Enhancing the Reliability of Out-of-Distribution Image Detection in Neural Networks

ICLR 2018

Shiyu Liang, Yixuan Li, R. Srikant

01. Introduction

✓ 최근 Neural Network는 동일한 분포에서 샘플링 된 학습/테스트 데이터에서 상당히 일반화가 잘 되어 있음

✓ 하지만, 이를 실제 Application에 적용될 때, Testing Data Distribution에 대한 접근은 적음

✓ 최근 발표된 논문에 따르면, **Unrecognizable** 또는 **Irrelevant** 입력에 대해 Neural Network가 높은 Confidence Prediction을 만드는 경향이 있음

✓ 이러한 새로운 유형의 불확실성(i.e. **Out-of-Distribution**)을 처리할 필요성이 존재함

✓ 가장 직관적인 방법은 학습 데이터에 In/Out-of-Distribution 예제를 함께 포함하여 학습하는 것임

-> ODD 샘플은 매우 다양함, 재학습은 Computational Expensive, Intractable

01. Introduction

Anomaly Detection의 용어 분류 방법

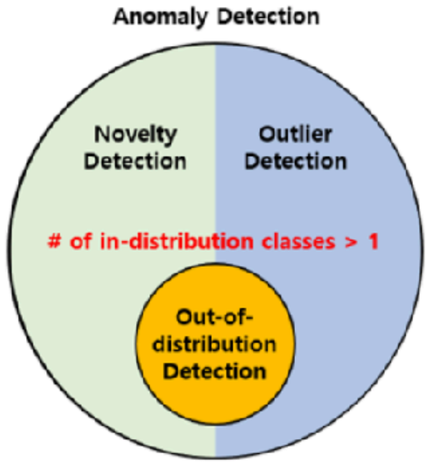
1. 학습시 비정상 sample의 사용 여부 및 label 유무에 따른 분류

용어	정상 sample	비정상 sample
Supervised Anomaly Detection	학습에 사용	학습에 사용
Semi-Supervised (One-Class) Anomaly Detection	학습에 사용	**학습에 사용 X**
Unsupervised Anomaly Detection	**모름.(label이 없음)** 학습에 사용하는 데이터의 대다수가 정상 sample일 것이라고 가정.	

2. 비정상 sample 정의에 따른 분류

용어	비정상 sample
Novelty Detection	지금까지 등장하지 않았지만 충분히 등장할 수 있는 sample
Outlier Detection	지금까지 등장하지 않았고 앞으로도 등장할 가능성이 없는, 데이터에 오염이 발생했을 가능성이 있는 sample

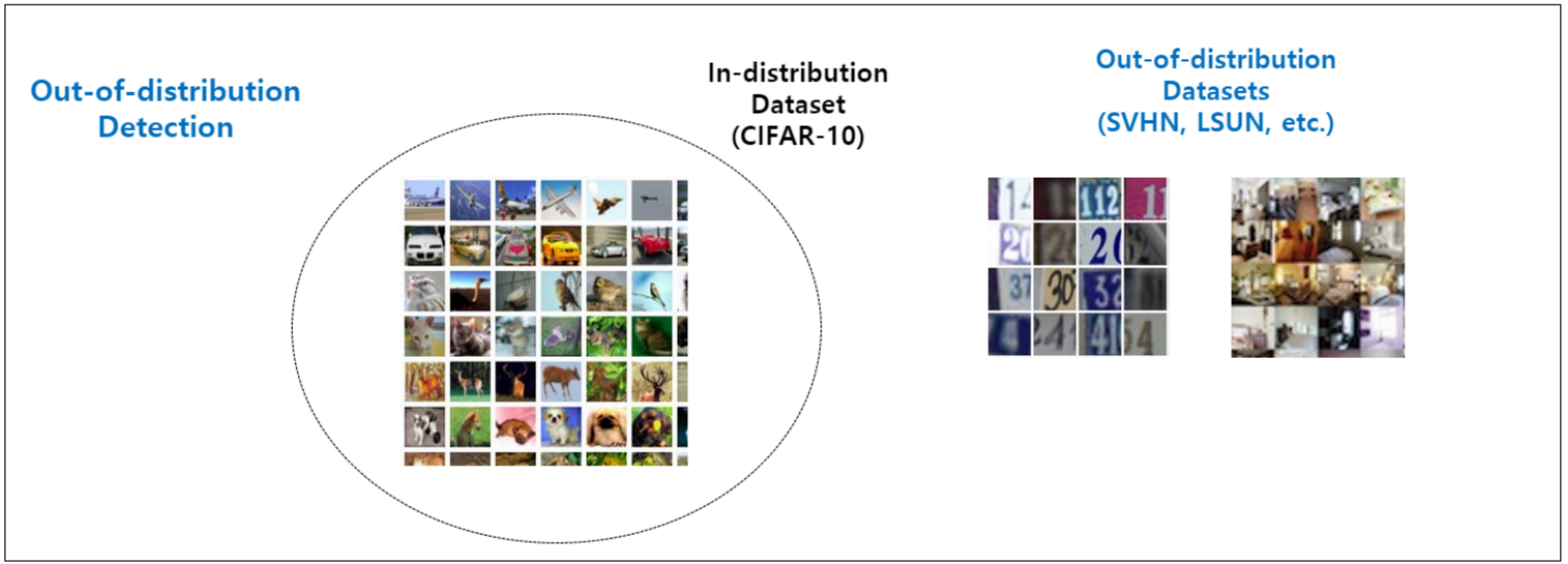
3. 정상 sample의 class 개수에 따른 분류



[Anomaly Detection 관련 3가지 용어의 분류 방법 정리]

01. Introduction

정상 샘플의 클래스 개수에 따른 분류



[Out-of-distribution sample 예시]

01. Introduction

✓ 간단하고 효율적인 ODIN(Out-of-Distribution detector for Neural Networks)을 제안함

Neural Network에 대한 추가 학습 필요 없음, 어떠한 Neural Network에도 쉽게 적용이 가능함

✓ 다양한 In/Out-of-Distribution 데이터셋과 SOTA Architecture(e.g. DenseNet, Wide ResNet)에서 ODIN을 실험함

기존의 방법론을 뛰어넘는 성능을 보여줌

✓ 실험적으로 어떠한 Parameter 설정이 ODIN의 성능에 영향을 미치는지 분석함

02. Problem Statement

Problem: 사전 학습된 Neural Network에서 In/Out-of-Distribution 구분

Neural Network f Image Space X

In-Distribution P_X Out-Distribution Q_X Mixture Distribution $P_{X\times Z}$

✓ OOD Detection뿐만 아니라 In-Distribution 이미지를 정확한 클래스로 분류하는 것도 중요함

이미지가 In-Distribution으로 구분되면, 단순히 원본 이미지를 통해 Neural Network에서 분류함

03. Method

Temperature Scaling

KD 방법론으로 In-Distribution과 Out-Distribution 샘플 간의 Softmax Score를 멀어지게 하여 구분을 쉽게 함

$$S_i(\boldsymbol{x};T) = \frac{\exp\left(f_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^{N}\exp\left(f_j(\boldsymbol{x})/T\right)},$$

Input Preprocessing

FGSM의 방법론으로 역으로 True Label에 대한 Softmax Socre를 높여줌

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \mathrm{sign}(-\nabla_{\boldsymbol{x}} \log S_{\hat{y}}(\boldsymbol{x};T)),$$

OOD Detector

$$g(\boldsymbol{x};\delta,T,\varepsilon) = \begin{cases} 1 & \text{if } \max_i p(\tilde{\boldsymbol{x}};T) \leq \delta, \\ 0 & \text{if } \max_i p(\tilde{\boldsymbol{x}};T) > \delta. \end{cases}$$

04. Experiment

Model

Architecture	C-10	C-100
Dense-BC	4.81	22.37
WRN-28-10	3.71	19.86

Table 1: Test error rates on CIFAR-10 and CIFAR-100 datasets.

OOD Test Dataset

(1) **TinyImageNet.** The Tiny ImageNet dataset[3] consists of a subset of ImageNet images (Deng et al., 2009). It contains 10,000 test images from 200 different classes. We construct two datasets, *TinyImageNet (crop)* and *TinyImageNet (resize)*, by either randomly cropping image patches of size 32×32 or downsampling each image to size 32×32.

(2) **LSUN.** The Large-scale Scene UNderstanding dataset (LSUN) has a testing set of 10,000 images of 10 different scenes categories such as *bedroom, kitchen room, living room, etc.* (Yu et al., 2015). Similar to TinyImageNet, we construct two datasets, *LSUN (crop)* and *LSUN (resize)*, by randomly cropping and downsampling the LSUN testing set, respectively.

(3) **Gaussian Noise.** The synthetic Gaussian noise dataset consists of 10,000 random 2D Gaussian noise images, where each RGB value of every pixel is sampled from an i.i.d Gaussian distribution with mean 0.5 and unit variance. We further clip each pixel value into the range $[0, 1]$.

(4) **Uniform Noise.** The synthetic uniform noise dataset consists of 10,000 images where each RGB value of every pixel is independently and identically sampled from a uniform distribution on $[0, 1]$.

04. Experiment

In-Distribution 샘플의 TPR이 95%가 되는 시점에서의 HyperParameter 설정

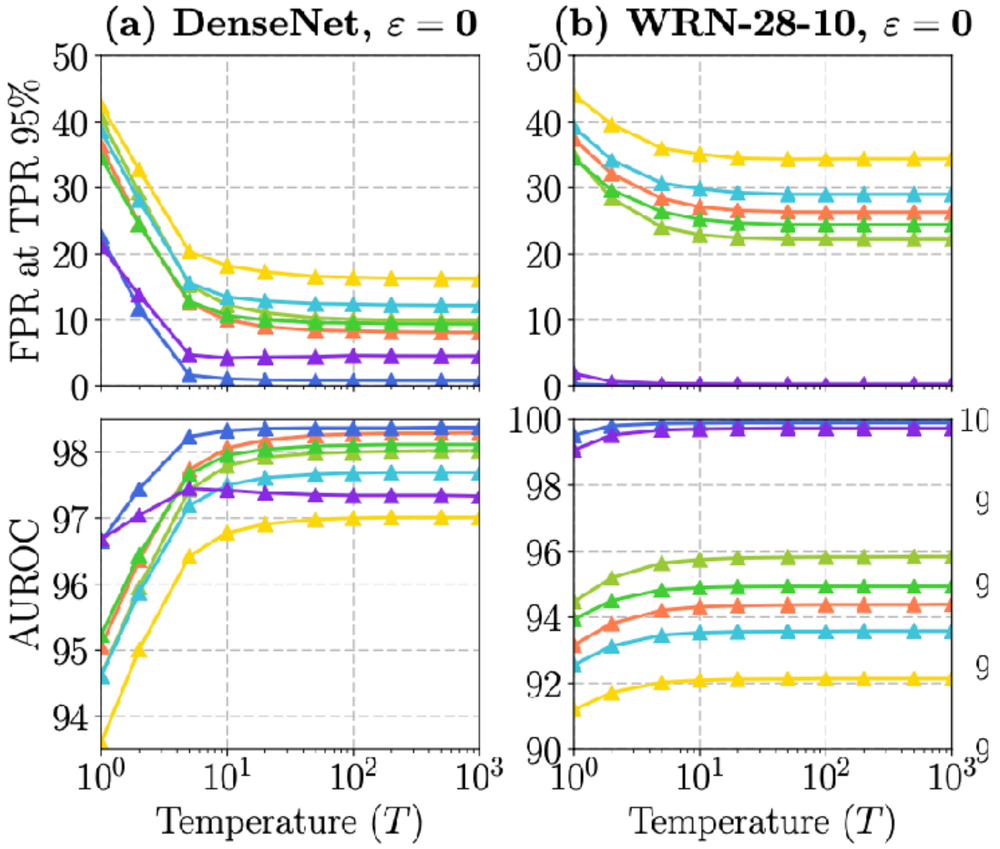
T = 1000, Epsilon(C10) = 0.0014, Epsilon(C100) = 0.002

	Out-of-distribution dataset	FPR (95% TPR) ↓	Detection Error ↓	AUROC ↑	AUPR In ↑	AUPR Out ↑
		Baseline (Hendrycks & Gimpel, 2017) / ODIN				
Dense-BC CIFAR-10	TinyImageNet (crop)	34.7/**4.3**	10.0/**4.7**	95.3/**99.1**	96.4/**99.1**	93.8/**99.1**
	TinyImageNet (resize)	40.8/**7.5**	11.5/**6.1**	94.1/**98.5**	95.1/**98.6**	92.4/**98.5**
	LSUN (crop)	39.3/**11.4**	10.2/**7.2**	94.8/**97.9**	96.0/**98.0**	93.1/**97.9**
	LSUN (resize)	33.6/**3.8**	9.8/**4.4**	95.4/**99.2**	96.4/**99.3**	94.0/**99.2**
	Uniform	23.5/**0.0**	5.3/**0.5**	96.5/**99.0**	97.8/**100.0**	93.0/**99.0**
	Gaussian	12.3/**0.0**	4.7/**0.2**	97.5/**100.0**	98.3/**100.0**	95.9/**100.0**
Dense-BC CIFAR-100	TinyImageNet (crop)	67.8/**26.9**	36.4/**12.9**	83.0/**94.5**	85.3/**94.7**	80.8/**94.5**
	TinyImageNet (resize)	82.2/**57.0**	43.6/**22.7**	70.4/**85.5**	71.4/**86.0**	68.6/**84.8**
	LSUN (crop)	69.4/**18.6**	37.2/**9.7**	83.7/**96.6**	86.2/**96.8**	80.9/**96.5**
	LSUN (resize)	83.3/**58.0**	44.1/**22.3**	70.6/**86.0**	72.5/**87.1**	68.0/**84.8**
	Uniform	100.0/**100.0**	35.86/**17.9**	43.1/**99.5**	63.2/**87.5**	41.9/**65.1**
	Gaussian	100.0/**100.0**	41.2/**38.0**	30.6/**40.5**	53.4/**60.5**	37.6/**40.9**

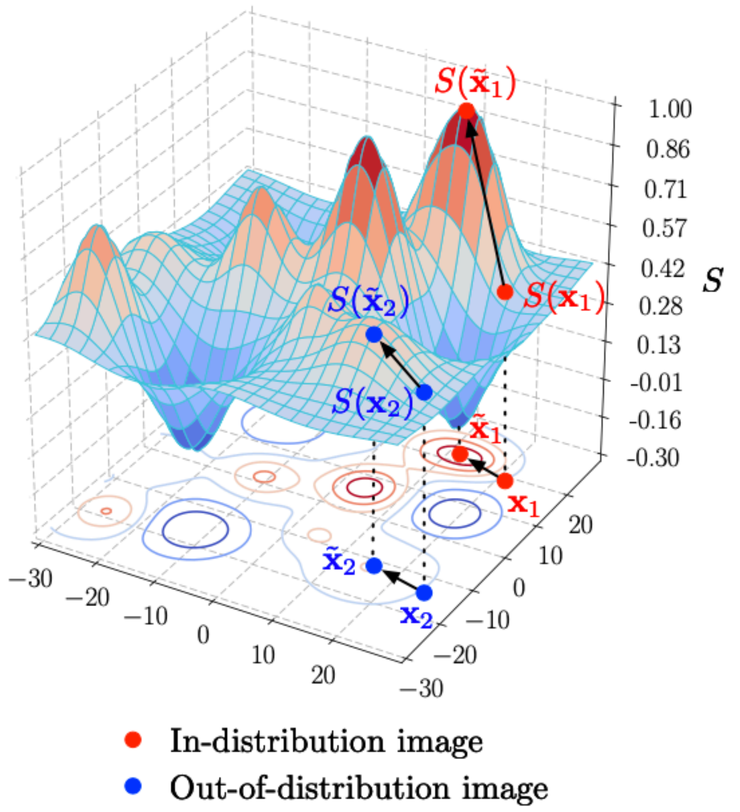
Table 2: Distinguishing in- and out-of-distribution test set data for image classification. All values are percentages. ↑ indicates larger value is better, and ↓ indicates lower value is better. We use $T = 1000$ for all experiments. The noise magnitude ε was selected on a **separate validation dataset**, which is different from the out-of-distribution test sets. On CIFAR-10 pretrained model, we use $\varepsilon = 0.0014$ for all OOD test datasets; and $\varepsilon = 0.002$ for CIFAR-100 pretrained model.

05. Discussion

Temperature의 영향



Gradient의 영향



06. Conclusion

- ✓ 간단하고 효율적인 OOD Detection 방법인 ODIN을 제안함
- ✓ 추가적인 학습이 필요 없을뿐만 아니라 기존의 방법보다 성능이 뛰어남
- ✓ 다양한 모델 아키텍쳐와 데이터셋에 대해 실험적으로 Parameter를 설정하고 이를 통한 Insight를 제공함
- ✓ 추후에는 Speech Recognition, NLP 분야에서 해당 실험을 진행할 예정임

07. References

https://arxiv.org/pdf/1706.02690.pdf

https://hoya012.github.io/blog/anomaly-detection-overview-1/

https://hoya012.github.io/blog/anomaly-detection-overview-2/