EC6.21

Methodological Guideline for OOD detection

L3.5.2.3

contact@confiance-ai.fr | www.confiance.ai

Document reference: 321GA

Contributors

	Name	Organisation	Role
Responsible for the deliverable	Daniel MONTOYA, Fabio ARNEZ	CEA	Responsible for deliverable
Scientific responsible			
Co-authors	Fabio ARNEZ	CEA	Author

Document Control

Revision	Date	Commentary	Author
v1.0	4/12/2023		Daniel MONTOYA, Fabio ARNEZ, Ansgar RADERMACHER

Contents

A. Introduction and abstract

A.1. General introduction to trustworthy AI challenges

Trustworthiness in AI within critical systems (systems that can directly or indirectly affect human life and moral entities) is essential for its widespread adoption (by the industry, the decision makers, the general public, etc.) and poses the following significant challenges.

- First, how to design AI models in a way that assures the satisfaction of trustworthy properties (accuracy, robustness...) by construction.
- Secondly, how to characterize these AI models, for example to understand and explain their behavior and their adequacy to the operational domain.
- Then, how to implement and embed those AI models on hardware, by making them fit for the target without losing their trustworthy properties.
- Another question is, what data engineering methods to apply in order to manage important volumes of data and adapt to the evolution of the operational domain.
- At system level, what verification and certification processes to consider specifically for AI-based systems.
- Finally, a federation of all these matters is necessary to build an end-to-end methodological approach, supported by a consistent engineering environment compatible with industrial practices.

These are the challenges, among others, that the Confiance.ai program addresses.

In particular for this text, it is complementary with the Benchmarking and use-case level documents, where the results are shown and discussed. The introductions for all three texts are identical.

A.2. Introduction

As highly automated systems increasingly rely on DNNs to perform safety-critical tasks, confidence representation in DNN predictions has become crucial when deployed in the open world. Trustworthy DNN models should provide accurate predictions and detect samples that differ from those observed in the training distribution. Therefore, capturing information about *"what the model does not know"* is not only helpful but essential in safety-critical tasks and real-world deployment Sun et al. (2021).

In image classification, multiple methods have been proposed for distribution shift detection by building DNN prediction confidence scores, among which post-hoc methods stand out mainly by their less-invasive nature and practical use Yang et al. (2021); Ruff et al. (2021). DNN predictive uncertainty offers a plain confidence representation. Existing Bayesian Deep Learning (BDL) methods offer a simple and principled approach to estimating DNN uncertainty. DNN predictive uncertainty with BDL methods has been used for detecting Out-of-Distribution (OoD) samples under the assumption that samples far away from the training distribution provide higher predictive uncertainty than samples observed in the training data Ovadia et al. (2019); Kendall and Gal (2017).

While Bayesian Deep Learning (BDL) sampling-based methods are conceptually straightfor-

ward (*e.g.*, Monte-Carlo dropout), their practical implementation is hindered by substantial computational costs, limiting widespread adoption. Furthermore, recent research works Yang et al. (2021); Kirsch et al. (2021) argue that BDL uncertainty is comparatively less effective for OoD detection when contrasted with more direct (deterministic) post-hoc methods.
In addition, these problems can scale up to more complex computer vision tasks. In semantic segmentation, the lack of information on semantic structures and contexts yields miss-matches between anomaly pixel masses and pixel uncertainty regions Di Biase et al. (2021); Xia et al. (2020). In object detection, object distance and occlusion can impact the bounding-box predictive uncertainty for regression and classification Feng et al. (2021); Wang et al. (2020). Therefore, the above-mentioned limitations lead to the open question: *Is DNN uncertainty estimation with simple sample-based methods still a competitive confidence measure for distribution shift detection?*
In this work, we propose to use the uncertainty from intermediate latent representations (feature maps and embeddings) by using *Dropout* at inference to detect distribution shifts at the image level. We leverage the latent representation entropy density from the training dataset and propose the `LaRED` & `LaREM` scores. `LaRED` uses the entropy density log-likelihood value as a score function, while `LaREM` employs the Mahalanobis distance to the entropy density as a score function. Our approach offers compelling benefits: 1) OoD data agnostic, *i.e.*, the score threshold is estimated only with InD data; 2) simple post-hoc method that requires a single DropBlock (or Dropout) layer; 3) reduced runtime compared to sample-based BDL techniques and comparable to deterministic counterpart methods; 4) the presented scores can be applied to different CNN-based model architectures from different tasks *i.e.*, CNN model task agnostic score. The contributions for both this document and the benchmarking one are summarized below:

1. We present the `LaRED` & `LaREM` scores (`LaREx` for short to refer to both scores) for image-level distribution shift detection, exploring and demonstrating the efficacy of our approach by combining the benefits of simple sample-based methods for uncertainty estimation with density & distance-based methods for OoD detection.
2. We demonstrate the applicability of `LaRED` & `LaREM` beyond image classification with more complex computer vision tasks. We performed experiments in semantic segmentation and object detection tasks with the different corresponding DNN architectures and by adapting common post-hoc methods to perform image-level detection.
3. We present perspectives on enhancing the practical effectiveness of `LaRED` & `LaREM`, encompassing aspects such as regularization, dimensionality reduction, and the layer at which we take samples. Ablation studies substantiate our conclusions.

A.3. Related Work

In distribution shift detection, post-hoc methods aim to create confidence scores that have a minimal impact on the DNN architecture and the training process without altering the loss function. Post-hoc methods are presented below.
Output-based Methods. These methods aim at devising confidence scores based on the DNN outputs. Hendrycks and Gimpel (2016) proposed the first simple baseline method that uses the maximum softmax probability (MSP) as an InD membership score. Later work suggests using the maximum logit to outperform MSP Hendrycks et al. (2019). More recently, Liu et al. (2020) proposed the energy score by summing up the prediction logits over all classes. In this line of work, ASH Djurisic et al. (2022), DICE Sun and Li (2022), and ReAct Sun et al. (2021) have worked on improving the energy score separability for InD and OoD data by modifying

the activations of the penultimate layer and applying thresholding and scaling, sparsification, or clipping. In the context of uncertainty estimation, sample-based approximate Bayesian inference methods Gal and Ghahramani (2016); Lakshminarayanan et al. (2017) are used to generate multiple predictions for the same input sample, from which the predictive entropy and mutual information can be used as confidence scores Kirsch et al. (2021); Mukhoti et al. (2023). Unlike the previous methods, we do not use the DNN outputs for our confidence score and instead use the uncertainty from an intermediate latent representation.

Density-based Methods. Density-based methods aim at modeling the InD density with probabilistic models. Naturally, a line of work in the literature employs generative models to represent the training data distribution. The rationale of the approach is that high-likelihood values will be assigned to InD samples, while low-likelihood values are assigned to OoD samples. However, Nalisnick et al. (2018) and Choi et al. (2018) showed that this assumption does not hold since the typical set of the data may not intersect with the high-likelihood region. In particular, the latter work argues that OoD data is assigned higher likelihoods due to epistemic errors and proposes using an ensemble of density models. Similarly, in the context of discriminative models, deterministic uncertainty estimation methods Postels et al. (2020); Blum et al. (2021); Mukhoti et al. (2023) aim to estimate the embedding density while connecting to the traditional BDL approach. Unlike previous works, our approach estimates and uses the entropy density from intermediate representations.

Distance-based Methods. These methods assume that OoD samples reside in farther locations than InD samples from the training reference examples. Lee et al. (2018) proposed using the minimum Mahalanobis distance to all embedding centroids per class, assuming that the feature space centroids follow a multivariate normal distribution. Recent work from Sun et al. (2022) shows promising results by following a non-parametric approach and not imposing distribution assumptions in the feature space. Other works Techapanurak et al. (2020); Nitsch et al. (2021) use the cosine similarity between class embeddings and test sample embeddings as a confidence score. In the case of `LaRED` & `LaREM`, the former follows the non-parametric approach for density estimation. The latter assumes a parametric density whose parameters are used to compute the Mahalanobis distance.

B. Description of the method

B.1. Problem Formulation

Data distribution shift detection can be framed as a binary classification task. The classifier Ω aims at using a confidence score $\mathscr{S}$ with a corresponding threshold τ to determine (at inference time) whether a new input sample $\mathbf{x}^*$ belongs to the training data distribution $\mathbb{P}^+$ or not (OoD, anomalous samples), as presented in eq. (B.1):

$$\Omega\Big(\mathscr{S}(\boldsymbol{x}^*), \tau\Big) \begin{cases} 1 & InD \quad \mathscr{S}(\boldsymbol{x}^*) \geq \tau \\ 0 & OoD \quad \mathscr{S}(\boldsymbol{x}^*) < \tau \end{cases} \tag{B.1}$$

Therefore, following the equation above, the goal is to derive a confidence score such that–*by convention in the literature*–positive InD samples have higher confidence scores and vice versa for OoD or anomalous input samples. Then, the classifier Ω uses the confidence score $\mathscr{S}$ to get a notion of trust in the DNN and elicit its verdict.

B.2. Method

We propose an uncertainty-based score that leverages the entropy from an intermediate DNN latent representation given InD samples to enable the detection of newly shifted samples (OoD, anomalous samples). Taking inspiration from Morningstar et al. (2021), in our formulation, the DNN latent representation entropy is represented as a random variable $\Psi \sim f_\Psi(\psi)$. The following sections describe our approach to capture latent representation entropy, the InD entropy density $f_\Psi(\psi)$ estimation, and the score computation using $\hat{f}_\Psi(\psi)$.

B.2.1 Latent Representations Uncertainty

Key to our approach is the estimation of uncertainty from a DNN latent representation. A simple way to estimate uncertainty is by applying dropout Srivastava et al. (2014) to a latent representation $\tilde{\boldsymbol{z}}$ from a trained DNN, *i.e.*, adding multiplicative noise to $\tilde{\boldsymbol{z}}$, as presented below in eq. (B.2):

$$\begin{aligned} \mathbf{m} &\sim \mathscr{B}(p_m) \\ \boldsymbol{z} &= \mathbf{m} \odot \tilde{\boldsymbol{z}} \end{aligned} \tag{B.2}$$

Where $\mathbf{m}$ is the vector of independent *Bernoulli* random variables—*the dropout mask*—where p_m is the drop probability and has the same dimension as $\tilde{\boldsymbol{z}}$. A vector $\boldsymbol{m}$ is sampled and multiplied element-wise with the latent code $\tilde{\boldsymbol{z}}$ to produce a modified *"noisy"* latent code $\boldsymbol{z}$, for which we would like to marginalize out the dropout mask noise as follows:

$$\begin{aligned} p_\theta(\mathbf{z} \mid \boldsymbol{x}) &= \int p_\theta(\mathbf{z} \mid \boldsymbol{x}, \boldsymbol{m}) \underbrace{p(\boldsymbol{m})\, d\boldsymbol{m}}_{\textit{dropout masks}} \\ p_\theta(\mathbf{z} \mid \boldsymbol{x}) &= \mathbb{E}_{p(\boldsymbol{m})}\big[p_\theta(\mathbf{z} \mid \boldsymbol{x}, \boldsymbol{m})\big] \end{aligned} \tag{B.3}$$

Thus, to get the uncertainty of the latent code $\mathbf{z}$, we take multiple samples from $\mathbf{m}$ to generate multiple dropout masks so that we can produce a set of M samples $\mathbf{z}$, $\{\mathbf{z}_i\}_{i=1}^M$ that approximate

eq. (B.3). This set of samples, produced with a DNN with weights θ and input $\boldsymbol{x}$, help us characterize the sampling distribution $p_\theta(\mathbf{z} \mid \boldsymbol{x})$, whose entropy is presented in eq. (B.4).

$$\mathbb{H}(\mathbf{z} \mid \boldsymbol{x}) = -\int p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) \ln p_\theta(\boldsymbol{z} \mid \boldsymbol{x}) d\boldsymbol{z} \tag{B.4}$$

From a practical point of view, we need a single dropout layer to get the samples $\{\boldsymbol{z}_i\}_{i=1}^M$ to approximate the integral from eq. (B.3). In addition, during deployment, this situation allows us to speed up the sampling acquisition since we no longer need to pass an input sample throughout the whole DNN. We perform a single forward pass for a given input sample and capture the latent representation just before the target dropout layer. Then, we apply different dropout masks to the captured latent representation.

Naturally, our approach to capture the latent representation uncertainty is akin to the Monte-Carlo dropout (MCD) Gal and Ghahramani (2016) method for Bayesian approximation. However, our method differs since we apply dropout to produce multiple noisy versions of the representation. Thus, to distinguish with MCD, we use the term *z Monte-Carlo Dropout (zMCD)* henceforth.

zMCD on feature maps. Standard dropout is ineffective when applied to convolutional neural networks (CNNs) since it does not remove semantic and spatial information from CNN feature maps. On the other hand, dropping continuous regions in 2D feature maps with *DropBlock* can help remove semantic information and enforce remaining units to learn features for the assigned task Ghiasi et al. (2018). This effect is also desired for capturing uncertainties to overcome the standard dropout limitation. Therefore, we follow the approach from Deepshikha et al. (2021) and use DropBlock to capture the uncertainty from feature maps.

Feature map dimensionality reduction. CNN feature maps are of the form $\boldsymbol{z} \in \mathbb{R}^{C \times H \times W}$, where C, H and W denote the feature map number of channels, height, and width respectively. We compute the mean of the feature map across the spatial dimensions (H and W) so that the latent feature representation is reduced to a vector:

$$\boldsymbol{z}_{\mu_c} = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \boldsymbol{z}(c,h,w), \text{ where } \boldsymbol{z}_{\mu_c} \in \mathbb{R}^C \tag{B.5}$$

B.2.2 Representation Entropy Density for Detecting Distribution Shift

To start the entropy computation for detecting shifted samples, we first assume access to a training dataset $\mathscr{D}_t = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_n^N$ with N samples. Now, we generate a set of zMCD samples $\{\boldsymbol{z}_i\}_{i=1}^M$ for each training sample $\boldsymbol{x}_n$ The resulting zMCD samples can then be used to approximate the entropy from eq. (B.4), *e.g.*, using standard entropy estimators such as nearest-neighbor methods Kozachenko and Leonenko (1987):

$$\hat{\mathbb{H}}_n\left(\{\boldsymbol{z}_i\}_{i=1}^M\right) \approx \mathbb{H}_n(\mathbf{z} \mid \boldsymbol{x}_n) \tag{B.6}$$

Consequently, we produce entropy estimation vector samples $\{\boldsymbol{\psi}_n\}_n^N$ for the training dataset $\mathscr{D}_t$ (InD) samples:

$$\begin{aligned} \boldsymbol{\psi} &= \hat{\mathbb{H}}(\mathbf{z} \mid \boldsymbol{x}) \\ \{\boldsymbol{\psi}_n\}_n^N &= \hat{\mathbb{H}}_n(\mathbf{z} \mid \boldsymbol{x}_n), \forall \boldsymbol{x}_n \in \mathscr{D}_t \end{aligned} \tag{B.7}$$

The entropy estimation samples $\{\boldsymbol{\psi}_n\}_n^N$ from $\mathscr{D}_t$ are used to estimate the InD entropy density function $f_\Psi \approx \hat{f}_\Psi$. In the case of `LaRED`, f_Ψ is estimated using Kernel Density Estimation (KDE):

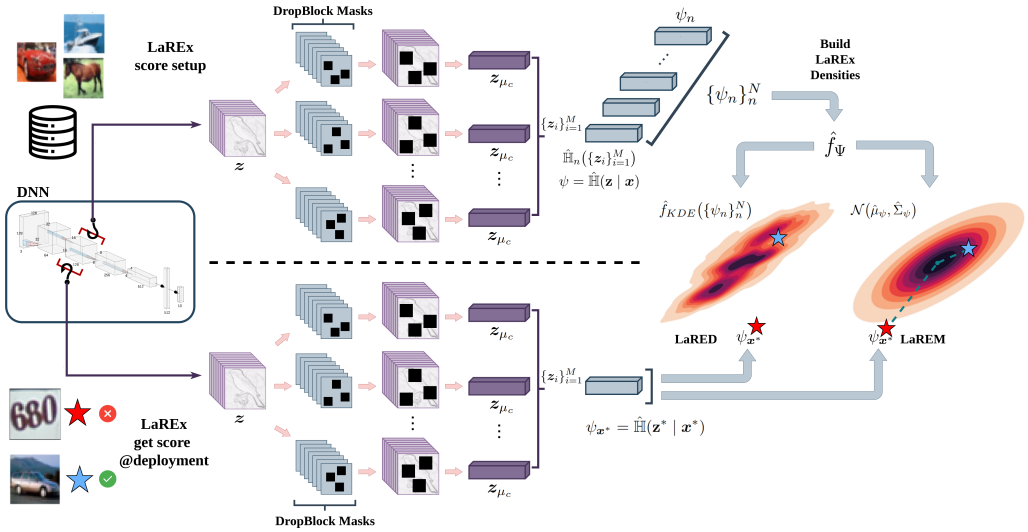

Figure B.1: LaRED & LaREM confidence score overview. The upper part of the figure depicts the score setup computation to get the entropy density estimates $\hat{f}_\Psi$. The lower part of the figure shows the score computation during deployment.

$$\hat{f}_\Psi = \hat{f}_{KDE}\left(\{\psi_n\}_n^N\right) \tag{B.8}$$

In the case of LaREM, we assume that f_Ψ is a multivariate Normal distribution, parameterized by the estimated mean $\hat{\mu}_\psi$ and covariance $\hat{\Sigma}_\psi$ from $\{\psi_n\}_n^N$, as presented below:

$$\hat{f}_\Psi = \mathcal{N}\left(\hat{\mu}_\psi, \hat{\Sigma}_\psi\right) \tag{B.9}$$

At test or deployment time, we use the estimated InD entropy density $\hat{f}_\Psi$ to produce a confidence score for a new input sample $\boldsymbol{x}^*$. To this end, we produce a set of zMCD samples $\{\boldsymbol{z}_i^*\}_{i=1}^M$ to estimate the latent code $\mathbf{z}^*$ entropy for a new input sample $\boldsymbol{x}^*$:

$$\psi_{\boldsymbol{x}^*} = \hat{\mathbb{H}}(\mathbf{z}^* \mid \boldsymbol{x}^*) \tag{B.10}$$

In the case of LaRED—*Latent Representation Entropy Density log-likelihood*—score, we compute the score using the log-likelihood of the entropy estimation sample $\psi_{\boldsymbol{x}^*}$ for a new input sample $\boldsymbol{x}^*$, using the estimated entropy density function from eq. (B.8):

$$\texttt{LaRED}(\boldsymbol{x}^*) = \log \hat{f}_{KDE}\left(\psi_{\boldsymbol{x}^*}\right) \tag{B.11}$$

For the LaREM—*Latent Representation Entropy density Mahalanobis distance*—score, we compute the negative Mahalanobis distance, using the estimated density $\hat{f}_\Psi$ parameters from eq. (B.9) and the entropy estimation vector $\psi_{\boldsymbol{x}^*}$ for a new input sample $\boldsymbol{x}^*$:

$$\texttt{LaREM}(\boldsymbol{x}^*) = -\left(\left(\psi_{\boldsymbol{x}^*} - \hat{\mu}_\psi\right)^\top \hat{\Sigma}_\psi^{-1}\left(\psi_{\boldsymbol{x}^*} - \hat{\mu}_\psi\right)\right) \tag{B.12}$$

The expression in eq. (B.12) is based on the score from Lee et al. (2018). However, we do not perform per-class centroid distance computations. Moreover, the LaREM score uses negative distance values to align with the convention where InD samples have higher confidence score values.

Entropy vector dimensionality reduction. Following previous works Lee et al. (2018); Postels et al. (2020); Yang et al. (2023), we apply principal components analysis (PCA) to reduce the dimensionality of the obtained entropy vectors ψ_{x^*}. Entropy vectors have the same dimensions as the latent code z or z_{μ_c}. Thus, the goal is to reduce the dimensions from C to C' so that $\psi_{x^*} \in \mathbb{R}^{C'}$, where $C' < C$. Applying PCA is particularly important for the `LaRED` score, given the common limitations of the KDE algorithm in high-dimensional spaces.
Figure fig. B.1 presents an overview of `LaRED` & `LaREM` confidence score setup and computation.

B.3. `LaREM & LaRED` Algorithm

The computation details for `LaRED & LaREM` are available in Algorithm 1.

Algorithm 1 Latent Representation Entropy Density-based Distribution Shift Detection: `LaRED` & `LaREM` Confidence Scores.

Definitions:
- Trained DNN $p_\theta(\mathbf{y} \mid \boldsymbol{x})$ with Dropout or DropBlock layer
- Feature extractor $p_\theta(\mathbf{z} \mid \boldsymbol{x})$ (Hook on Dropout or DropBlock layer)
- Training dataset samples $\mathscr{D}_t = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_n^N$

procedure: `setup_LaREx_score`:

for each $\boldsymbol{x}_n \in \mathscr{D}_t$ **do**
get M zMCD samples $\{z_i\}_{i=1}^M \sim p_\theta(\mathbf{z} \mid \boldsymbol{x}_n)$
$\psi_n \leftarrow$ entropy $\left(\{z_i\}_{i=1}^M\right)$
save ψ_n sample into Ψ
end for

$\Psi = \{\psi_n\}_n^N$

if `LaRED` **then**
$\hat{f}_\Psi = \hat{f}_{KDE}(\Psi)$
end if
if `LaREM` **then**
$\hat{\mu}_\Psi \leftarrow mean(\Psi)$; $\hat{\Sigma}_\Psi \leftarrow covariance(\Psi)$
$\hat{f}_\Psi = \mathscr{N}(\hat{\mu}_\Psi, \hat{\Sigma}_\Psi)$
end if
end procedure

function: `get_LaREx_score`(*new sample* $\boldsymbol{x}^*$):
get M zMCD samples $\{z_i\}_{i=1}^M \sim p_\theta(\mathbf{z} \mid \boldsymbol{x}^*)$
$\psi_{x^*} \leftarrow$ entropy $\left(\{z_i\}_{i=1}^M\right)$
if `LaRED` **then**
$\mathscr{S} = \log \hat{f}_{KDE}(\psi_{\boldsymbol{x}^*})$
end if
if `LaREM` **then**
$\mathscr{S} = -\left((\psi_{\boldsymbol{x}^*} - \hat{\mu}_\Psi)^\top \hat{\Sigma}_\Psi^{-1} (\psi_{\boldsymbol{x}^*} - \hat{\mu}_\Psi)\right)$
end if
Return $\mathscr{S}$
end function

Finally, for the results of this method, refer to the bench-marking document.

Bibliography

Blum, H., Sarlin, P.-E., Nieto, J., Siegwart, R., and Cadena, C. (2021). The fishyscapes benchmark: measuring blind spots in semantic segmentation. *International Journal of Computer Vision*, 129(11):3119–3135.

Choi, H., Jang, E., and Alemi, A. A. (2018). Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*.

Deepshikha, K., Yelleni, S. H., Srijith, P., and Mohan, C. K. (2021). Monte carlo dropblock for modelling uncertainty in object detection. *arXiv preprint arXiv:2108.03614*.

Di Biase, G., Blum, H., Siegwart, R., and Cadena, C. (2021). Pixel-wise anomaly detection in complex driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16918–16927.

Djurisic, A., Bozanic, N., Ashok, A., and Liu, R. (2022). Extremely simple activation shaping for out-of-distribution detection. *arXiv preprint arXiv:2209.09858*.

Feng, D., Harakeh, A., Waslander, S. L., and Dietmayer, K. (2021). A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):9961–9980.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Ghiasi, G., Lin, T.-Y., and Le, Q. V. (2018). Dropblock: A regularization method for convolutional networks. *Advances in Neural Information Processing Systems*, 31:10727–10737.

Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., and Song, D. (2019). Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*.

Hendrycks, D. and Gimpel, K. (2016). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584.

Kirsch, A., Mukhoti, J., van Amersfoort, J., Torr, P. H. S., and Gal, Y. (2021). On pitfalls in ood detection: Entropy considered harmful. Uncertainty & Robustness in Deep Learning Workshop, ICML.

Kozachenko, L. and Leonenko, N. N. (1987). Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in neural information processing systems*, pages 6402–6413.

Lee, K., Lee, K., Lee, H., and Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Liu, W., Wang, X., Owens, J., and Li, Y. (2020). Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

Morningstar, W., Ham, C., Gallagher, A., Lakshminarayanan, B., Alemi, A., and Dillon, J. (2021). Density of states estimation for out of distribution detection. In *International Conference on Artificial Intelligence and Statistics*, pages 3232–3240. PMLR.

Mukhoti, J., Kirsch, A., van Amersfoort, J., Torr, P. H., and Gal, Y. (2023). Deep deterministic uncertainty: A new simple baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24384–24394.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. (2018). Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*.

Nitsch, J., Itkina, M., Senanayake, R., Nieto, J., Schmidt, M., Siegwart, R., Kochenderfer, M. J., and Cadena, C. (2021). Out-of-distribution detection for automotive perception. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2938–2943. IEEE.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, 32:13991–14002.

Postels, J., Blum, H., Strümpler, Y., Cadena, C., Siegwart, R., Van Gool, L., and Tombari, F. (2020). The hidden uncertainty in a neural networks activations. *arXiv preprint arXiv:2012.03082*.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. (2021). A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Sun, Y., Guo, C., and Li, Y. (2021). React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157.

Sun, Y. and Li, Y. (2022). Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, pages 691–708. Springer.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. (2022). Out-of-distribution detection with deep nearest neighbors. *arXiv preprint arXiv:2204.06507*.

Techapanurak, E., Suganuma, M., and Okatani, T. (2020). Hyperparameter-free out-of-distribution detection using cosine similarity. In *Proceedings of the Asian conference on computer vision*.

Wang, A., Sun, Y., Kortylewski, A., and Yuille, A. L. (2020). Robust object detection under occlusion with context-aware compositionalnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12645–12654.

Xia, Y., Zhang, Y., Liu, F., Shen, W., and Yuille, A. L. (2020). Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In *European Conference on Computer Vision*, pages 145–161. Springer.

Yang, J., Zhou, K., Li, Y., and Liu, Z. (2021). Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*.

Yang, J., Zhou, K., and Liu, Z. (2023). Full-spectrum out-of-distribution detection. *International Journal of Computer Vision*, pages 1–16.

Confiance ai

Title: Latent Representation Entropy Density for Distribution Shift Detection

Keywords: OoD Detection, Object detection, image segmentation

Distribution shift detection is paramount in safety-critical tasks that rely on Deep Neural Networks (DNNs). The detection task entails deriving a confidence score to assert whether a new input sample aligns with the training data distribution of the DNN model. While DNN predictive uncertainty offers an intuitive measure of confidence, the exploration of uncertainty-based distribution shift detection with simple sample-based techniques has been relatively overlooked in recent years due to computational overhead and lower performance when compared to plain post-hoc methods. In this paper, we propose using simple sample-based techniques for uncertainty estimation and employing the entropy density from intermediate representations for detecting distribution shifts. We demonstrate the effectiveness of our method using common benchmark datasets for Out-of-Distribution detection and across different common perception tasks with CNN-based architectures. Notably, our scope extends beyond classification, encompassing image-level distribution shift detection within DNNs for object detection and semantic segmentation tasks. Our results show that our method's performance is comparable to existing state-of-the-art baseline methods, affirming its practical utility.


Air Liquide
AIRBUS
Atos
cea
Inria
NAVAL GROUP
GROUPE RENAULT
SAINT EXUPÉRY
SAFRAN
sopra steria
SystemX
THALES
Building a future we can all trust
Valeo
INVESTIR L'AVENIR