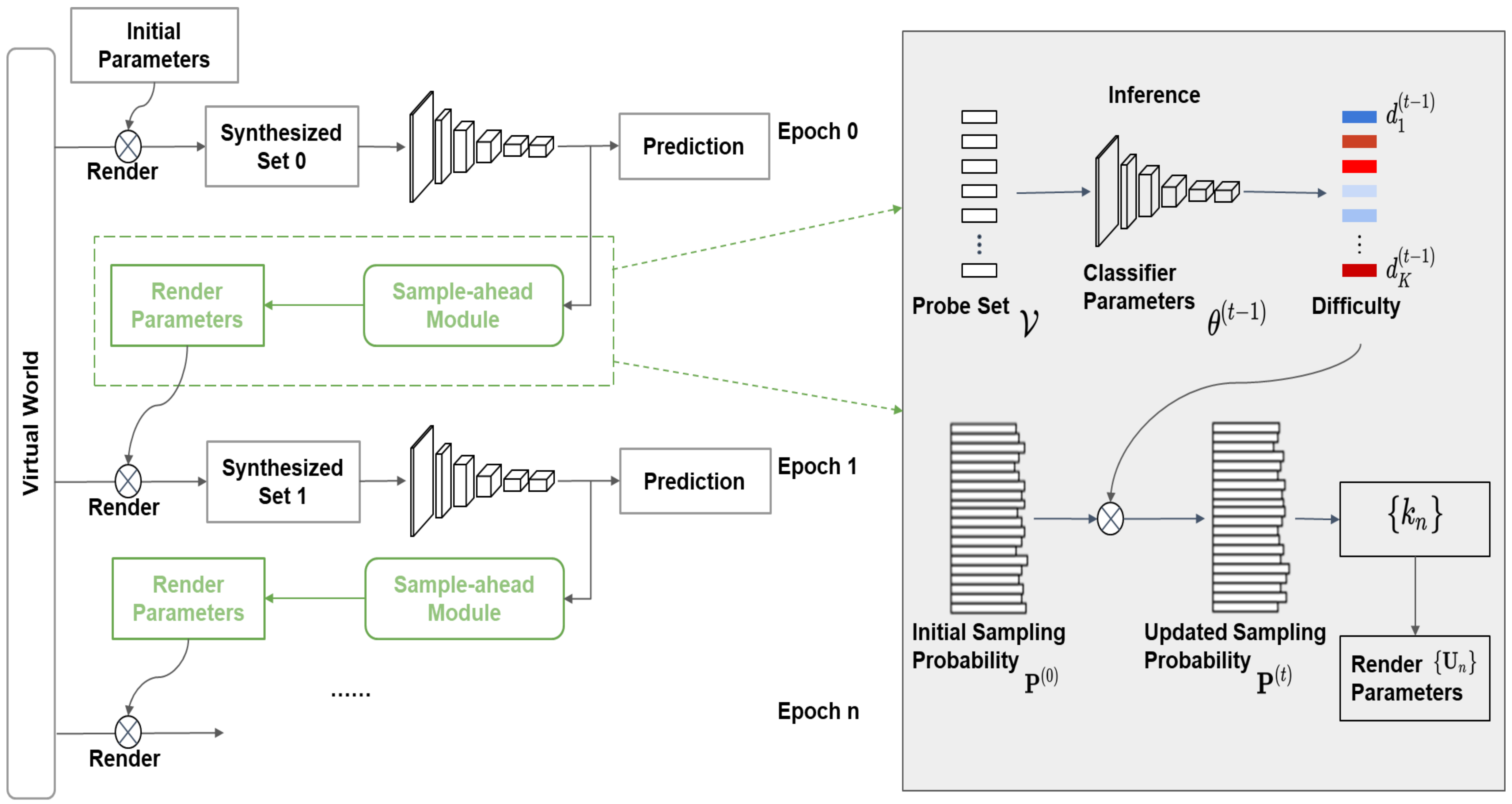
Key references are numbered as they appear in the paper.

[Su et al., 2015] H. Su, C.R. Qi, Y. Li, and L.J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In ICCV, 2015

[Xiang et al., 2014] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3D object detection in the wild. In WACV, 2014.

THE PROPOSED ALGORITHM

Overall Framework: Embedding SampleAhead module to Deep Convolutional Networks



Algorithm

Key idea:

- Increase the probability of seeing difficult training data

```
Algorithm 1: Training a Classifier with SampleAhead
Input  : parameter space $\mathcal{U}$;buckets $\{\mathcal{B}_k\}_{k=1}^{K}$;
         generator $\mathbf{g}(\cdot)$, initial model $\mathbf{f}\left(\cdot;\boldsymbol{\theta}^{(0)}\right)$;
         probe set $\mathcal{V}=\{\mathbf{V}_1,\mathbf{V}_2,\ldots,\mathbf{V}_M\}$;
         # of samples of each iteration $N$;
         hyper-parameters $\alpha$, $\beta$, max # of iterations $T$;
Output: trained model $\mathbf{f}\left(\cdot;\boldsymbol{\theta}^{(T)}\right)$;
t ← 0, $P_k^{(0)} \leftarrow 1/K, k=1,2,\ldots,K$;
repeat
    compute $d^{(t-1)}(\mathbf{V}_m)=$
      $1-\Pr\left[\mathbf{f}\left(\mathbf{g}(\mathbf{V}_m);\boldsymbol{\theta}^{(t-1)}\right)\text{ is correct}\right], \forall m$;
    compute $d_k^{(t-1)}=\frac{\sum_{\mathbf{V}_m\in\mathcal{B}_k} d^{(t-1)}(\mathbf{V}_m)}{|\mathcal{V}\cup\mathcal{B}_k|}, \forall k$;
    update $P_k^{(t)}=\alpha\cdot P_k^{(0)}+(1-\alpha)\cdot P_k^{(0)}\cdot e^{\beta\cdot d_k^{(t-1)}}$,
      $\forall k$;
    sample $\{k_n\}_{n=1}^{N}$ from distribution $\mathbf{P}^{(t)}$;
    sample $\{\mathbf{U}_n\}_{n=1}^{N}$ uniformly from $\{\mathcal{B}_{k_n}\}_{n=1}^{N}$;
    generate $\mathbf{X}_n=\mathbf{g}(\mathbf{U}_n), \forall n$;
    train $\mathbf{f}\left(\cdot;\boldsymbol{\theta}^{(t-1)}\right)$ into $\mathbf{f}\left(\cdot;\boldsymbol{\theta}^{(t)}\right)$ with $\{\mathbf{X}_n\}_{n=1}^{N}$;
until $t=T$;
Return: $\mathbf{f}\left(\cdot;\boldsymbol{\theta}^{(T)}\right)$.
```

Experiments on MNIST (Digit Classification)

Experimental settings:

- Synthesizing Methods: We consider seven types of augmentation, i.e. digit rotation, vertical/horizontal scaling, horizontal/vertical shifting and horizontal/vertical shearing.
- Transformation parameters: Rotation angle and Scaling/Shifting/Shearing parameters. We divide these parameters into 16 buckets.
- Backbone network: LeNet [LeCun et al., 1998]
- Other settings: please refer to our paper.

Classification error rates (%) on the MNIST Dataset

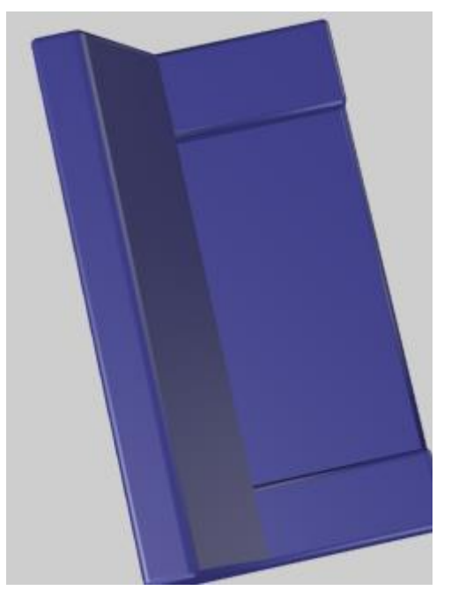
#images	10,000	20,000	30,000	40,000	50,000
Uniform Sampling	0.890 ± 0.021	0.835 ± 0.024	0.816 ± 0.021	0.796 ± 0.018	0.793 ± 0.035
Our Approach	0.819 ± 0.025	0.787 ± 0.019	0.756 ± 0.022	0.758 ± 0.026	0.757 ± 0.014
P-value	6.407 × 10^{-5}	2.434 × 10^{-3}	4.545 × 10^{-4}	8.097 × 10^{-3}	3.504 × 10^{-2}

Experiments on Pascal3D+ (Object Pose Estimation)

Experimental settings:

- Synthesizing Methods: Render images from 3D CAD models of ShapeNet [Chang et al., 2015] -> Crop -> Overlay with backgrounds.
- Rendering parameters: Object class, 3D model type, Azimuth, Elevation, Tilt, Object-camera distance etc. We set other parameters random, while dividing object class, azimuth and elevation into 2376 buckets.
- The basic structure: RCNN (fixed for detection) and AlexNet.

Generated image samples

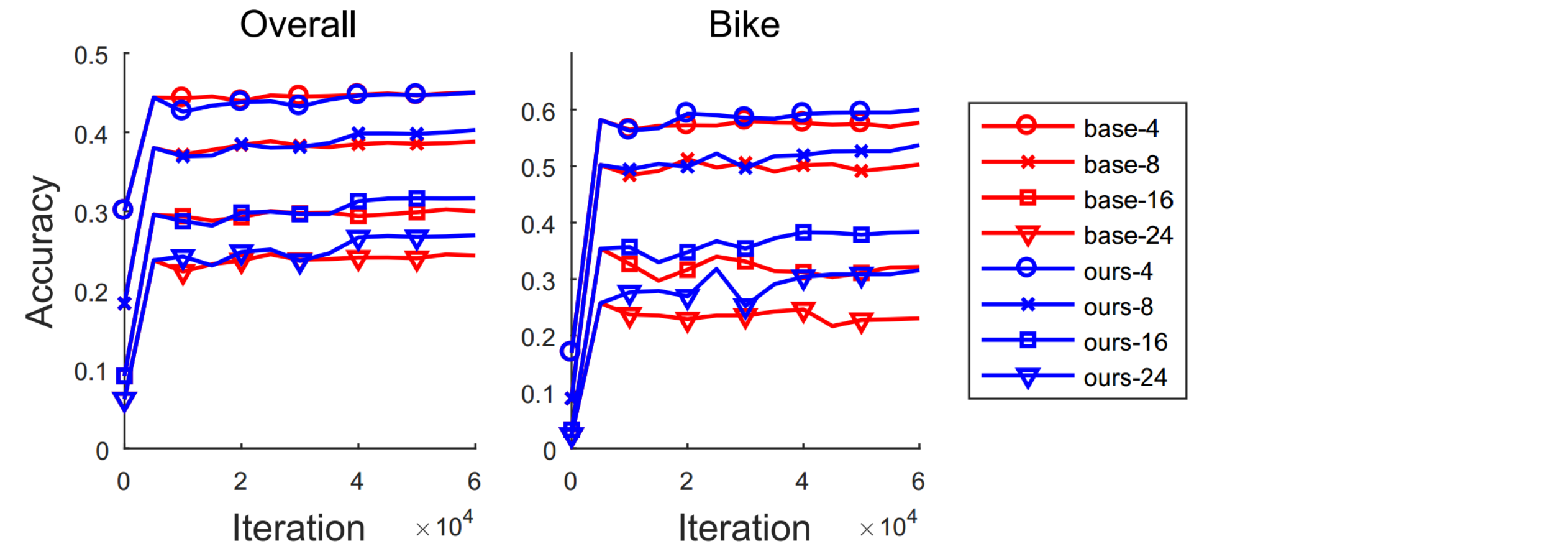


Accuracy (%) of object detection and pose estimation

L is the number of azimuth bins. The higher, the task is more difficult.

Approach	Mean (L = 4)	Mean (L = 8)	Mean (L =16)	Mean (L = 24)
[Su et al., 2015]	39.7	32.9	24.2	19.8
Baseline	44.8	38.2	29.3	23.5
Ours	45.0	40.1	31.9	27.2

Our approach works better in the situation of limited data sampling



Accuracy and Median Error (radius) on view estimation with ground-truth bounding boxes provided

Approach	$\mathrm{Acc}^{\pi/6}$ (Baseline)	$\mathrm{Acc}^{\pi/6}$ (Ours)	MedErr (Baseline)	MedErr (Ours)
mean	0.84	0.84	10.2	8.6