Paper Reading
Minghao Liu
2023/05/28

Progressive Transformation Learning for Leveraging Virtual Images in Training

Yi-Ting Shen[⋆,1] Hyungtae Lee[⋆,2] Heesung Kwon[2] Shuvra S. Bhattacharyya[1]
[⋆] equal contribution
[1] University of Maryland, College Park [2] DEVCOM Army Research Laboratory

CVPR 2023 HIGHLIGHT

Content

- Authors
- Background
- Method
- Experiments

Content

- Authors
- Background
- Method
- Experiments

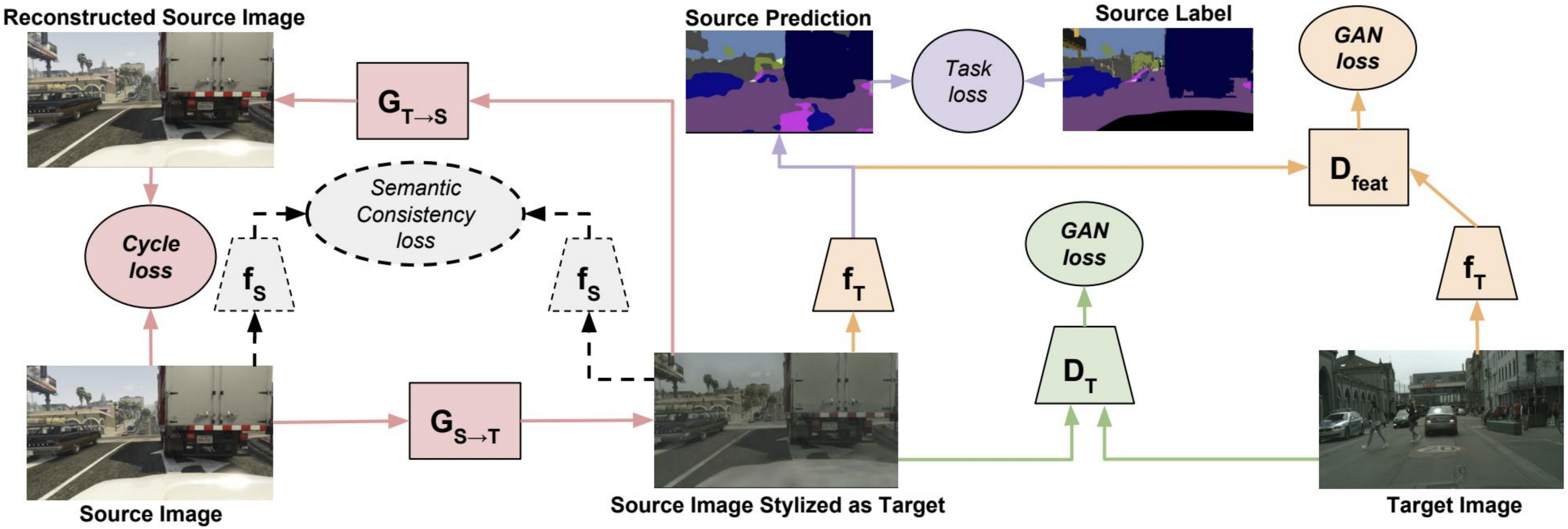
Background

CyCADA

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. *CyCADA: Cycle-consistent adversarial domain adaptation.* In Proc. ICML, 2018.

Background

CyCADA



Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. *CyCADA: Cycle-consistent adversarial domain adaptation.* In Proc. ICML, 2018.

Background

CyCADA

$$\begin{aligned}
&\mathcal{L}_{\text{CyCADA}}(f_T, X_S, X_T, Y_S, G_{S\to T}, G_{T\to S}, D_S, D_T) \\
&\quad = \mathcal{L}_{\text{task}}(f_T, G_{S\to T}(X_S), Y_S) \\
&\quad + \mathcal{L}_{\text{GAN}}(G_{S\to T}, D_T, X_T, X_S) + \mathcal{L}_{\text{GAN}}(G_{T\to S}, D_S, X_S, X_T) \\
&\quad + \mathcal{L}_{\text{GAN}}(f_T, D_{\text{feat}}, f_S(G_{S\to T}(X_S)), X_T) \\
&\quad + \mathcal{L}_{\text{cyc}}(G_{S\to T}, G_{T\to S}, X_S, X_T) + \mathcal{L}_{\text{sem}}(G_{S\to T}, G_{T\to S}, X_S, X_T, f_S).
\end{aligned}$$

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. *CyCADA: Cycle-consistent adversarial domain adaptation.* In Proc. ICML, 2018.

Background

Deep Mahalanobis detector

- Mahalanobis distance：
  - Single sample： $D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$
  - Two samples: $D_M(x, y) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)}$

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks.* In Proc. NeurIPS, 2018.

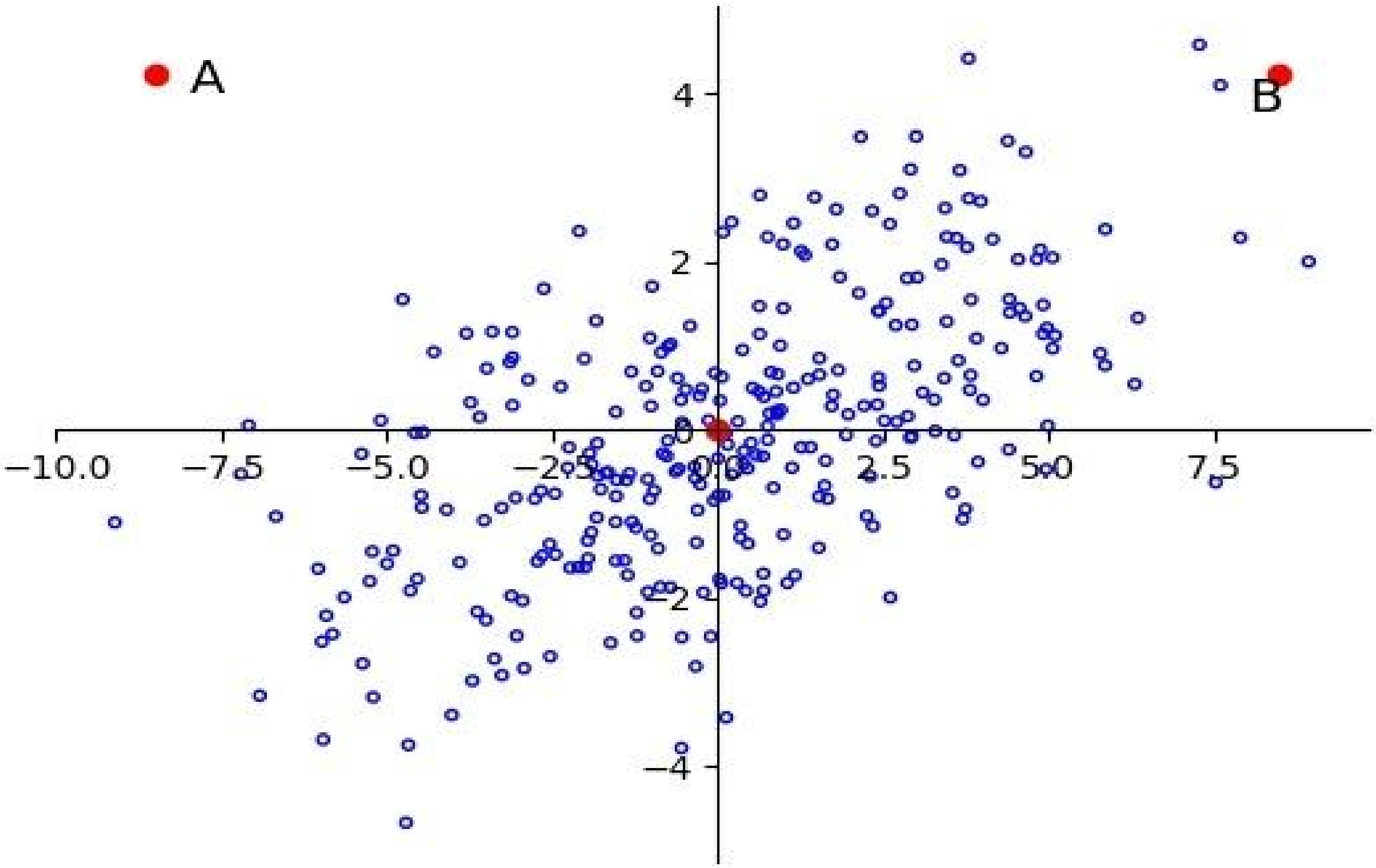
Background

Deep Mahalanobis detector

- Mahalanobis distance:
  - Single sample: $D_M(x) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)}$ covariance matrix mean of samples
  - Two samples: $D_M(x,y) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)}$

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks.* In Proc. NeurIPS, 2018.

Background



Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks.* In Proc. NeurIPS, 2018.

Background

- Mahalanobis distance: $Z = Y \begin{bmatrix} \frac{1}{\sqrt{\sigma_1}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{\sigma_F}} \end{bmatrix} \Leftrightarrow = Y\Lambda$ $\quad \Sigma = \mathrm{U}^{\mathrm{T}}\mathrm{Q}\mathrm{U}$ $\quad \mathrm{Y} = \mathrm{X}\mathrm{U}$

$$
\begin{aligned}
\mathrm{Dis}_{\mathrm{Euclidean}}(\mathrm{z}_1, \mathrm{z}_2) &= \sqrt{(\mathrm{z}_1 - \mathrm{z}_2)(\mathrm{z}_1 - \mathrm{z}_2)^{\mathrm{T}}} \\
&= \sqrt{(\mathrm{y}_1\Lambda - \mathrm{y}_2\Lambda)(\mathrm{y}_1\Lambda - \mathrm{y}_2\Lambda)^{\mathrm{T}}} \\
&= \sqrt{(\mathrm{x}_1\mathrm{U}\Lambda - \mathrm{x}_2\mathrm{U}\Lambda)(\mathrm{x}_1\mathrm{U}\Lambda - \mathrm{x}_2\mathrm{U}\Lambda)^{\mathrm{T}}} \\
&= \sqrt{(\mathrm{x}_1 - \mathrm{x}_2)\mathrm{U}\Lambda\Lambda\mathrm{U}^{\mathrm{T}}(\mathrm{x}_1 - \mathrm{x}_2)^{\mathrm{T}}} \\
&= \sqrt{(\mathrm{x}_1 - \mathrm{x}_2)\Sigma^{-1}(\mathrm{x}_1 - \mathrm{x}_2)^{\mathrm{T}}} \\
&= \mathrm{Dis}_{\mathrm{mahalanobis}}(\mathrm{x}_1, \mathrm{x}_2)
\end{aligned}
$$

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks.* In Proc. NeurIPS, 2018.

Background

- Mahalanobis distance:

$$Z = Y\begin{bmatrix}\frac{1}{\sqrt{\sigma_1}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{\sigma_F}}\end{bmatrix} \Leftrightarrow = Y\Lambda \qquad \Sigma = \mathrm{U}^{\mathrm{T}}\mathrm{Q}\mathrm{U} \qquad \mathrm{Y} = \mathrm{X}\mathrm{U}$$

$$\begin{aligned}
\mathrm{Dis}_{\mathrm{Euclidean}}(\mathrm{z}_1, \mathrm{z}_2) &= \sqrt{(\mathrm{z}_1 - \mathrm{z}_2)(\mathrm{z}_1 - \mathrm{z}_2)^{\mathrm{T}}} \\
&= \sqrt{(\mathrm{y}_1\Lambda - \mathrm{y}_2\Lambda)(\mathrm{y}_1\Lambda - \mathrm{y}_2\Lambda)^{\mathrm{T}}} \\
&= \sqrt{(\mathrm{x}_1\mathrm{U}\Lambda - \mathrm{x}_2\mathrm{U}\Lambda)(\mathrm{x}_1\mathrm{U}\Lambda - \mathrm{x}_2\mathrm{U}\Lambda)^{\mathrm{T}}} \\
&= \sqrt{(\mathrm{x}_1 - \mathrm{x}_2)\mathrm{U}\Lambda\Lambda\mathrm{U}^{\mathrm{T}}(\mathrm{x}_1 - \mathrm{x}_2)^{\mathrm{T}}} \\
&= \sqrt{(\mathrm{x}_1 - \mathrm{x}_2)\Sigma^{-1}(\mathrm{x}_1 - \mathrm{x}_2)^{\mathrm{T}}} \\
&= \mathrm{Dis}_{\mathrm{mahalanobis}}(\mathrm{x}_1, \mathrm{x}_2)
\end{aligned}$$

Hypothesis: The features of each layer of convolutional neural networks follow a multivariate Gaussian distribution

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks.* In Proc. NeurIPS, 2018.

Background

We can calculate the mean value and the covariance matrix of all samples of each category in each convolution layer.

$$\hat{\mu_c} = \frac{1}{N_c} \sum_{i:y_i=c} f(x_i)$$

$$\hat{\Sigma} = \frac{1}{N} \sum_c \sum_{i:y_i=c} (f(x_i) - \hat{\mu_c})(f(x_i) - \hat{\mu_c})^T$$

For a new sample x, its degree of category confidence:

$$M(x) = \max_c -(f(x) - \hat{\mu_c})^T \hat{\Sigma}^{-1} (f(x) - \hat{\mu_c})$$

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks.* In Proc. NeurIPS, 2018.

Background

Deep Mahalanobis detector

In-dist (model)	OOD	Validation on OOD samples			Validation on adversarial samples		
		TNR at TPR 95%	AUROC	Detection acc.	TNR at TPR 95%	AUROC	Detection acc.
		Baseline [13] / ODIN [21] / Mahalanobis (ours)			Baseline [13] / ODIN [21] / Mahalanobis (ours)		
CIFAR-10 (DenseNet)	SVHN	40.2 / 86.2 / **90.8**	89.9 / 95.5 / **98.1**	83.2 / 91.4 / **93.9**	40.2 / 70.5 / **89.6**	89.9 / 92.8 / **97.6**	83.2 / 86.5 / **92.6**
	TinyImageNet	58.9 / 92.4 / **95.0**	94.1 / 98.5 / **98.8**	88.5 / 93.9 / **95.0**	58.9 / 87.1 / **94.9**	94.1 / 97.2 / **98.8**	88.5 / 92.1 / **95.0**
	LSUN	66.6 / 96.2 / **97.2**	95.4 / 99.2 / **99.3**	90.3 / 95.7 / **96.3**	66.6 / 92.9 / **97.2**	95.4 / 98.5 / **99.2**	90.3 / 94.3 / **96.2**
CIFAR-100 (DenseNet)	SVHN	26.7 / 70.6 / **82.5**	82.7 / 93.8 / **97.2**	75.6 / 86.6 / **91.5**	26.7 / 39.8 / **62.2**	82.7 / 88.2 / **91.8**	75.6 / 80.7 / **84.6**
	TinyImageNet	17.6 / 42.6 / **86.6**	71.7 / 85.2 / **97.4**	65.7 / 77.0 / **92.2**	17.6 / 43.2 / **87.2**	71.7 / 85.3 / **97.0**	65.7 / 77.2 / **91.8**
	LSUN	16.7 / 41.2 / **91.4**	70.8 / 85.5 / **98.0**	64.9 / 77.1 / **93.9**	16.7 / 42.1 / **91.4**	70.8 / 85.7 / **97.9**	64.9 / 77.3 / **93.8**
SVHN (DenseNet)	CIFAR-10	69.3 / 71.7 / **96.8**	91.9 / 91.4 / **98.9**	86.6 / 85.8 / **95.9**	69.3 / 69.3 / **97.5**	91.9 / 91.9 / **98.8**	86.6 / 86.6 / **96.3**
	TinyImageNet	79.8 / 84.1 / **99.9**	94.8 / 95.1 / **99.9**	90.2 / 90.4 / **98.9**	79.8 / 79.8 / **99.9**	94.8 / 94.8 / **99.8**	90.2 / 90.2 / **98.9**
	LSUN	77.1 / 81.1 / **100**	94.1 / 94.5 / **99.9**	89.1 / 89.2 / **99.3**	77.1 / 77.1 / **100**	94.1 / 94.1 / **99.9**	89.1 / 89.1 / **99.2**
CIFAR-10 (ResNet)	SVHN	32.5 / 86.6 / **96.4**	89.9 / 96.7 / **99.1**	85.1 / 91.1 / **95.8**	32.5 / 40.3 / **75.8**	89.9 / 86.5 / **95.5**	85.1 / 77.8 / **89.1**
	TinyImageNet	44.7 / 72.5 / **97.1**	91.0 / 94.0 / **99.5**	85.1 / 86.5 / **96.3**	44.7 / 69.6 / **95.5**	91.0 / 93.9 / **99.0**	85.1 / 86.0 / **95.4**
	LSUN	45.4 / 73.8 / **98.9**	91.0 / 94.1 / **99.7**	85.3 / 86.7 / **97.7**	45.4 / 70.0 / **98.1**	91.0 / 93.7 / **99.5**	85.3 / 85.8 / **97.2**
CIFAR-100 (ResNet)	SVHN	20.3 / 62.7 / **91.9**	79.5 / 93.9 / **98.4**	73.2 / 88.0 / **93.7**	20.3 / 12.2 / **41.9**	79.5 / 72.0 / **84.4**	73.2 / 67.7 / **76.5**
	TinyImageNet	20.4 / 49.2 / **90.9**	77.2 / 87.6 / **98.2**	70.8 / 80.1 / **93.3**	20.4 / 33.5 / **70.3**	77.2 / 83.6 / **87.9**	70.8 / 75.9 / **84.6**
	LSUN	18.8 / 45.6 / **90.9**	75.8 / 85.6 / **98.2**	69.9 / 78.3 / **93.5**	18.8 / 31.6 / **56.6**	75.8 / 81.9 / **82.3**	69.9 / 74.6 / **79.7**
SVHN (ResNet)	CIFAR-10	78.3 / 79.8 / **98.4**	92.9 / 92.1 / **99.3**	90.0 / 89.4 / **96.9**	78.3 / 79.8 / **94.1**	92.9 / 92.1 / **97.6**	90.0 / 89.4 / **94.6**
	TinyImageNet	79.0 / 82.1 / **99.9**	93.5 / 92.0 / **99.9**	90.4 / 89.4 / **99.1**	79.0 / 80.5 / **99.2**	93.5 / 92.9 / **99.3**	90.4 / 90.1 / **98.8**
	LSUN	74.3 / 77.3 / **99.9**	91.6 / 89.4 / **99.9**	89.0 / 87.2 / **99.5**	74.3 / 76.3 / **99.9**	91.6 / 90.7 / **99.9**	89.0 / 88.2 / **99.5**

Table 2: Distinguishing in- and out-of-distribution test set data for image classification under various validation setups. All values are percentages and the best results are indicated in bold.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. *A simple unified framework for detecting out-of-distribution samples and adversarial attacks.* In Proc. NeurIPS, 2018.

Content

Motivation

Unmanned aerial vehicle (UAV) $\longleftrightarrow$ Ground-based cameras

Wide variety of camera viewing angles

Expand the training set $\longrightarrow$ Domain gap

Motivation

Unnmanned aerial vehicle (UAV) $\longleftrightarrow$ Ground-based cameras

Wide variety of camera viewing angles

Expand the training set $\longrightarrow$ Domain gap

How to measure accurately the domain gap?

Method

Softmax-based classifier: representation space

The distribution of each category——multivariate Gaussian distribution.

$$\Downarrow$$

Sigmoid-based detector: representation space

The distribution of each category——multivariate Gaussian distribution.

Method

- The representation space of the sigmoid-based detector:

$$P(f(\mathbf{x})|y_c = 1) \sim \mathcal{N}(f(\mathbf{x})|\mu_c, \mathbf{\Sigma}_c), \qquad (1)$$

where

$$\begin{aligned} \mu_c &= \frac{1}{|\mathrm{D}_c|} \sum_{\mathbf{x} \in \mathrm{D}_c} f(\mathbf{x}), \\ \mathbf{\Sigma}_c &= \frac{1}{|\mathrm{D}_c|} \sum_{\mathbf{x} \in \mathrm{D}_c} \left(f(\mathbf{x}) - \mu_c\right) \left(f(\mathbf{x}) - \mu_c\right)^\top, \qquad (2) \end{aligned}$$

Method

- The domain gap between a new image x_{new} and D_c:

$$d(\mathbf{x}_{new}) = \left(f(\mathbf{x}_{new}) - \mu_c\right)^\top \mathbf{\Sigma}_c^{-1} \left(f(\mathbf{x}_{new}) - \mu_c\right) . \quad (3)$$

To mitigate the effect of image size on measuring the domain gap——multiple image scales:

$$d(\mathbf{x}_{new}) = \min_{s \in S}(\{d(\mathbf{x}_{new}^s)\}), \quad (4)$$

where S={128, 256, 384, 512}.

Method

Progressive Transformation Learning:

- Transformation candidate selection

$$w(\mathbf{x}) = \exp\left(-\frac{d(\mathbf{x})}{\tau}\right), \qquad (5)$$

Method

Progressive Transformation Learning:

- Virtual2Real transformation：

  Crop the person region——apply the transformation

- CycleGAN

Method

Set update: $\mathrm{R}^{t+1} = \mathrm{R}^t \cup \mathrm{C_R}^t \qquad \mathrm{V}^{t+1} = \mathrm{V}^t / \mathrm{C_V}^t \qquad (6)$

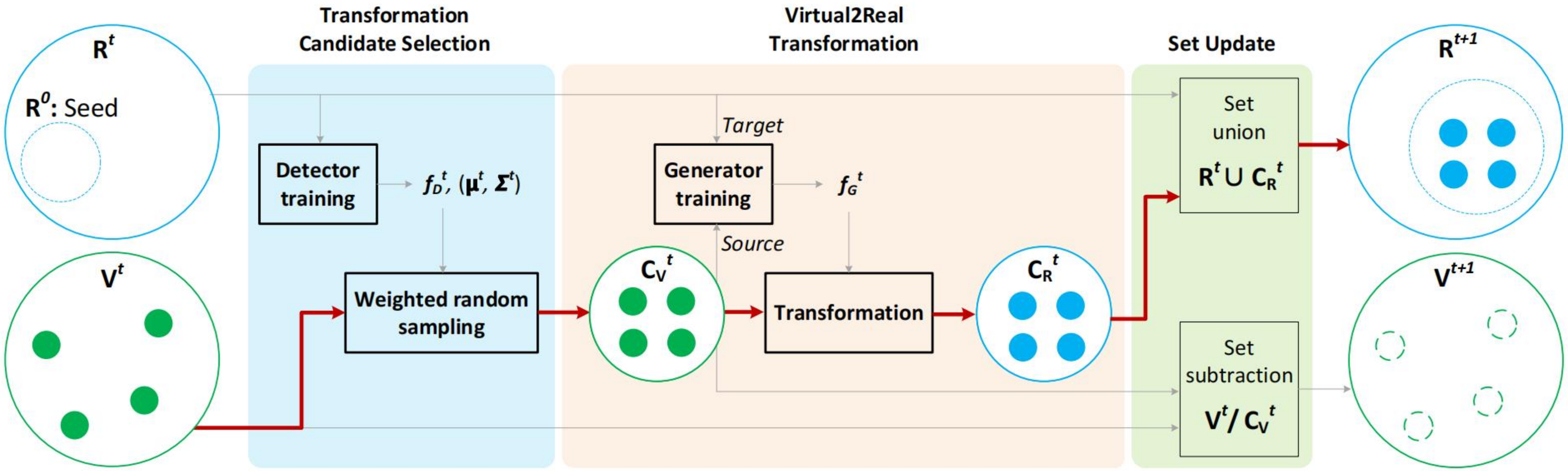


Figure 2. **Progressive Transformation Learning (PTL) pipeline.** The red arrow indicates the processing flow of the virtual images selected to be added to the training set.

Experiments

Real UAV-based datasets:

- VisDrone
- Okutama Action
- ICG

Virtual dataset

- Archangel Synthetic

Metrics

- AP@.5
- AP@[.5:.95]

Experiments

Detector—— RetinaNet

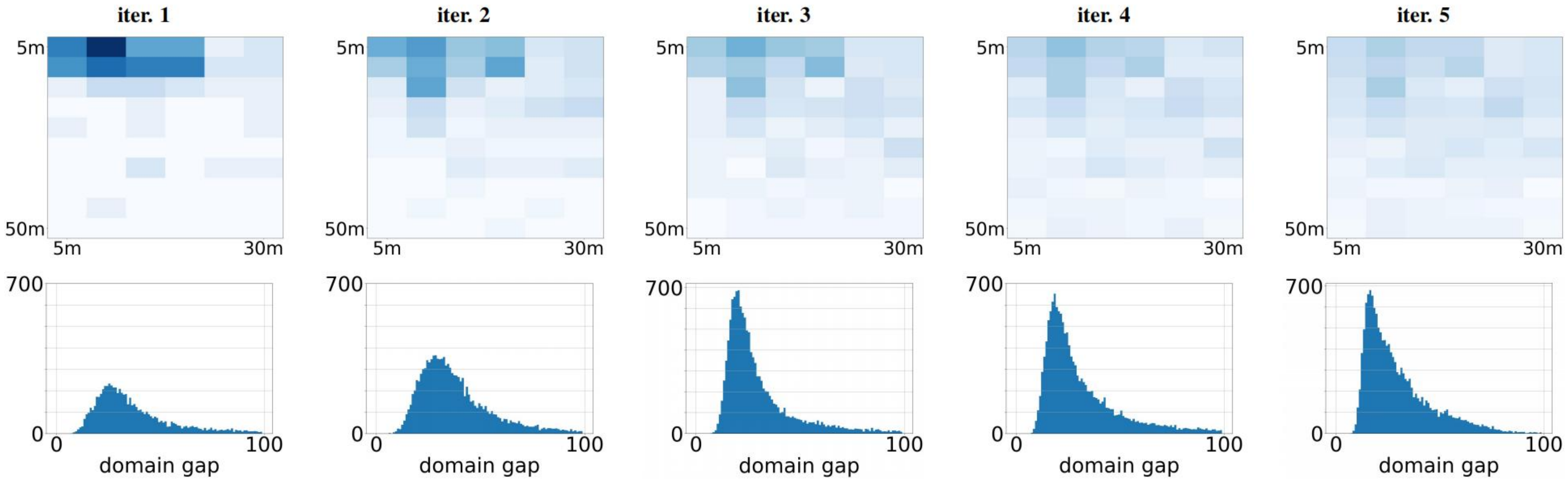


Figure 3. **Analysis of the use of virtual images when PTL progresses.** The figures in the top row show the accumulated distribution of transformation candidates with respect to camera locations (i.e., altitude and rotation circle radius from the target human in x and y axes, respectively) for each PTL iteration. Darker bins indicate that more virtual images have been added to the training set. The figures in the bottom row (x axis: domain gap, y axis: the corresponding number of virtual images) show the domain gap distribution of virtual images measured by eq. 4. These figures are collected from the experimental setup of using 50 real images from the VisDrone dataset for training.

Experiments

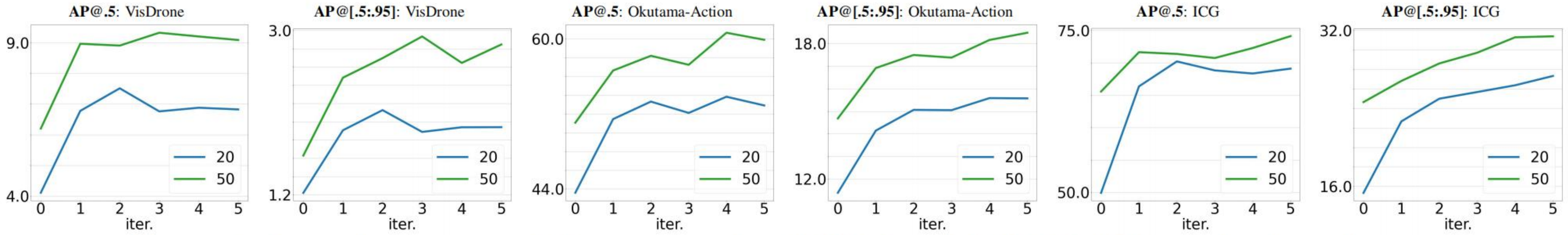


Figure 4. **Learning curves** of the two metrics (AP@.5 and AP@[.5:.95]) on the three datasets

method	train set	VisDrone 20	VisDrone 50	Okutama-Action 20	Okutama-Action 50	ICG 20	ICG 50
baseline	R	3.74/ 1.09	6.42/ 1.86	41.61/ 11.23	49.84/ 13.76	49.35/ 14.69	66.75/ 23.91
pretrain-finetune	R+V	4.99/ 1.46	6.25/ 1.99	44.57/ 12.78	49.06/ 15.08	66.92/ 26.67	68.41/ 29.73
naive merge	R+V	3.41/ 1.02	5.18/ 1.65	34.26/ 9.21	48.33/ 14.61	55.95/ 20.76	65.68/ 26.73
w/ transform	R+V	1.26/ 0.49	4.02/ 1.37	27.37/ 7.84	41.36/ 12.64	48.02/ 17.62	65.03/ 27.21
PTL (5th itr.)	R+V	6.83/ 1.94	9.09/ 2.85	52.89/ 15.57	59.90/ **18.48**	69.11/ **27.33**	**74.14/ 31.41**
		+3.09/+0.85	+2.67/+0.99	+11.28/ +4.34	+10.06/ +4.72	+19.76/+12.64	+7.39/ +7.50
PTL (best)	R+V	**7.52/ 2.13**	**9.33/ 2.94**	**53.82/ 15.59**	**60.65/ 18.48**	**70.23/ 27.33**	**74.14/ 31.41**
		+3.78/+1.04	+2.91/+1.08	+12.21/ +4.36	+10.81/ +4.72	+20.88/+12.64	+7.39/ +7.50

Table 1. **Low-shot learning accuracy** with 20 and 50 real images. AP@.5 and AP@[.5:.95] are reported in each bin. For PTL, the margin from the baseline accuracy is shown below the reported accuracy. The best accuracy for each setting is shown in bold. R and V denote the set of real images and the set of virtual images, respectively.

Experiments

τ	VisDrone	Okutama-Action	ICG
1	9.48/ 3.01	39.37/ 10.45	27.87/ 7.75
5	9.09/ 2.85	**42.39/ 11.41**	29.26/ 7.27
10	**9.68/ 2.87**	37.48/ 9.51	33.06/ 7.66
100	8.97/ 2.54	38.25/ 10.18	33.78/ 9.29
1000	8.90/ 2.63	39.15/ 10.00	**43.90/ 11.97**

Table 2. **Varying τ** in PTL (after 5th iteration). Models are trained on the VisDrone dataset with 50 shot learning setup.

# img per itr	VisDrone	Okutama-Action	ICG
50	**9.59/** 2.90	41.62/ 11.19	25.94/ 6.04
100	9.33/ **2.94**	**42.39/ 11.46**	30.01/ 7.36
200	9.04/ 2.91	41.29/ 11.18	**35.50/ 10.20**

Table 3. **Varying # of virtual images added to the training set per PTL iteration.** Models are trained on the VisDrone dataset with 50 shot learning setup. The reported accuracies are obtained by using the best PTL models.

Naive merge w/ transform PTL

Figure 5. **Sample Virtual2Real transformation output (VisDrone, 50-shot).** Each set consists of three images: original virtual image (left), transformed image (middle), and transformed image with background (right).

Experiments

method	VisDrone 20	VisDrone 50	Okutama-Action 20	Okutama-Action 50	ICG 20	ICG 50
	Vis → *Vis*		*Oku* → *Oku*		*ICG* → *ICG*	
baseline	3.74/ 1.09	6.42/ 1.86	41.61/ 11.23	49.84/ 13.76	49.35/ 14.69	66.75/ 23.91
PTL (5th itr.)	6.83/ 1.94	9.09/ 2.85	52.89/ 15.57	59.90/ 18.48	69.11/ 27.33	74.14/ 31.41
PTL (best)	7.52/ 2.13	9.33/ 2.94	53.82/ 15.59	60.65/ 18.48	70.23/ 27.33	74.14/ 31.41
	Oku → *Vis*		*Vis* → *Oku*		*Vis* → *ICG*	
baseline	1.62/ 0.47	2.04/ 0.57	17.13/ 4.53	36.82/ 9.87	2.92/ 0.56	7.46/ 1.83
PTL (5th itr.)	2.72/ 0.94	3.05/ 1.07	30.72/ 7.45	42.39/ 11.41	26.86/ 7.22	29.26/ 7.27
PTL (best)	3.00/ 1.22	3.56/ 1.17	33.25/ 8.59	42.39/ 11.46	29.60/ 7.69	30.01/ 7.36
	ICG → *Vis*		*ICG* → *Oku*		*Oku* → *ICG*	
baseline	0.54/ 0.13	0.99/ 0.26	3.56/ 0.75	10.27/ 2.49	5.37/ 1.25	5.23/ 1.20
PTL (5th itr.)	1.09/ 0.33	1.61/ 0.50	11.19/ 2.58	14.20/ 3.56	28.98/ 8.14	25.39/ 6.53
PTL (best)	1.58/ 1.02	1.70/ 0.63	12.82/ 2.96	14.20/ 3.71	28.98/ 8.14	26.62/ 6.53

Table 4. **Cross-domain detection accuracy.** The table shows experiments with 3×3 cross-domain setups. For each setup, datasets shown to the left and right of the arrow are the training and test sets, respectively. The accuracies of PTL and the baseline without using virtual images for training are shown. Setups on the top use training and test images from the same domain, which provides a baseline accuracy in the cross-domain setups. All setups in each column are tested on the same dataset and the same low-shot regime.

Thanks For Listening