DOMAIN GENERALIZATION WITH MIXSTYLE

K Zhou etc.
Cite: 3

2021.03.31.수
발표자: 임진혁

0 MIIL Index

0 MIIL Index

1. DG가 무엇인지
2. DA랑 어떻게 다른지
3. 기본적인 방법론
4. 이게 왜 좋은지, 어떤 특징을 보이는지
5. IN 설명
6. Style transfer 설명
7. Adain 설명

References

https://theonly1.tistory.com/301 [Domain Adoption 문제 정의]

http://dsba.korea.ac.kr/seminar/?mod=document&uid=1325 [Domain Adoption 방법론]

1 Introduction: what is Domain Generalization?

CNN의 한계: only success on the i.i.d. assumption

➔ Train / Test should be drawn from same distribution

따라서 논문 실험에서는 강력한 성능을 발휘하지만(ex- ImageNet)
i.i.d. assumption을 기대하기 힘든 (Train/Test => different distribution = Unseen Domain)
Real World applications 에서는 performance degradation인 경우가 흔함

1 Introduction: what is Domain Generalization?

CNN의 한계: only success on the i.i.d. assumption

➔ Train / Test should be drawn from same distribution

따라서 논문 실험에서는 강력한 성능을 발휘하지만(ex- ImageNet)
i.i.d. assumption을 기대하기 힘든 (Train/Test => different distribution = Unseen Domain)
Real World applications 에서는 performance degradation인 경우가 흔함

=> Domain Generalization은 Unseen Domain Performance를 해결하고자 하는 문제.

Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

Class마다 다양한 source domains로 이루어진 visual images setting에서
<Data Distribution changing across domain problem>(=Domain Drifts)에 강건한 모델 학습

➔ Trained Model can generalize unseen domain

Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

What is different from “Domain Adoption”?

임진혁<eeplearning@units.ac.kr>

Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

"Domain Adoption": Train / Test가 같은 distrubution에 있지 않아서 Sourse dataset Train이 의미 없어지는(Domain Drift) problem

➔ S(source) Domain Data학습으로 T(target) Domain에서 good Performance

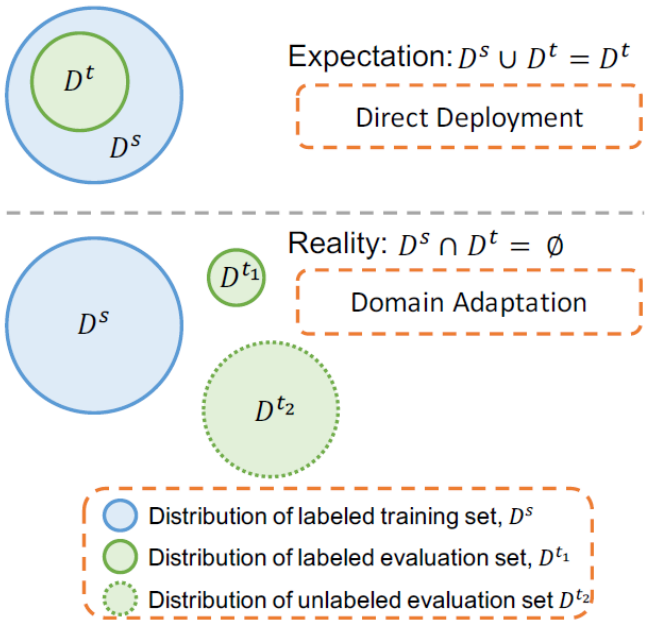


Figure 1. Domain adaptation in the true data space: Expectation vs. Reality.

임진혁<eeplearning@units.ac.kr>

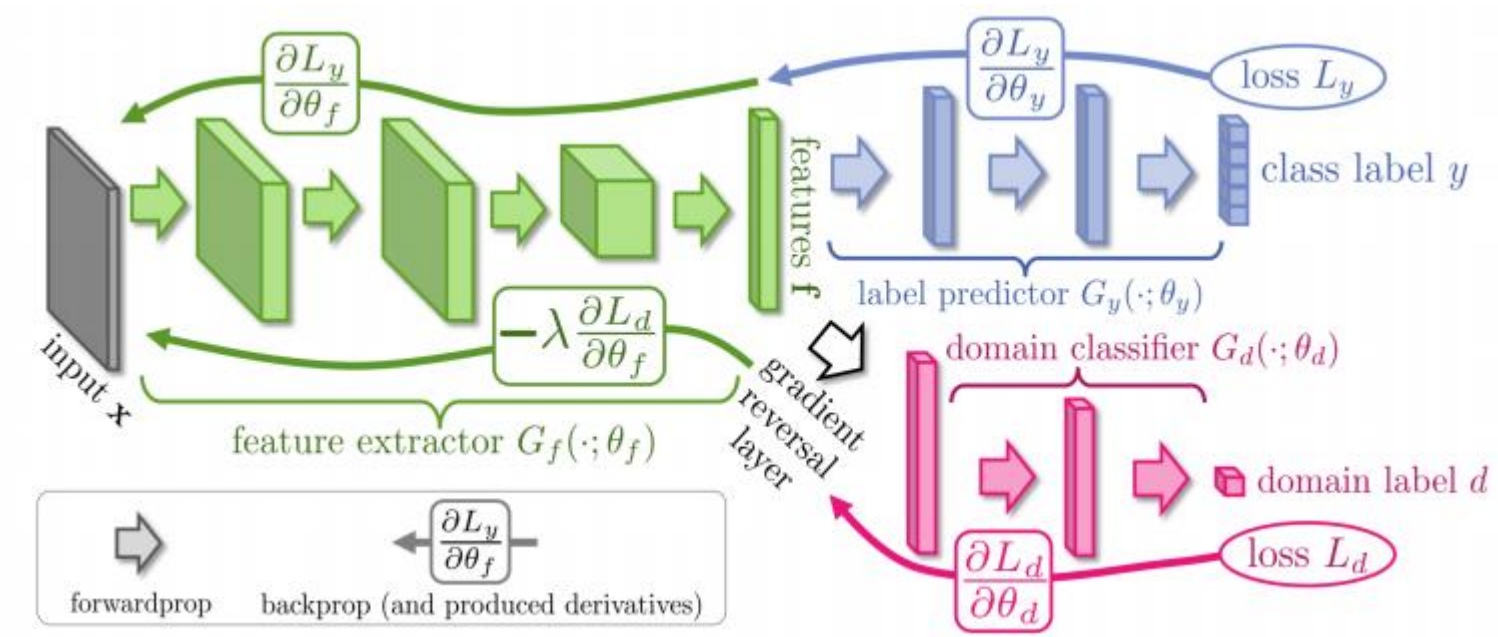
Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

“Domain Adoption”:

[1] Ganin et al,2015,“Unsupervised Domain Adaptation by Backpropagation”

Gradient Reversal Layer(GRL): Encoder(=>feature map) / Classifier(Domain) / Classifier(Class)



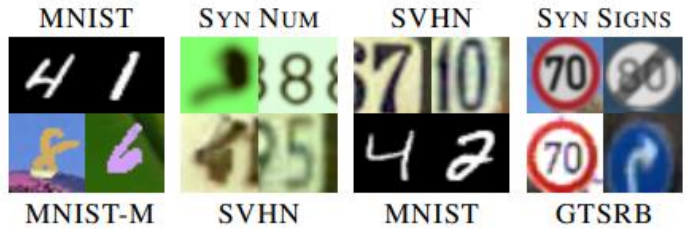


Figure 2. Examples of domain pairs (top – source domain, bottom – target domain) used in the small image experiments. See Section 4.1 for details.

임진혁<eeplearning@units.ac.kr>

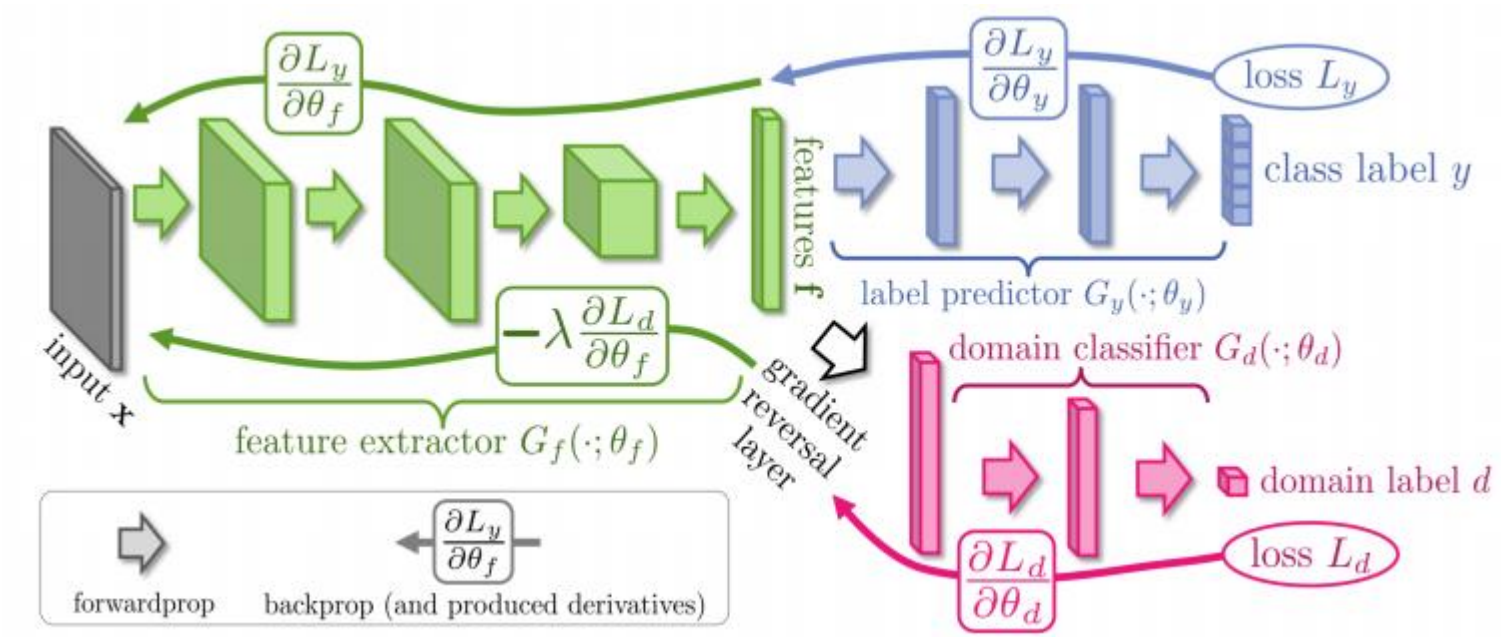
1 Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

“Domain Adoption”:

[1] Ganin et al,2015,“Unsupervised Domain Adaptation by Backpropagation”

Gradient Reversal Layer(GRL): Encoder(=>feature map) / Classifier(Domain) / Classifier(Class)



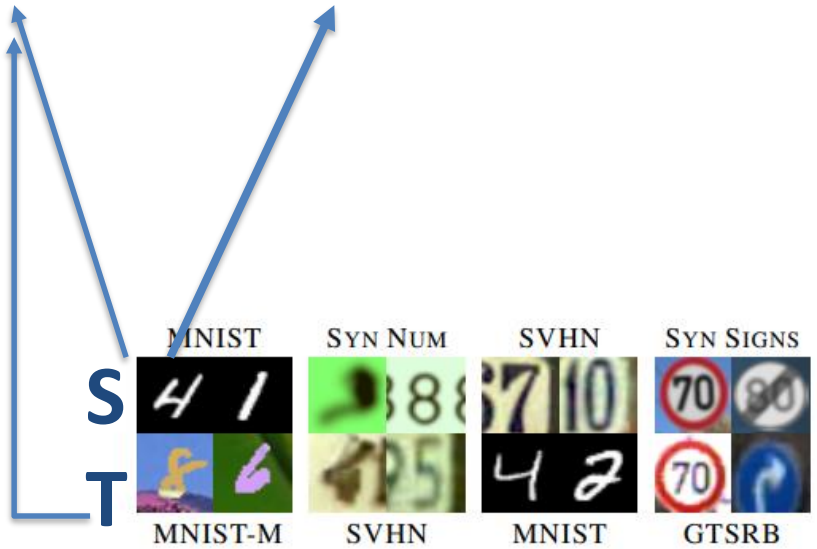


Figure 2. Examples of domain pairs (top – source domain, bottom – target domain) used in the small image experiments. See Section 4.1 for details.

임진혁<eeplearning@units.ac.kr>

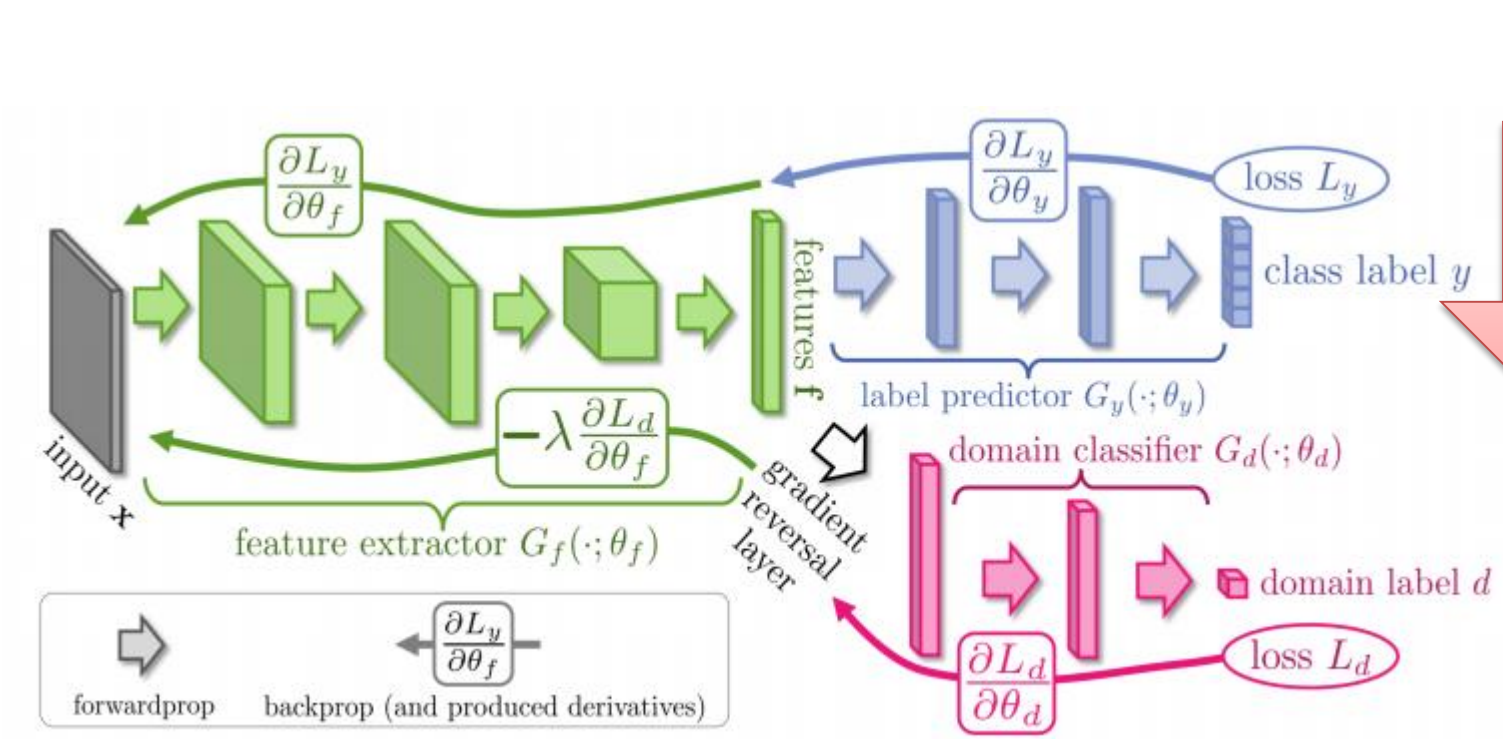
1 Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

“Domain Adoption”:

[1] Ganin et al,2015,“Unsupervised Domain Adaptation by Backpropagation”

Gradient Reversal Layer(GRL): Encoder(=>feature map) / Classifier(Domain) / Classifier(Class)



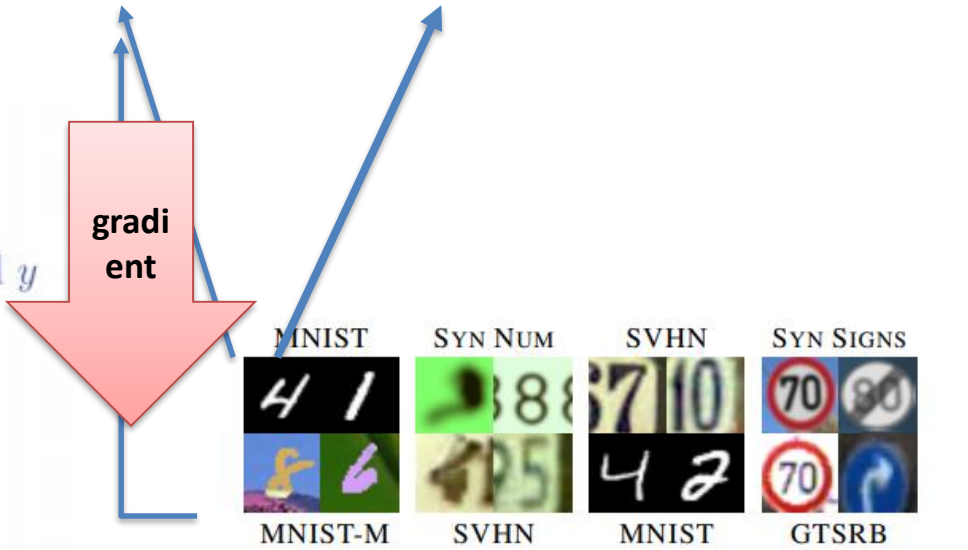


Figure 2. Examples of domain pairs (top – source domain, bottom – target domain) used in the small image experiments. See Section 4.1 for details.

임진혁<eeplearning@units.ac.kr>

Introduction: what is Domain Generalization?

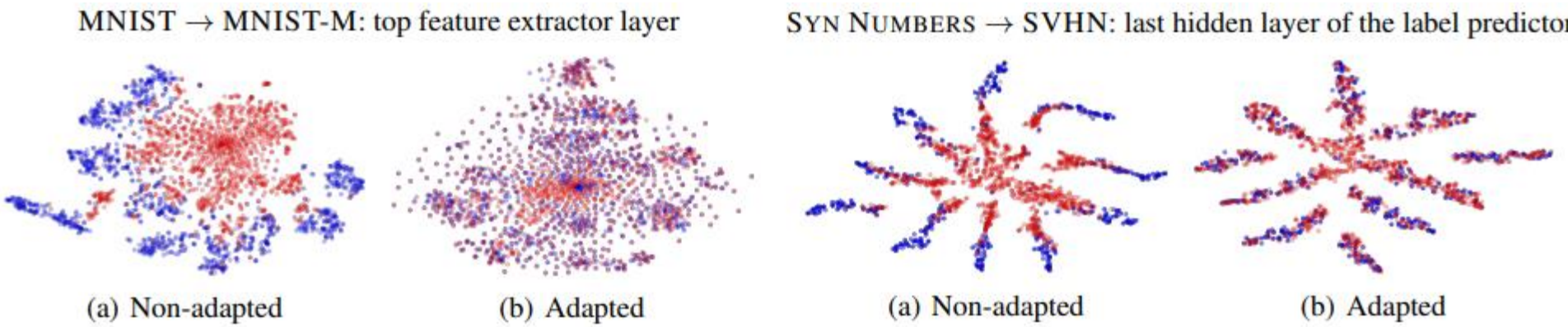
Domain Generalization의 문제 정의(Problem Setting)

“Domain Adoption”:

[1] Ganin et al,2015,“Unsupervised Domain Adaptation by Backpropagation”

Gradient Reversal Layer(GRL): Encoder(=>feature map) / Classifier(Domain) / Classifier(Class)

Domain을 일부러 못맞추게 학습하여 (gradient reversal : 반대 전파) source와 target을 구분하지 못하도록함



임진혁<eeplearning@units.ac.kr>

Introduction: what is Domain Generalization?

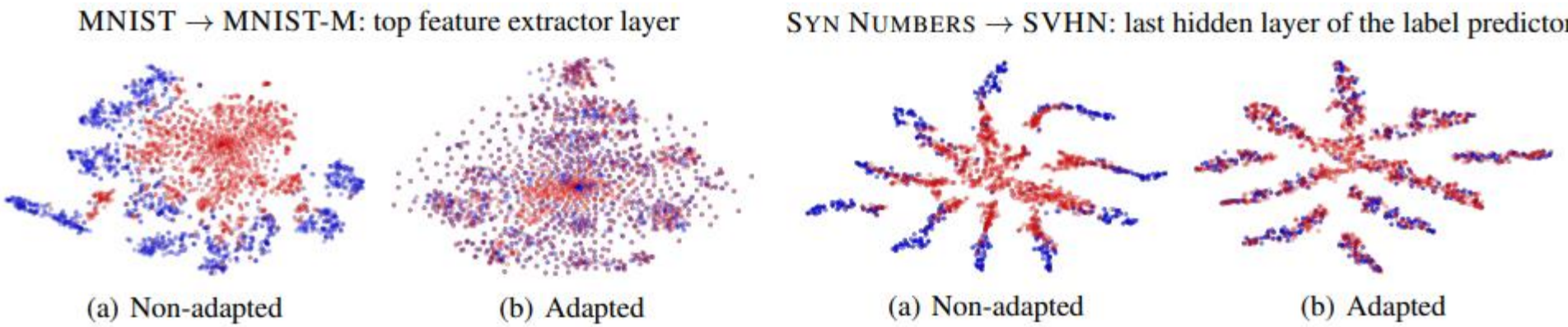
Domain Generalization의 문제 정의(Problem Setting)

“Domain Adoption”:

[1] Ganin et al,2015,“Unsupervised Domain Adaptation by Backpropagation”

Gradient Reversal Layer(GRL): Encoder(=>feature map) / Classifier(Domain) / Classifier(Class)

Domain을 일부러 못맞추게 학습하여 (gradient reversal : 반대 전파) source와 target을 구분하지 못하도록함



임진혁<eeplearning@units.ac.kr>

Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

“Domain Adoption”:

[2] Tzeng et al,2017,“Adversarial Discriminative Domain Adaptation”

Adversarial Learning: source CNN(Encoder) , Target CNN(Encoder), Discriminator(Domain) (also FM)

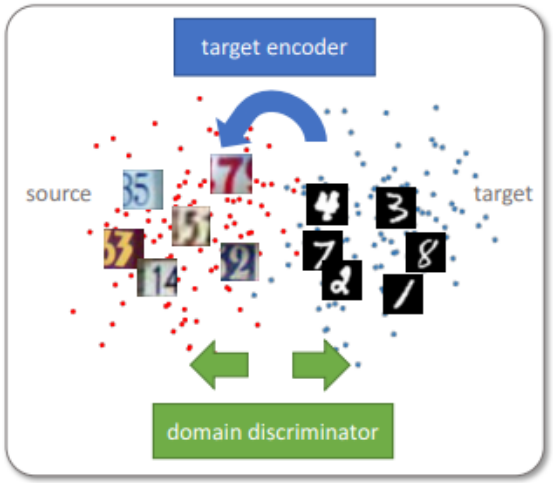


Figure 1: We propose an improved unsupervised domain adaptation method that combines adversarial learning with discriminative feature learning. Specifically, we learn a discriminative mapping of target images to the source feature space (target encoder) by fooling a domain discriminator that tries to distinguish the encoded target images from source examples.

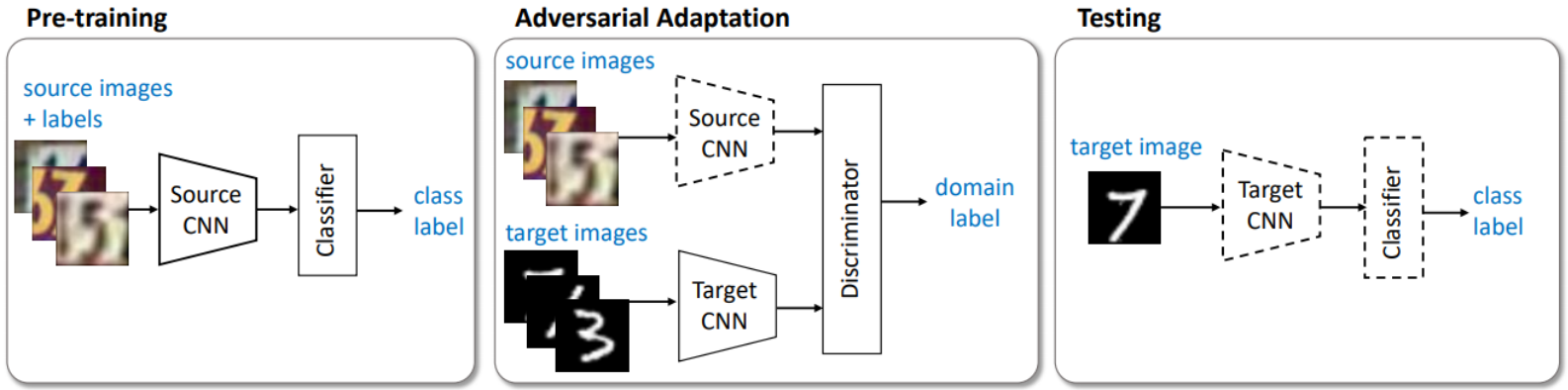


Figure 3: An overview of our proposed Adversarial Discriminative Domain Adaptation (ADDA) approach. We first pre-train a source encoder CNN using labeled source image examples. Next, we perform adversarial adaptation by learning a target encoder CNN such that a discriminator that sees encoded source and target examples cannot reliably predict their domain label. During testing, target images are mapped with the target encoder to the shared feature space and classified by the source classifier. Dashed lines indicate fixed network parameters.

1 MIIL Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

“Domain Adoption”:

[2] Tzeng et al,2017,“Adversarial Discriminative Domain Adaptation”

Adversarial Learning: source CNN(Encoder) , Target CNN(Encoder), Discriminator(Domain) (also FM)

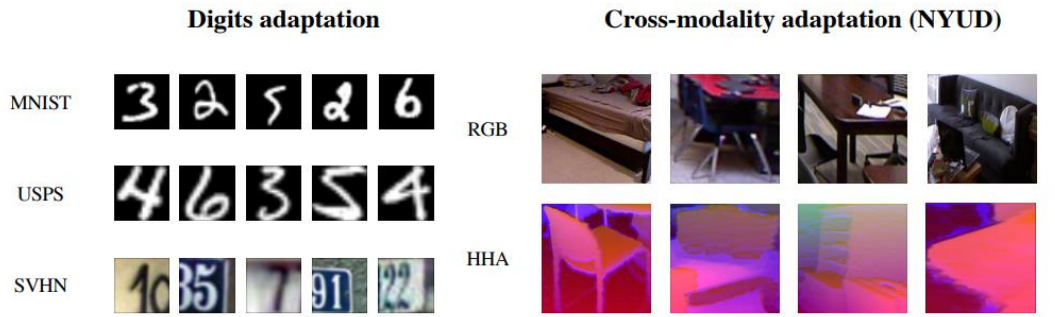


Figure 4: We evaluate ADDA on unsupervised adaptation across four domain shifts in two different settings. The first setting is adaptation between the MNIST, USPS, and SVHN datasets (left). The second setting is a challenging cross-modality adaptation task between RGB and depth modalities from the NYU depth dataset (right).

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST
Source only	0.752 ± 0.016	0.571 ± 0.017	0.601 ± 0.011
Gradient reversal	0.771 ± 0.018	0.730 ± 0.020	0.739 [16]
Domain confusion	0.791 ± 0.005	0.665 ± 0.033	0.681 ± 0.003
CoGAN	0.912 ± 0.008	0.891 ± 0.008	did not converge
ADDA (Ours)	0.894 ± 0.002	0.901 ± 0.008	0.760 ± 0.018

Table 2: Experimental results on unsupervised adaptation among MNIST, USPS, and SVHN.

임진혁<eeplearning@units.ac.kr>

Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

SO What is different from “Domain Adoption”?

임진혁<eeplearning@units.ac.kr>

Introduction: what is Domain Generalization?

Domain Generalization의 문제 정의(Problem Setting)

DG IS much harder

- **No (T) Data to analyze the distribution shift(Domain Shift) to overcome negative effects**
- **Only rely on the (S)**
- **Learning domain-invariant feature representation**

2 MIIL Related: Prior Domain Generalization

Learning domain-invariant feature representation(Generalizable features)

⇒ Prior Domain Generalization :

expose Model with variety of source domains as possible as many

➔ Reduce the burden for designing ALGORITHMS for DG

임진혁<eeplearning@unist.ac.kr>

2 Related: Prior Domain Generalization

Learning domain-invariant feature representation(Generalizable features)

⇒ Prior Domain Generalization :

expose Model with variety of source domains as possible as many

➔ Reduce the burden for designing ALGORITHMS for DG

⇒ Collecting data of large variety domains : high cost & impssible

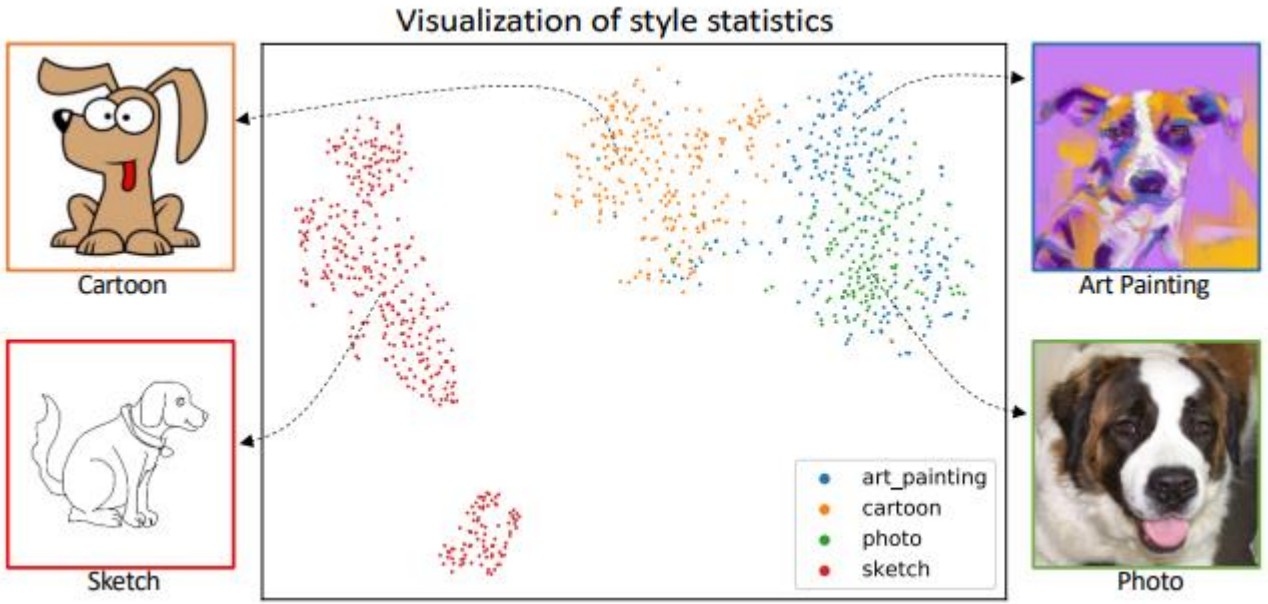
Propose *MixStyle* : mix style across source domians

Why?

Related: Prior Domain Generalization

Learning domain-invariant feature representation(Generalizable features)

visual domain is closely related to image style

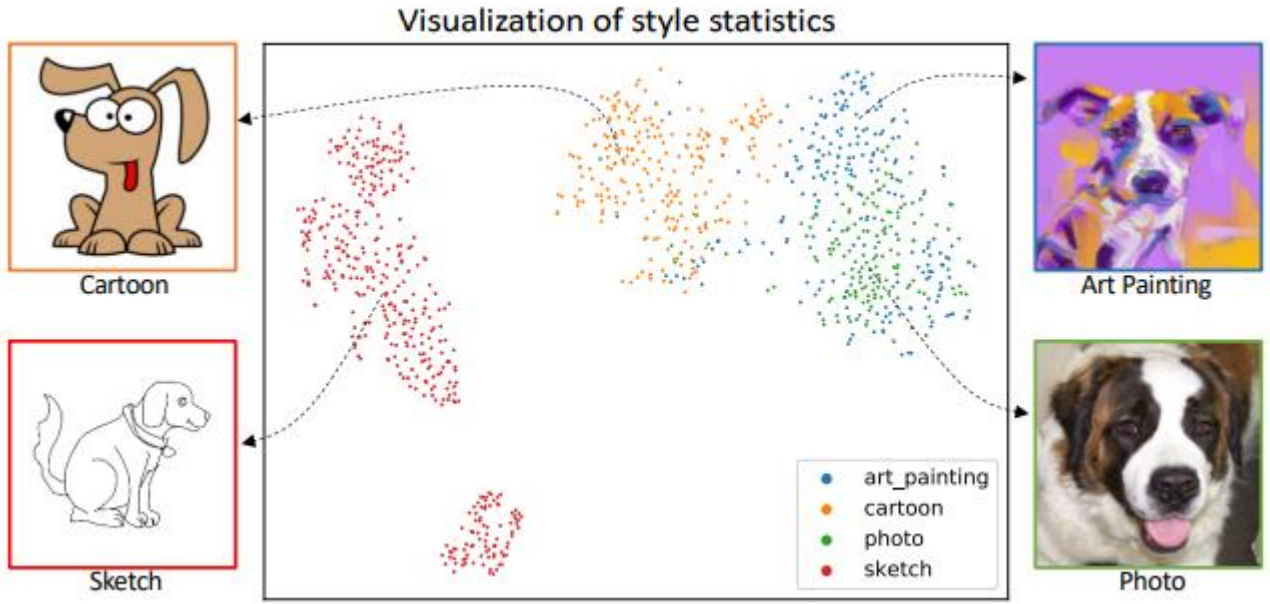


임진혁<eeplearning@unist.ac.kr>

Learning domain-invariant feature representation(Generalizable features)

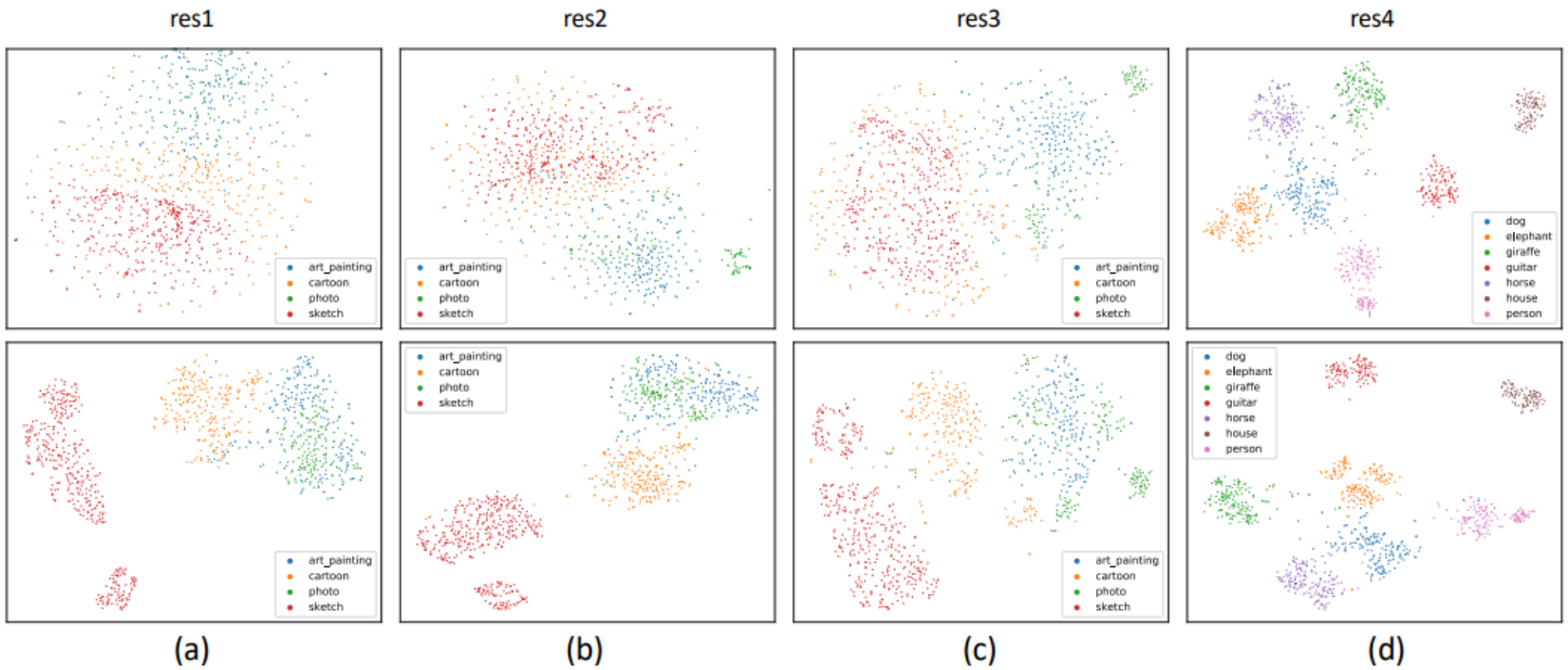
걍 다 박으면 어떻게 될까?

Style Research: (Huang & Belongie, 2017; Dumoulin et al., 2017)



Learning domain-invariant feature representation(Generalizable features)

"MixStyle can significantly improve CNNs' cross-domain generalization performance"



임진혁<eeplearning@unist.ac.kr>

Proposed Method: MixStyle

Very Simple : just make some normalization using "Mix"

$$\text{MixStyle}(x) = \gamma_{mix}\frac{x-\mu(x)}{\sigma(x)} + \beta_{mix}.$$

$$\gamma_{mix} = \lambda\sigma(x) + (1-\lambda)\sigma(\tilde{x}),$$
$$\beta_{mix} = \lambda\mu(x) + (1-\lambda)\mu(\tilde{x}),$$

Output: **style-normalized x**

λ ~ Beta(α, α)
α ∈ (0, ∞) (hyper-parameter)
set α to 0.1 (in this paper)

임진혁<eeplearning@unist.ac.kr>

Proposed Method: MixStyle

just make some normalization using “Mix”

Why didn’t use BN?

임진혁<eeplearning@unist.ac.kr>

Proposed Method: MixStyle

Why didn't use BN?

BN: save "mini – batch"(x)'s mean/variance and use it for normalization & scale/shift

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

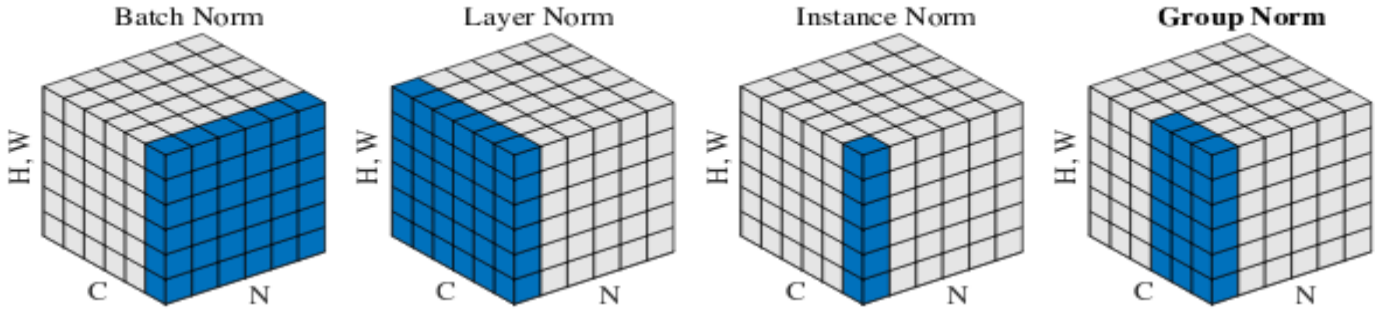
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

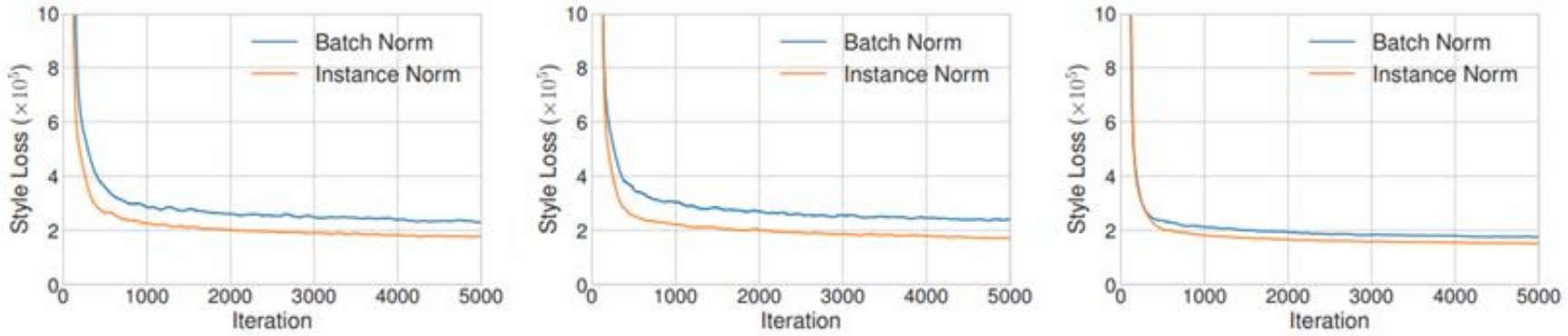
임진혁<eeplearning@unist.ac.kr>

3 MIIL Proposed Method: MixStyle

Why didn't use BN?



There is IN



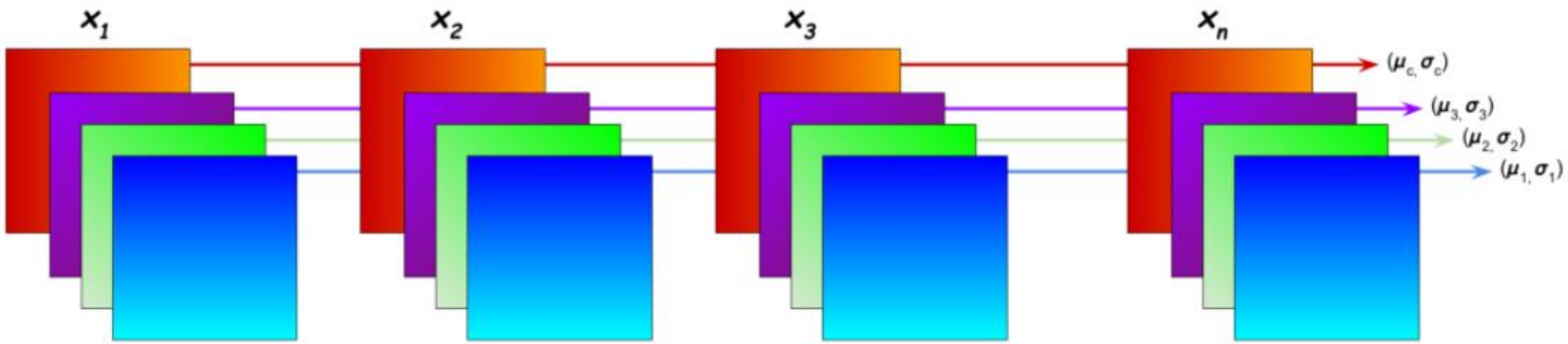
(a) Trained with original images. (b) Trained with contrast normalized images. (c) Trained with style normalized images.

Figure 1. To understand the reason for IN's effectiveness in style transfer, we train an IN model and a BN model with (a) original images in MS-COCO [36], (b) contrast normalized images, and (c) style normalized images using a pre-trained style transfer network [24]. The improvement brought by IN remains significant even when all training images are normalized to the same contrast, but are much smaller when all images are (approximately) normalized to the same style. Our results suggest that IN performs a kind of style normalization.

임진혁<eeplearning@unist.ac.kr>

3 MIIL Proposed Method: MixStyle

Batch Normalization



$$\mu_c = \frac{1}{NHW}\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{k=1}^{W} x_{icjk}$$

$$\sigma_c^2 = \frac{1}{NHW}\sum_{i=1}^{N}\sum_{j=1}^{H}\sum_{k=1}^{W}(x_{icjk} - \mu_c)^2$$

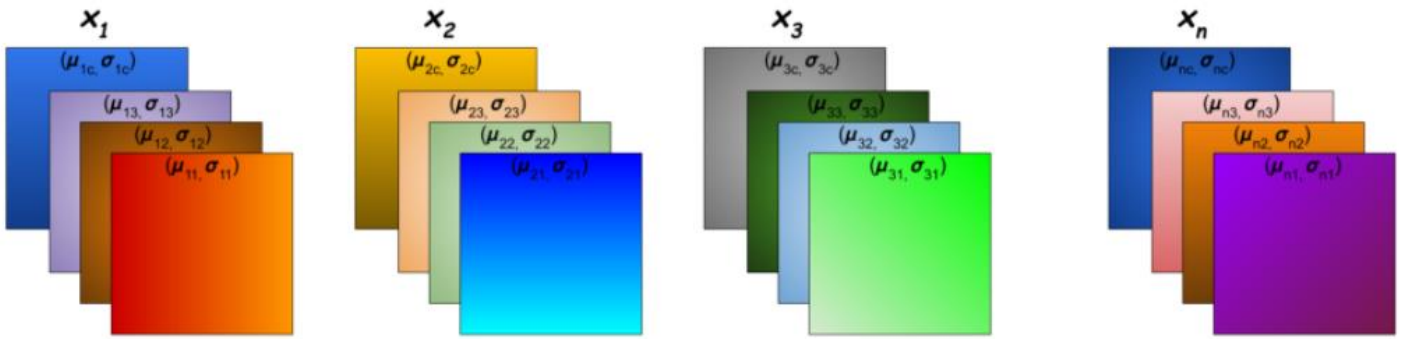
$$\hat{x} = \frac{x - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

In "*Batch Normalization*", mean and variance are calculated ***for*** each individual channel ***across*** all samples and both spatial dimensions.

임진혁<eeplearning@unist.ac.kr>

3 MIIL Proposed Method: MixStyle

Instance Normalization



$$\mu_{nc} = \frac{1}{HW}\sum_{j=1}^{H}\sum_{k=1}^{W} x_{ncjk}$$

$$\sigma_{nc}^2 = \frac{1}{HW}\sum_{j=1}^{H}\sum_{k=1}^{W}(x_{ncjk} - \mu_{nc})^2$$

$$\hat{x} = \frac{x - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \epsilon}}$$

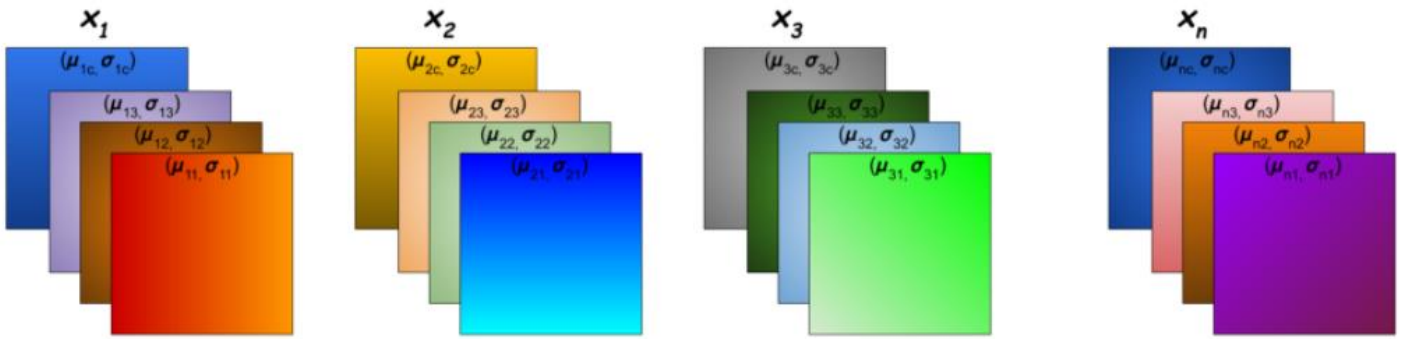
In "*Instance Normalization*", mean and variance are calculated ***for*** each individual channel ***for*** each individual sample ***across*** both spatial dimensions.

$$\mathbf{IN}(x) = \gamma\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \beta$$

임진혁<eeplearning@unist.ac.kr>

3 MIIL Proposed Method: MixStyle

Instance Normalization



$$\mu_{nc} = \frac{1}{HW}\sum_{j=1}^{H}\sum_{k=1}^{W} x_{ncjk}$$

$$\sigma_{nc}^2 = \frac{1}{HW}\sum_{j=1}^{H}\sum_{k=1}^{W}(x_{ncjk} - \mu_{nc})^2$$

$$\hat{x} = \frac{x - \mu_{nc}}{\sqrt{\sigma_{nc}^2 + \epsilon}}$$

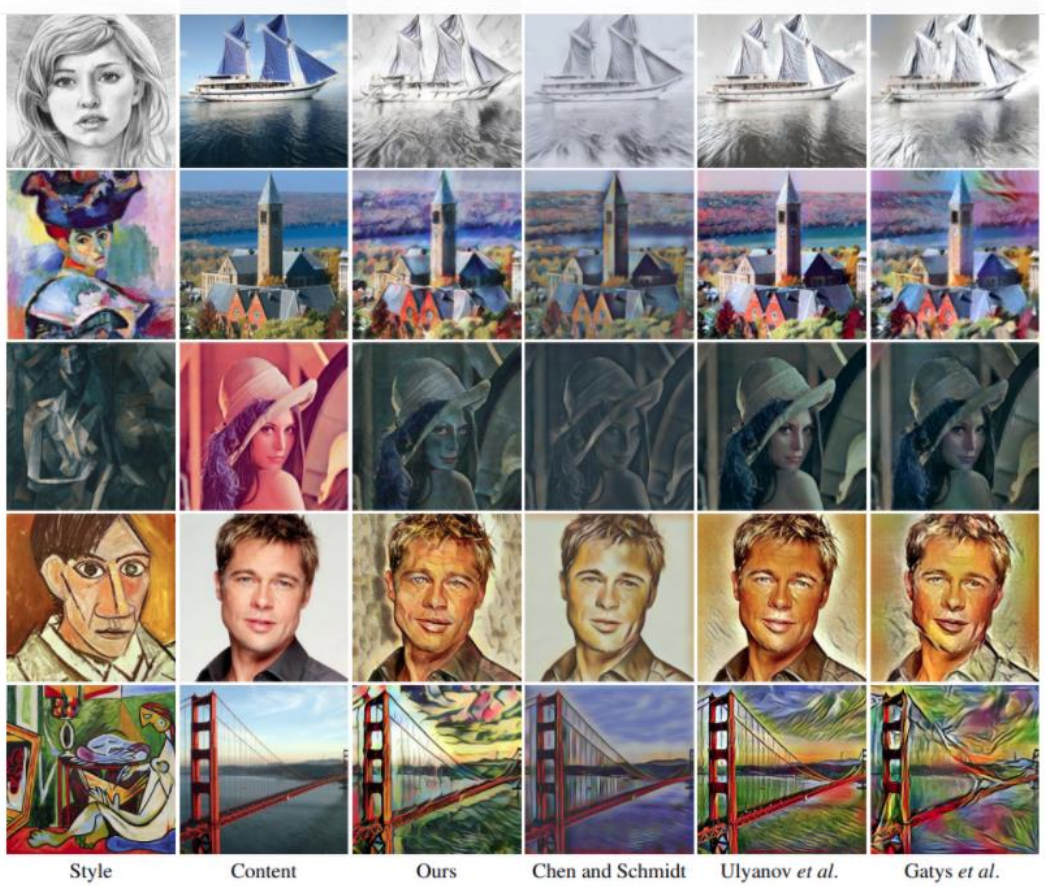
In "*Instance Normalization*", mean and variance are calculated ***for*** each individual channel ***for*** each individual sample ***across*** both spatial dimensions.

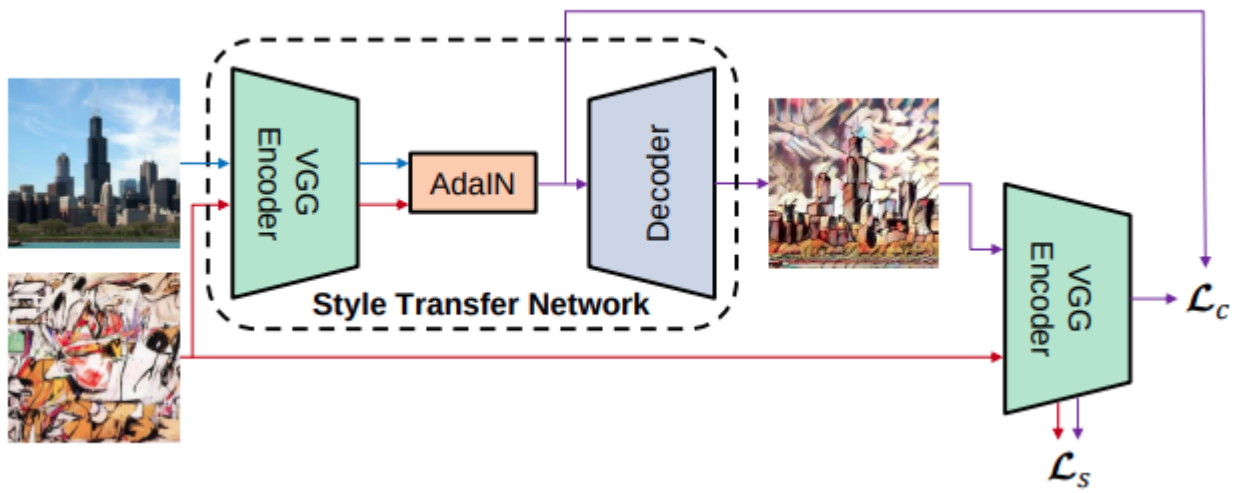
$$\mathbf{IN}(x) = \gamma\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \beta$$

임진혁<eeplearning@unist.ac.kr>

3 MIIL Proposed Method: MixStyle

$$AdaIN(x,y) = \sigma(y)\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \mu(y)$$





임진혁<eeplearning@unist.ac.kr>

Proposed Method: MixStyle

다시 말해서 :

regularizing CNN training by perturbing the style information of source domain training instances

쉽게 말해서 :

mixes feature statistics of two instances with a random convex weight to simulate new styles

임진혁<eeplearning@unist.ac.kr>

3 MIIL Proposed Method: MixStyle

쉽게 말해서 :

mixes feature statistics of two instances with a random convex weight to simulate new styles

(1)Given an input batch x

(2) MixStyle generates a reference batch x˜ from x

:x = [x(i) , x(j)] (x i and x j have the same batch size)

(3) x˜ is obtained by swapping the position of x(i) and x(j)

IF there is no label data?

x˜ = [Shuffle(x(j)), Shuffle(x(i))].

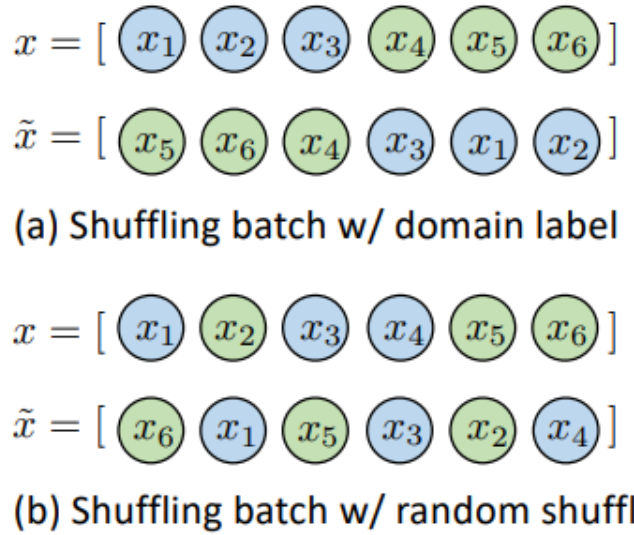


Figure 2: A graphical illustration of how a reference batch is generated. Domain label is denoted by color.

Proposed Method: MixStyle

Very Simple : just make some normalization using "Mix"

$$\text{MixStyle}(x) = \gamma_{mix}\frac{x-\mu(x)}{\sigma(x)} + \beta_{mix}.$$

$$\gamma_{mix} = \lambda\sigma(x) + (1-\lambda)\sigma(\tilde{x}),$$
$$\beta_{mix} = \lambda\mu(x) + (1-\lambda)\mu(\tilde{x}),$$

Output: **style-normalized x**

λ ~ Beta(α, α)
α ∈ (0, ∞) (hyper-parameter)
set α to 0.1 (in this paper)

임진혁<eeplearning@unist.ac.kr>

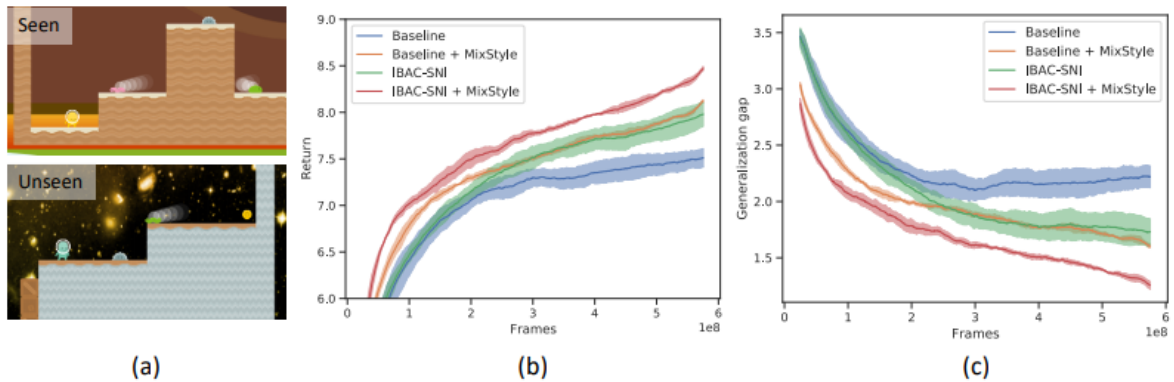
Table 1: Leave-one-domain-out generalization results on PACS.

Method	Art	Cartoon	Photo	Sketch	Avg
MMD-AAE	75.2	72.7	96.0	64.2	77.0
CCSA	80.5	76.9	93.6	66.8	79.4
JiGen	79.4	75.3	96.0	71.6	80.5
CrossGrad	79.8	76.8	96.0	70.2	80.7
Epi-FCR	82.1	77.0	93.9	73.0	81.5
Metareg	83.7	77.2	95.5	70.3	81.7
L2A-OT	83.3	78.2	96.2	73.6	82.8
ResNet-18	77.0±0.6	75.9±0.6	96.0±0.1	69.2±0.6	79.5
+ Manifold Mixup	75.6±0.7	70.1±0.9	93.5±0.7	65.4±0.6	76.2
+ Cutout	74.9±0.4	74.9±0.6	95.9±0.3	67.7±0.9	78.3
+ CutMix	74.6±0.7	71.8±0.6	95.6±0.4	65.3±0.8	76.8
+ Mixup (w/o label interpolation)	74.7±1.0	72.3±0.9	93.0±0.4	69.2±0.2	77.3
+ Mixup	76.8±0.7	74.9±0.7	95.8±0.3	66.6±0.7	78.5
+ DropBlock	76.4±0.7	75.4±0.7	95.9±0.3	69.0±0.3	79.2
+ MixStyle w/ random shuffle	82.3±0.2	**79.0**±0.3	**96.3**±0.3	73.8±0.9	82.8
+ MixStyle w/ domain label	**84.1**±0.4	78.8±0.4	96.1±0.3	**75.9**±0.9	**83.7**

임진혁<eeplearning@unist.ac.kr>

Table 2: Generalization results on the cross-dataset person re-ID task.

Model	Market1501→Duke				Duke→Market1501			
	mAP	R1	R5	R10	mAP	R1	R5	R10
ResNet-50	19.3	35.4	50.3	56.4	20.4	45.2	63.6	70.9
+ RandomErase	14.3	27.8	42.6	49.1	16.1	38.5	56.8	64.5
+ DropBlock	18.2	33.2	49.1	56.3	19.7	45.3	62.1	69.1
+ MixStyle w/ random shuffle	**23.8**	42.2	58.8	**64.8**	24.1	51.5	69.4	76.2
+ MixStyle w/ domain label	23.4	**43.3**	**58.9**	64.7	**24.7**	**53.0**	**70.9**	**77.8**
OSNet	25.9	44.7	59.6	65.4	24.0	52.2	67.5	74.7
+ RandomErase	20.5	36.2	52.3	59.3	22.4	49.1	66.1	73.0
+ DropBlock	23.1	41.5	56.5	62.5	21.7	48.2	65.4	71.3
+ MixStyle w/ random shuffle	27.2	**48.2**	**62.7**	**68.4**	27.8	58.1	74.0	**81.0**
+ MixStyle w/ domain label	**27.3**	47.5	62.0	67.1	**29.0**	**58.2**	**74.9**	80.9



(a) (b) (c)

Figure 3: (a) Coinrun benchmark. (b) Test performance in unseen environments. (c) Difference between training and test performance.

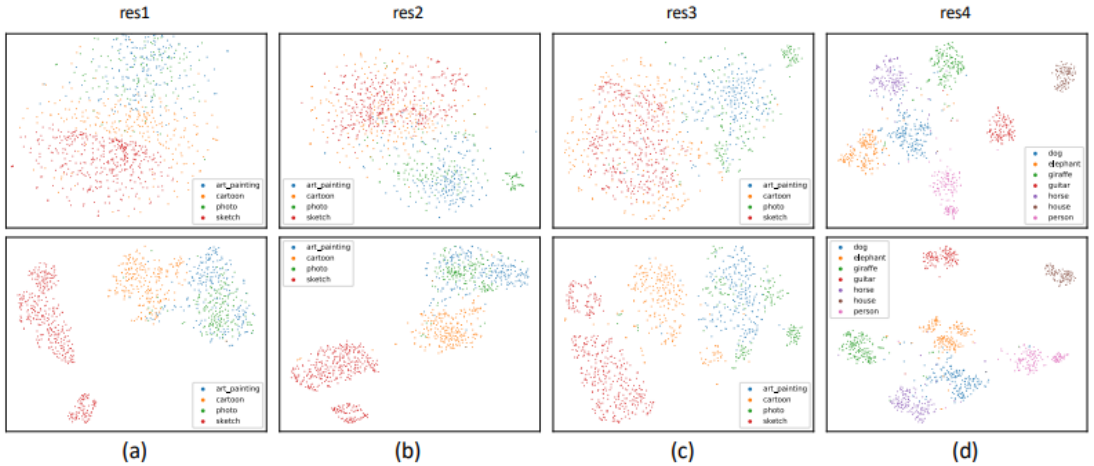


Figure 4: 2-D visualization of flattened feature maps (top) and the corresponding style statistics (bottom). `res1-4` denote the four residual blocks in order in a ResNet architecture. We observe that `res1` to `res3` contain domain-related information while `res4` encodes label-related information.

Table 3: Ablation study on where to apply MixStyle in the ResNet architecture.

(a) Category classification on PACS.

Model	Accuracy
ResNet-18	79.5
+ MixStyle (`res1`)	80.1
+ MixStyle (`res12`)	81.6
+ MixStyle (`res123`)	**82.8**
+ MixStyle (`res1234`)	75.6
+ MixStyle (`res14`)	76.3
+ MixStyle (`res23`)	81.7

(b) Cross-dataset person re-ID.

Model	mAP
ResNet-50	19.3
+ MixStyle (`res1`)	22.6
+ MixStyle (`res12`)	**23.8**
+ MixStyle (`res123`)	22.0
+ MixStyle (`res1234`)	10.2
+ MixStyle (`res14`)	11.1
+ MixStyle (`res23`)	20.6

캄사…

임진혁<eeplearning@units.ac.kr>