Contextualised Out-of-Distribution Detection using Pattern Identification

Romain Xu-Darme (CEA LIST),
Julien Girard-Satabin (CEA LIST),
Darryl Hond (Thales UK, Research, Technology and Innovation),
Gabriele Incorvaia (Thales UK, Research, Technology and Innovation),
Zakaria Chihani (CEA LIST)
Correspondance: julien.girard2@cea.fr

Out-of-Distribution and why is it hard

Core assumption in Deep Learning

For image classification: a neural network is trained on a dataset X_{train} drafted from an (unknown) distribution $\mathscr{D}$

What *should* happen when $X_{production}$ is not drawn from $\mathscr{D}$?

Out-of-Distribution detection

"How to know where we will meet dragons?"

Spotting that $x \in X_{production} \nsim \mathscr{D}$:
Out-of-Distribution detection

Challenges

1. defining Out-of-Distribution?
2. leveraging Inside-of-Distribution definition?
3. reliability on the base model ?
4. ensuring your Out-of-Distribution detection is reliable?
5. justifying the Out-of-Distribution-ness of a sample to a user?

Explainable AI for Out-of-Distribution detection

Our approach

Leveraging explainable AI: *representation learning*

1. Learning *recurring patterns* in the latent space of the classifier
2. Compute a *confidence score* based on how new patterns are correlated

Contextualized Out-of-Distribution Detection using Pattern Identification (CODE)

CODE overview

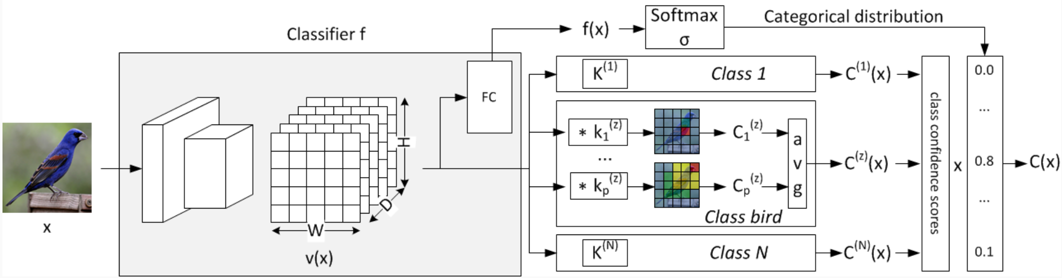


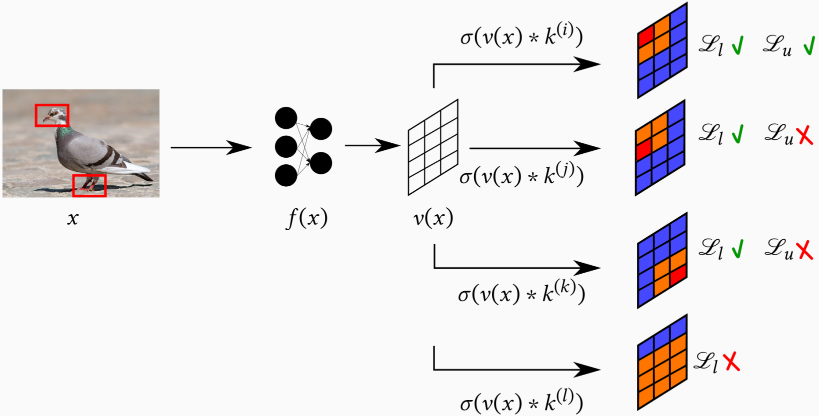
Figure 1: CODE inference overview. When processing a new sample x, the confidence measure sums up the average contribution of the detectors from each class weighted by the probability of x belonging to that class.

Cracking the CODE open - ingredients

Ingredients:

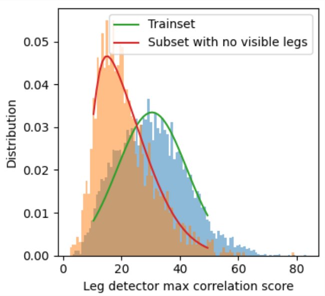
1. a neural network f and v its restriction to the last convolutional layer, of size $H \times W \times D$
2. p 1×1 convolutional kernels per class (*detectors*), of size $1 \times 1 \times D$
3. a vizualisation technique (e.g. SmoothGrads)

Cracking the CODE open - recipe



Kernels needs to: (1) correlates on a small part of the image (locality constraint $\mathscr{L}_l$) and (2) correlate to multiple activation locations (unicity constraint $\mathscr{L}_u$)

Cracking the CODE open - recipe

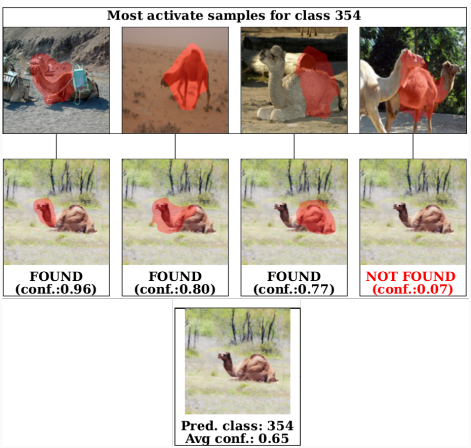


Maximum correlation score $H_i^{(c)}(x) = \max_{v^* \in v(x)} (v^* * k_i^{(c)})$. From [Xu-+22].

Given a new x', correlation between $v(x')$ with the distribution of $v(x), x \in X_{iod}$

Key points

1. *Does not* require an explicit definition of Out-of-Distribution
2. *Does not* require retraining an existing model
3. *Does* provide a visual clue along with the confidence score



Comparing Out-of-Distribution methods: an open problem

Existing approaches

Plenty of methods: adding a new score in the mix is of little interest if we cannot compare it!

Existing methods either require retraining of the classifier (Bayesian based approaches) or an explicit definition of Out-of-Distribution (which is usually not available at inference time)

Existing comparisons

Cross-dataset validation:

1. train/calibrate a score on a dataset X_{iod}
2. evaluate it on another dataset considered Out-of-Distribution: X_{ood}

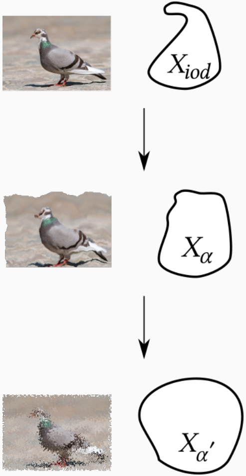
Questions:

1. how is Out-of-Distribution defined?
2. how to assert the quality of a score *independently* of X_{iod} and X_{ood}?

What is a good Out-of-Distribution metric?

Hypothesis: metrics should increase with an increased *Out-of-Distribution-ness* (*e.g.* gradually increasing blur on all CIFAR-100 images)

If multiple metrics are similarly correlated for the same perturbation, it shows they capture the *increase* in Out-of-Distribution-ness



Evaluation

1. OpenOOD [Yan+22] suite for **cross-dataset Out-of-Distribution evaluation** to compare methods across multiple datasets; evaluating using *Area Under the Receiving Operator Curve (AUROC) score*
2. Increasingly perturbating all images from a dataset X_{iod} using blur, gaussian noise, brightness and rotations for **consistencty of metrics under perturbation**; evaluating using *Spearman Rank Correlation (SRC) score*

Cross-dataset Out-of-Distribution evaluation

	OSR					OoD Detection (Near-OoD / Far-OoD)				
	M-6	C-6	C-50	T-20	Avg.	MNIST	CIFAR-10	CIFAR-100	ImageNet	Avg.
MSP* [HG17]	96.2	85.3	81.0	73.0	83.9	91.5 / 98.5	86.9 / 89.6	80.1 / 77.6	69.3 / 86.2	81.9 / 87.9
ODIN* [LLS18]	**98.0**	72.1	80.3	**75.7**	81.8	92.4 / 99.0	77.5 / 81.9	79.8 / 78.5	73.2 / 94.4	80.7 / 88.4
MDS* [Lee+18]	89.8	42.9	55.1	57.6	62.6	**98.0** / 98.1	66.5 / 88.8	51.4 / 70.1	68.3 / 94.0	71.0 / 87.7
Gram* [SO20]	82.3	61.0	57.5	63.7	66.1	73.9 / **99.8**	58.6 / 67.5	55.4 / 72.7	68.3 / 89.2	64.1 / 82.3
MaxLogit* [Hen+22]	**98.0**	84.8	**82.7**	75.5	85.3	92.5 / 99.1	86.1 / 88.8	**81.0** / 78.6	73.6 / 92.3	83.3 / 89.7
KNN* [Sun+22]	97.5	**86.9**	**83.4**	74.1	**85.5**	96.5 / 96.7	**90.5** / **92.8**	79.9 / 82.2	**80.8** / 98.0	**86.9** / 92.4
FNRD [Hon+21]	59.4	68.2	58.4	54.3	60.1	84.8 / 97.1	70.2 / 71.5	54.6 / 58.5	75.4 / 87.5	71.3 / 78.7
CODE (p=4)	74.7	86.7	76.5	62.4	75.1	81.8 / 99.5	87.8 / 90.7	73.9 / 72.4	76.6 / 84.4	80.0 / 86.8

Table 1: Partial comparison of AUROC scores between CODE and state-of-the-art methods on a cross-dataset benchmark. Results with * are extracted from [Yan+22] - keeping only OoD-agnostic methods. **bold** = higher score.

Consistency of CODE

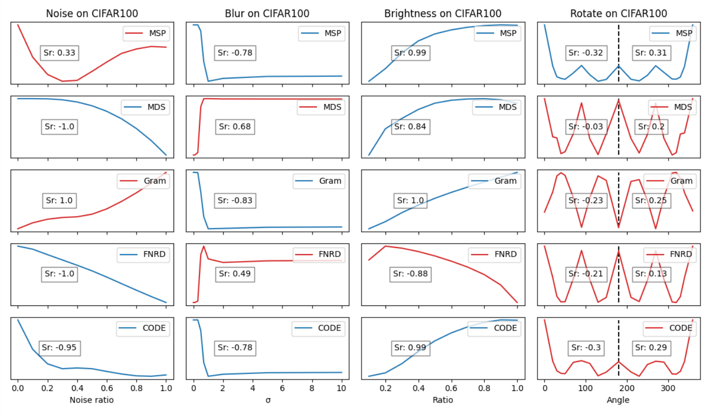


Figure 2: Evolution of the average confidence score v. magnitude of the perturbation.

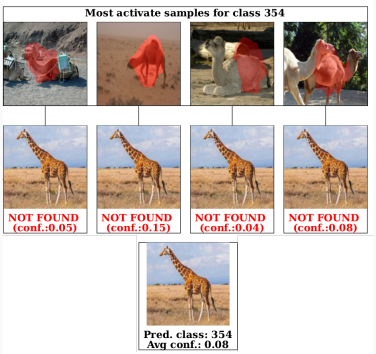
Curves in red indicate anomalous behaviours.

Consistency of CODE

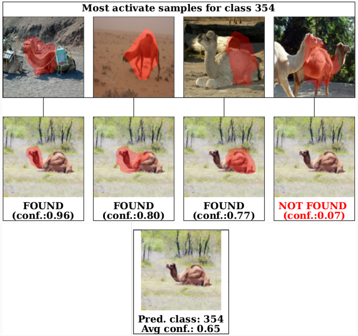
	CIFAR10					CIFAR100					ImageNet					Avg.
	Noise ↓	Blur ↓	Bright. ↑	R+ ↓	R- ↑	Noise ↓	Blur ↓	Bright. ↑	R+ ↓	R- ↑	Noise ↓	Blur ↓	Bright. ↑	R+ ↓	R- ↑	
MSP	-0.22	-0.88	**0.98**	-0.55	0.56	0.33	-0.78	**0.99**	-0.32	0.31	0.71	**-1.0**	**1.0**	-0.77	0.85	0.54
ODIN	-0.85	-0.7	0.18	-0.15	0.13	-0.15	-0.77	0.75	-0.22	0.21	0.12	-0.87	0.2	-0.81	0.81	0.45
MDS	**-1.0**	0.41	0.84	-0.03	0.19	**-1.0**	0.68	0.84	-0.03	0.2	**-1.0**	0.98	-0.35	-0.16	0.11	0.20
Gram	1.0	**-1.0**	**1.0**	-0.15	-0.02	1.0	-0.83	**1.0**	-0.23	0.25	🕒	🕒	🕒	🕒	🕒	0.24*
MaxLogit	-0.62	-0.88	**0.96**	-0.33	0.33	0.0	-0.78	**0.99**	-0.22	0.22	0.65	**-0.93**	**1.0**	-0.78	0.78	0.54
KNN	-0.36	-0.88	**0.99**	-0.46	0.4	-0.02	-0.79	**1.0**	-0.26	0.35	**-0.99**	**-1.0**	**1.0**	-0.5	0.5	0.63
FNRD	**-1.0**	0.58	-0.99	-0.11	0.08	**-1.0**	0.49	-0.88	-0.21	0.13	**-1.0**	-0.85	**0.99**	-0.35	0.35	0.21
CODE	-0.69	-0.88	**1.0**	-0.5	0.35	**-0.95**	-0.78	**0.99**	-0.3	0.29	-0.85	**-0.93**	**1.0**	-0.85	0.83	**0.75**

Table 2: Partial comparison of OoD methods on our perturbation benchmark. ↑ (resp. ↓) = average confidence should increase (resp. decrease) with α. red = weak correlation or unexpected sign of the correlation coefficient. **bold** = strong expected correlation. 🕒 = timeout. Note that CODE is consistent accross almost all perturbations.

Contextualized Out-of-Distribution



(a) Out-of-distribution image.



(b) Inside-Of-Distribution image.

Figure 3: Explanations generated by CODE for ID and OoD samples.

Limitations and future steps

1. CODE provides an Out-of-Distribution detection score that is consistent accross two Out-of-Distribution modalities
2. We provide a benchmark for Out-of-Distribution score consistency checking
3. CODE is as good as the neural network internal representation
4. Contextualizations can only tell you so much (what happen when there is a high confidence score, but the upsampling matches not our expectations?)
5. Applications to other problem classes (object detection, time series)

References

[Hen+22] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Xiaodong Song. "Scaling Out-of-Distribution Detection for Real-World Settings". In: *International Conference on Machine Learning*. 2022 (cit. on p. 19).

Bibliography ii

[HG17] Dan Hendrycks and Kevin Gimpel. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017 (cit. on p. 19).

[Hon+21] Darryl Hond, Hamid Asgari, Daniel Jeffery, and Mike Newman. "An Integrated Process for Verifying Deep Learning Classifiers Using Dataset Dissimilarity Measures". In: *International Journal of Artificial Intelligence and Machine Learning* 11.2 (July 2021), pp. 1–21. DOI: 10.4018/ijaiml.289536 (cit. on p. 19).

Bibliography iii

[Lee+18] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *NeurIPS*. 2018 (cit. on p. 19).

[LLS18] Shiyu Liang, Yixuan Li, and R. Srikant. "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=H1VGkIxRZ (cit. on p. 19).

Bibliography iv

[SO20] Chandramouli Shama Sastry and Sageev Oore. “Detecting out-of-distribution examples with gram matrices”. In: *ICML*. 2020 (cit. on p. 19).

[Sun+22] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. “Out-of-distribution Detection with Deep Nearest Neighbors”. In: *International Conference on Machine Learning*. 2022 (cit. on p. 19).

Bibliography v

[Xu-+22] Romain Xu-Darme, Georges Quénot, Zakaria Chihani, and Marie-Christine Rousset. "PARTICUL: Part Identification with Confidence measure using Unsupervised Learning". Accepted at XAIE: 2nd Workshop on Explainable and Ethical AI – ICPR 2022. June 2022. URL: https://hal-cea.archives-ouvertes.fr/cea-03703962 (cit. on p. 12).

Bibliography vi

[Yan+22] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, WENXUAN PENG, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. "OpenOOD: Benchmarking Generalized Out-of-Distribution Detection". In: *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2022 (cit. on pp. 18, 19).

Cracking the CODE open - recipe

Locality:

$$\mathcal{L}_l = -\sum_{(x,y)\in X_{train}} \sum_{c=1}^{N} \sum_{i=1}^{p} \mathbb{1}_{[c=y]} \times \max\left(P_i^{(c)}(x) * u\right) \tag{1}$$

Unicity:

$$\mathcal{L}_u = \sum_{(x,y)\in X_{train}} \sum_{c=1}^{N} \mathbb{1}_{[c=y]} \times \max\left(0, \max\left(S^{(c)}(x)\right) - t\right) \tag{2}$$