INTERPRETABLE OUT-OF-DISTRIBUTION DETECTION USING PATTERN DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Out-of-distribution (OoD) detection for data-based programs is a goal of paramount importance. Common approaches in the literature tend to train binary classifiers requiring inside-of-distribution (IoD) and OoD validation samples, and/or implement confidence metrics that are often abstract and therefore difficult to interpret. In this work, we propose to use the PARTICUL pattern detection algorithm in order to build more interpretable and robust OoD detectors for visual classifiers. Crucially, this approach does not require to retrain the classifier and is tuned directly to the IoD dataset, making it applicable to domains where OoD does not have a clear definition. Moreover, pattern detection allows us to provide images from the IoD dataset as reference points to better explain our confidence scores. We illustrate the generalization abilities of our approach through an extensive benchmark across four datasets and two definitions of OoD. Our experiments show that the robustness of all metrics under test does not solely depend on the nature of the IoD dataset or the OoD definition, but also on the architecture of the classifier, which stresses the need for thorough experimentations for future work on OoD detection.

1 INTRODUCTION

A fundamental aspect of software safety is arguably the modelling of its expected operational domain through a formal or semi-formal specification, giving clear boundaries on when it is sensible to deploy the program, and when it is not. It is however difficult to define such boundaries for machine learning programs, especially for visual classifiers based on artificial neural networks (ANN), which are the subject of this paper. Indeed, such programs process high-dimensional data (images, videos) and are the result of a complex optimization procedure, but they do not embed clear failure modes that could get trigged in the case of an unknown distribution, with potentially dire consequences in critical applications.
Although it is difficult to characterize the operational distribution, one could still measure its dissimilarity with other distributions. In this context, Out-of-Distribution (OoD) detection - which aims to detect whether an input of an ANN is Inside-of-distribution (IoD) or outside of it - serves several purposes. It helps characterize the extent to which the ANN can operate outside a bounded dataset (which is important due to the incompleteness of the training set *w.r.t.* the operational domain). It also constitutes a surrogate measure of the generalization abilities of the ANN. Finally, OoD detection can help assess when an input is too far away from the operational domain, which prevents misuses of the program and increases its safety.

2 CONTRIBUTION AND RELATED WORK

Out-of-distribution detection The *maximum class probability* (MCP) obtained after softmax normalization of the classifier logits already constitutes a good baseline for OoD detection Hendrycks & Gimpel (2017). However, neural networks tend to be overconfident on their predictions Szegedy et al. (2014); Lee et al.

(2018a); Hein et al. (2019), even when they are wrong, which may result in false claims of confidence. Hence, the development of enhancements like temperature scaling Liang et al. (2017), ensemble learning Nguyen et al. (2020) or True-Class Probability learning Corbiere et al. (2021) (assuming such information is known). As an alternative, other types of confidence measures have also been developed. Approaches such as the Fractional Neuron Region Distance Hond et al. (2021) (FNRD) or the Attribution-Based Confidence Jha et al. (2019) (ABC) require only access to a trained network and, for FNRD, to the training set of the classifier. FNRD computes the range of activations for each neuron on the training set, then provides a score describing how many neurons are outside their boundaries for a given input, while ABC equates the confidence of the network to its local stability by sampling the neighbourhood of a given input and measuring stability through attribution methods. Although this last method relies on fewer prerequisites, it is computationally expensive (*e.g.,* gradient computation) and may not be suited to runtime constraints.
Other approaches also attempt to capture the "normal" behaviour of the classifier: Huang et al. (2021) states that the latent representations of OoD samples through a CNN classifier are clustered around a point called the *feature-space singularity* (FSS) which serves as a reference point for OoD detection. Similarly, Lee et al. (2018b) proposes a confidence score based on the Mahalanobis distance between a new sample and class conditional Gaussian distributions inferred from the training set. However, both approaches operate upon multiple layers of the network and require the use of a set of OoD samples for calibrating the relative importance of each layer in the final confidence score. As we will see later in this work, this reliance on additional samples can lead to very unstable results depending on their representativeness of OoD inputs.
A final line of research proposes to integrate the learning of a confidence measure directly during the training of the classifier itself (Lee et al. (2018a); Hein et al. (2019)). Similarly, the whole field of Bayesian Deep-Learning aims to capture uncertainty by design in the model, using for instance dropout as an approximation for Bayesian Inference Gal & Ghahramani (2016). While efficient, these approaches require to retrain the classifier, which can prove costly or impractical in an industrial context.
Importantly, all these methods were evaluated on different datasets and definitions of OoD (*e.g.,* different datasets, distribution shifts), which only gives a partial picture on their robustness Tajwar et al. (2021).

Pattern detection A lot of work has been focused on part detection, in supervised (using annotations Farhadi et al. (2009); Han et al. (2018); Zhao et al. (2019)), weakly-supervised (using class labels Li et al. (2020); Peng et al. (2018)) or unsupervised settings (Zheng et al. (2017); Han et al. (2022); Xu-Darme et al. (2022)), primarily with the goal of improving accuracy on hard classification tasks such as fine-grained recognition Welinder et al. (2010); Yang et al. (2015). To our knowledge, the PARTICUL algorithm described in Xu-Darme et al. (2022) is the only method including a confidence measure associated with the detected parts (in the paper, it is used to infer the visibility of a given part). PARTICUL aims at mining recurring patterns in the latent representation of a set of images processed through a CNN, in an unsupervised manner. It is however restricted to fine-grained recognition datasets where all images belong to the same macro-category.

Our contributions In this paper, we aim to investigate the following research tracks: 1) How well can OoD detection methods generalize across multiple definitions of OoD? 2) How to enhance the overall performance (precision, recall, genericity) of OoD detection? 3) How to improve the interpretability of confidence measures in the context of visual classifiers? As an answer to those questions, we make the following contributions:

1. We present a novel OoD measure relying on part detection learning based on work (Xu-Darme et al. (2022)) in explainable AI (xAI) that can use any definition of in-distribution dataset, does not depend on a definition of OoD, can provide visual justifications for confidence scores, is modular for the problem at hand and requires no retraining of the original classifier, and, in fact, needs to access only one layer of the model which allows it to be used in an almost black-box setting;

2. We demonstrate the ability of our approach to consistently capture various notions of OoD over an extensive benchmark comprising four datasets, two architectures and several evaluation metrics.

This paper is organized as follows: Sec. 3 formulates the problem and our OoD definition; Sec. 4 introduces our modifications to the PARTICUL algorithm with the goal of providing a more interpretable and robust confidence measure; Sec. 5 describes our experimental setup and results. Finally, Sec. 6 presents our conclusions and future work.

3 Problem formulation

Let $\mathcal{I}_s$ be the space of all RGB images of size s. Let M be a deep convolutional neural network (CNN) classifier trained on dataset $D_{iod} \subseteq \mathcal{I}_s$, approximating the distribution $\mathcal{D}_{iod}$ of images belonging to N categories. For any image $x \in \mathcal{I}_s$, M outputs a vector of logits $M(x) \in \mathbb{R}^N$, where the index of the highest value corresponds to the most probable category (or *class*) of x - relatively to all other categories. Since any image is assigned a category regardless of its similarity to the training distribution, $M(x)$ is often normalized using a softmax function in order to indicate a level of confidence in the prediction. More generally, a confidence measure $C_M : \mathcal{I}_s \to [0, 1]$ is a function assigning a score to each image $x \in \mathcal{I}_s$ processed through M. We denote $C_M(\mathcal{D}_{iod})$ the distribution of values computed over $\mathcal{D}_{iod}$ using the confidence measure C_M. As indicated above, $C_M(\mathcal{D}_{iod})$ is approximated by $C_M(\mathcal{D}_{iod}) \approx C_M(D_{iod}) = \big\{C_M(x) \mid x \in D_{iod}\big\}$. Ideally, images belonging to the training distribution $\mathcal{D}_{iod}$ should have a higher confidence score than images outside $\mathcal{D}_{iod}$. However, it is not possible to obtain a dataset representative of $\mathcal{I}_s \setminus \mathcal{D}_{iod}$, *i.e.,* encompassing all possible OoD inputs. Therefore, in this work we use two weaker modalities of OoD detection that serve as a benchmark for various proposals of confidence measures.

Cross-dataset detection In this experiment, we evaluate the separability of the distributions $C_M(D_{iod})$ and $C_M(D_{ood})$, where D_{ood} is a dataset different from D_{iod}. We use three complementary metrics that do not require to determine an acceptance threshold between IoD and OoD inputs, and that are suitable to compare different proposals of confidence measures: the Area Under the ROC curve (AUROC↑); the Area Under the Precision-Recall curve (AUPR↑); the False Positive Rate when the true positive rate is 80% (FPR80↓).

Sensitivity to dataset perturbation For this second experiment, we build the OoD dataset D_{ood} by applying a perturbation to all images of D_{iod}. More precisely, a perturbation P_λ is a function applying a transformation on an image $x \in \mathcal{I}_s$ with a variable intensity λ (*e.g.,* for a rotation, λ might correspond to the angle of the rotation). For a perturbation P_λ applied over D_{iod} , we define the average value of confidence measure C_M as

$$\Gamma(C_M, D_{iod}, P, \lambda) = \frac{1}{|D_{iod}|} \sum_{x \in D_{iod}} C_M(P_\lambda(x)) \tag{1}$$

which is extended to a set of perturbation intensities $\Lambda = (\lambda_0, \ldots, \lambda_n)$ as $\Gamma(C_M, D_{iod}, P, \Lambda) = \big(\Gamma(C_M, D_{iod}, P, \lambda_0), \ldots, \Gamma(C_M, D_{iod}, P, \lambda_n)\big)$. Although it would be again possible to measure the separability of IoD and OoD confidence distributions, perturbations of small intensities would result in almost identical distributions. Instead, we evaluate the correlation between the intensity of the perturbation and the average confidence value of the perturbed dataset by computing the Spearman Rank correlation coefficient r_s between Λ and $\Gamma(C_M, D_{iod}, P, \Lambda)\big)$. Indeed, $r_s\big(\Lambda, \Gamma(C_M, D_{iod}, P, \Lambda)\big) = 1$ (resp. -1) indicates that the average confidence measure increases (resp. decreases) monotonously with the value of λ.

4 Proposed model

In this section, we present two proposals for building a confidence measure based on the PARTICUL algorithm, described in Xu-Darme et al. (2022), which aims at mining recurring patterns in the latent representation of a set of images processed through a CNN. In the context of this paper, we learn patterns from the last convolutional layer of classifier M over the training set D_{iod}, in a plug-in fashion that does not require to

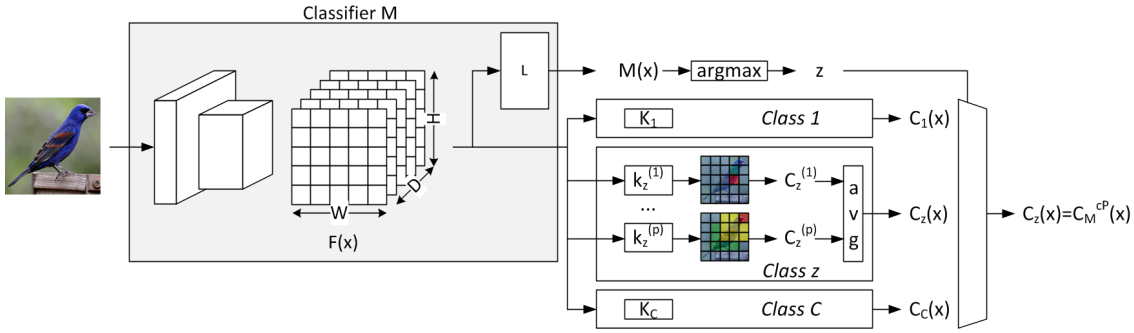


Figure 1: Class-based PARTICUL overview. When processing a new sample, the confidence measure averages the contribution of the detectors corresponding to the predicted class z only.

retrain the classifier. Let F be the restriction of classifier M up to its last convolutional layer, *i.e.,* $M = L \circ F$, where L corresponds to the last pooling layer followed by one or several fully connected layers. $\forall x \in \mathcal{I}_s$, $F(x) \in \mathbb{R}^{H \times W \times D}$ is a convolutional map of D-dimensional vectors. Let $X_{train} \subseteq D_{iod}$ be the subset of the IoD dataset D_{iod} used during the training of the classifier.

***Vanilla* PARTICUL** The purpose of the PARTICUL algorithm is to learn p distinct $1 \times 1 \times D$ convolutional kernels $K = [k^{(1)}, \dots, k^{(p)}]$, such that $\forall x \in X_{train}$, each kernel $k^{(i)}$ strongly correlates with exactly one vector $F_{[h,w]}(x)$ in $F(x)$ (*Locality* constraint). After training, PARTICUL uses the function $H^{(i)}(x) = \max_{h,w} \left(F_{[h,w]}(x) * k^{(i)}\right)$ returning the maximum correlation score between kernel $k^{(i)}$ and the convolutional map $F(x)$. In Xu-Darme et al. (2022), the distribution $H^{(i)}(\mathcal{D}_{iod})$ is modeled as a random variable following a normal distribution $\mathcal{N}(\mu^{(i)}, \sigma^{(i)})$ and estimated over X_{train}. Then, the corresponding confidence measure is computed using the cumulative distribution function (CDF) associated with $\mathcal{N}(\mu^{(i)}, \sigma^{(i)})$. However, such CDF does not have a closed-form solution, which can slow down operations on tensors. Therefore, in this work, we choose to model $H^{(i)}(\mathcal{D}_{iod})$ as a logistic distribution $\mathcal{L}(\mu^{(i)}, \sigma^{(i)})$, where $\mu^{(i)}$ and $\sigma^{(i)}$ are again estimated over X_{train}, and define the confidence associated with detector i for image x as:

$$C^{(i)}(x) = \frac{1}{1 + e^{-\frac{H^{(i)}(x) - \mu^{(i)}}{\sigma^{(i)}}}} \tag{2}$$

Finally, we average the confidence measures of all detectors and define, $\forall x \in \mathcal{I}_s$: $C_M^{vP}(x) = \frac{1}{p} \sum_{i=1}^{p} C^{(i)}(x)$.

PARTICUL is well-suited for datasets with low intra-variance - *i.e.,* when images in X_{train} are very similar - which is the case for fine-grained recognition datasets (Welinder et al. (2010); Yang et al. (2015)). However, for more heterogeneous datasets (Krizhevsky (2009); Li et al. (2022)), it becomes more difficult to find recurring patterns that are present across the entire training set. Moreover, in order to learn more consistent patterns, it might be interesting to learn detectors that are specific to each of the N classes of $\mathcal{D}_{iod}$.

***Class-based* PARTICUL** Here, we introduce a variant of PARTICUL that uses the labels of the training set X_{train} in order to learn p detectors *per class*. During the training and calibration phases, for each pair of images and label $(x, c) \in X_{train}$, only the detectors of class c are modified. More precisely, let $K_c = [k_c^{(1)}, \dots k_c^{(p)}]$ be the set of kernel detectors for class c. As in Xu-Darme et al. (2022), K_c is learned by minimizing the weighted combination of a locality function $\mathcal{L}_l$ and an unicity function $\mathcal{L}_u$,

$\mathcal{L}(K_c) = \mathcal{L}_l(K_c) + \lambda\mathcal{L}_u(K_c)$, restricted to $X_{train,c} = \{x|(x,c) \in X_{train}\}$. Importantly, since patterns may be similar across different classes (*e.g.,* the wheels on a car or a bus), we do not treat images from other classes as negative samples when learning our class-based detectors. Therefore, although this approach may allow learning more specific detectors than the *vanilla*-PARTICUL approach, it effectively restricts the number of training samples per detector to the images of a single class, which is potentially detrimental to the training process. The calibration of the detectors for class c also uses a logistic distribution model, which is estimated over training samples from class c only. Then, as shown in Fig. 1, for any image $x \in \mathcal{I}_s$, and with no access to the ground truth label of x, the confidence measure $C_M^{cP}(x)$ is obtained by averaging the confidence scores from the detectors associated with the predicted class $z = \text{argmax}\big(M(x)\big)$ of image x, *i.e.,*

$$C_M^{cP}(x) = \frac{1}{p}\sum_{i=1}^{p} C_z^{(i)}(x)$$

As a consequence, for an image $x \in \mathcal{D}_{iod}$, the quality of our class-based confidence measure is also dependent on the ability of the classifier to assign the correct class to x, *i.e.,* on the accuracy of M.

5 EXPERIMENTS

In this section, we present the experimental results obtained on our PARTICUL-based confidence measure, in the context of cross-dataset OoD detection and perturbation OoD detection.

Setup We performed our evaluation using a ResNet50 He et al. (2016) classifier trained on various datasets, with different image sizes: the Caltech-101 Li et al. (2022) dataset contains 9,144 images - from 101 different categories - that are resized to 224×224 during training (acc. 86.8%); the CUB-200 Welinder et al. (2010) dataset contains 11,788 images - from 200 different bird species - that are resized to 448×448 during training (acc. 83.6%); the Stanford Cars Yang et al. (2015) dataset contains 16,185 images - from 196 different car models - that are resized to 448×448 (acc. 90.5%). Both CUB-200 and Stanford Cars are fine-grained classification datasets and contain very homogeneous data from a single macro-category, while Caltech-101 contains more heterogeneous data from diverse categories. Each dataset is split between a training set X_{train} (used for calibration) and a test set X_{test} (used for evaluation).
Using the trained classifiers for feature extraction, we implemented our PARTICUL detectors using the two approaches described above: for *vanilla* PARTICUL, we trained 4 or 6 global detectors, ignoring image labels during training; for *class-based* PARTICUL, we trained 4 or 6 detectors *per class*, using the image labels of the training set. As a comparison, we also implemented a confidence measure $C_M^{fnrd}(x) = 1 - fNRD(x)$ based on the *fractional Neuron Region Distance* Hond et al. (2021) (fNRD), as well as the FSSD measure C_M^{fssd} presented in Huang et al. (2021). It is important to indicate that the FSSD is weakly supervised as it uses a *validation* OoD dataset $D_{ood,val}$ in order to learn a linear regressor with both positive and negative samples. In Huang et al. (2021), the authors choose $D_{ood,val} \sim \mathcal{D}_{ood}$, effectively training a binary classifier between $\mathcal{D}_{iod}$ and $\mathcal{D}_{ood}$. For a fair comparison between methods, we perform several experiments using different validation OoD datasets to perform FSSD calibration. Finally, since the output of the classifier can also constitute a valid confidence measure, we also compare our results with the maximum class probability $C_M^{MCP}(x) = \max\big(softmax\big(M(x)\big)\big)$.

Cross-dataset out-of-distribution In this section, we evaluate the ability of various confidence measures to distinguish the dataset D_{iod} - used when training the classifier - from another dataset $D_{ood} \neq D_{iod}$. As shown in Table 1 and illustrated in Fig. 2, the ability to discriminate D_{iod} from D_{ood} widely differs, depending on the choice of measure and the datasets themselves. In particular, when used as reference distributions, homogeneous datasets such as CUB200 and Stanford Cars are in general easier to discriminate than more heterogenous datasets, across all measures except C_M^{fnrd}. This is consistent with the intuition that Caltech101 distribution covers a broader area of the latent space and is therefore harder to circumscribe. As

D_{ood}	Metrics	Weakly supervised FSSD ($D_{ood,val}$) CT	CB	SC	Unsupervised MCP	fNRD	*vanilla* PARTICUL $P=4$	$P=6$	*class-based* PARTICUL $P=4$	$P=6$
D_{iod} = Caltech 101 (heterogeneous)										
CB	AUROC↑	/	(95.0)	28.6	**87.2**	46.8	72.5 ± 4.8	71.7 ± 1.4	72.7 ± 0.8	69.7 ± 1.7
	AUPR↑	/	(88.5)	12.1	**77.5**	17.7	44.0 ± 8.1	47.3 ± 4.6	38.6 ± 2.3	33.9 ± 2.4
	FPR80↓	/	(04.5)	94.5	**22.1**	86.3	49.9 ± 7.6	57.3 ± 1.8	49.5 ± 1.1	54.6 ± 2.2
SC	AUROC↑	/	41.3	(78.0)	**85.3**	70.1	47.3 ± 10.5	41.0 ± 2.4	67.1 ± 0.6	67.0 ± 2.8
	AUPR↑	/	15.3	(49.2)	**71.8**	44.8	19.9 ± 8.6	16.6 ± 3.5	24.0 ± 1.0	24.4 ± 2.3
	FPR80↓	/	96.4	(45.2)	**26.9**	68.1	88.9 ± 6.0	95.5 ± 1.5	58.1 ± 0.8	60.5 ± 4.2
D_{iod} = CUB200 (homogeneous)										
CT	AUROC↑	(94.7)	/	94.5	**96.1**	72.1	91.8 ± 3.3	93.7 ± 0.4	93.6 ± 0.3	93.1 ± 2.0
	AUPR↑	(98.3)	/	98.3	**99.1**	91.5	97.5 ± 0.8	97.7 ± 0.2	98.3 ± 0.1	98.1 ± 0.6
	FPR80↓	(7.0)	/	6.4	**5.1**	52.8	12.9 ± 5.7	9.5 ± 1.3	9.4 ± 1.6	12.8 ± 5.0
SC	AUROC↑	98.6	/	(99.3)	98.8	88.5	97.9 ± 1.8	**99.1** ± 0.1	94.4 ± 3.9	97.5 ± 1.6
	AUPR↑	98.6	/	(98.9)	**98.6**	88.8	97.3 ± 1.9	**98.6** ± 0.1	93.5 ± 4.6	96.0 ± 2.0
	FPR80↓	0.4	/	(0.5)	**0.4**	18.0	1.9 ± 1.5	0.9 ± 0.1	7.9 ± 8.5	4.9 ± 4.6
D_{iod} = Stanford Cars (homogeneous)										
CT	AUROC↑	(99.9)	99.1	/	96.7	45.5	97.4 ± 1.9	**98.4** ± 0.6	86.8 ± 1.6	94.0 ± 3.3
	AUPR↑	(100)	99.8	/	99.4	83.1	99.5 ± 0.4	**99.7** ± 0.1	97.5 ± 0.4	98.9 ± 0.7
	FPR80↓	(0.1)	0.8	/	1.9	79.0	3.9 ± 2.9	**1.8** ± 0.7	21.0 ± 6.8	8.0 ± 5.5
CB	AUROC↑	100	(99.9)	/	97.6	51.8	99.6 ± 0.5	**99.8** ± 0.2	95.4 ± 0.8	97.0 ± 0.9
	AUPR↑	100	(99.9)	/	98.0	56.6	99.7 ± 0.3	99.9 ± 0.1	96.7 ± 0.7	97.9 ± 0.7
	FPR80↓	0	(0)	/	3.7	72.5	0.1 ± 0.1	0.0 ± 0.1	7.2 ± 2.9	2.7 ± 0.9

Table 1: AUROC↑, AUPR↑ and FPR80↓ scores (0-100 scale) for different pairs (D_{iod}, D_{ood}). SC denotes StanfordCars, CB denotes CUB200 and CT denotes Caltech101. The thick vertical line separates FSSD (weakly supervised) from all other unsupervised methods. For FSSD scores, we indicate the validation dataset $D_{ood,val}$ used during the calibration, and scores in parenthesis correspond to the optimal case $D_{ood,val} \sim \mathcal{D}_{ood}$. For unsupervised approaches, we indicate in **bold** the best performing measure for each experiment and each metric.

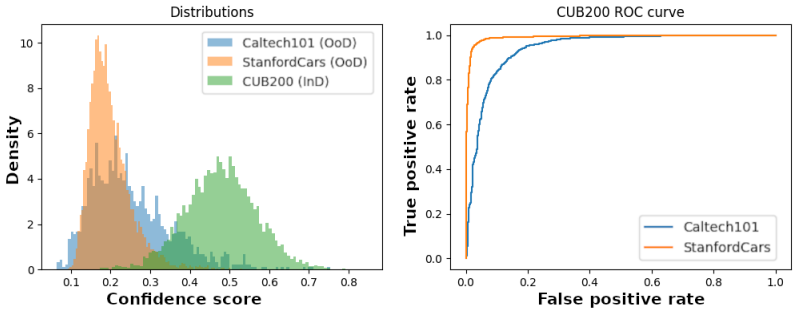


Figure 2: Example of distributions of confidence scores (left) and corresponding ROC curves (right) for a *vanilla* PARTICUL model (6 detectors) trained on CUB200 and tested against Caltech101 and Stanford Cars. Best viewed in color.

expected, when using $D_{ood,val} \sim \mathcal{D}_{ood}$, C_M^{fssd} becomes a binary classifier with fully-supervised training and displays great discriminative properties. However, when using $D_{ood,val} \not\sim \mathcal{D}_{ood}$, the ability of this confidence measure to generalize to unseen datasets can be questioned. Indeed, for Caltech101 v. CUB200 and Caltech101 v. Stanford Cars, we obtain results with a high variance depending on the choice of validation dataset. Surprisingly, the confidence measure C_M^{MCP} shows good discriminative properties for most pairs (D_{iod}, D_{ood}). Without contradicting the conclusions of Hein et al. (2019), this indicates that - outside an adversarial setting - using the normalized output logits of a classifier is often enough to distinguish OoD samples. On the contrary, the measure C_M^{fnrd} is a poor discriminator of OoD datasets. Results of our PARTICUL-based confidence measure are contrasted and also depend on the choice of architecture (*vanilla* or

class-based) *w.r.t.* to the IoD dataset. For CUB200 and Stanford Cars, not only does a *class-based* PARTICUL architecture constitute a good OoD detector (with AUROC scores greater than 90% in 67% of the cases), but *vanilla* PARTICUL fully exploits the homogeneity of the IoD dataset to train robust detectors, with AUROC scores greater than 90% in all cases, using fewer detectors compared to the class-based approach. For Caltech101, *class-based* PARTICUL provides similar (CUB200) or better (Stanford Cars) results than *vanilla* PARTICUL, but at the cost of a greater number of detectors, and with an AUROC score lower than C_M^{MCP}. This is probably due to the fact that Caltech101 already contains images of birds and cars with similar resolution, making the distinction between distributions harder.

This first experiment confirms that our PARTICUL-based confidence measure is, in general, suited to perform OoD detection in a cross-dataset setting. For homogeneous training datasets, *vanilla*-PARTICUL surpasses the other two unsupervised measures and provides results on par with a weakly supervised approach (FSSD). For heterogeneous training datasets, our *class-based* PARTICUL approach improves upon *vanilla*-PARTICUL by learning more specific detectors. Finally, it is interesting to point out that our results are stable, with a standard deviation of the AUROC/AUPR/FPR80 scores of less than 10% in the vast majority of our experiments, and that the number of detectors (4 or 6) does not have a significant impact on the results.

Perturbation P	Description	Values Λ	Sign
Blur	Gaussian blur with kernel 3×3, standard deviation λ	λ= 0.0 (no blur), 0.1, 0.3, 0.5, 0.7, 1.0 2.0, 5.0, 10.0	-
Noise	Gaussian noise with ratio λ	λ= 0.0 (no noise), 0.1, 0.2, 0.3, 0.4, 0.5 0.7, 1.0	-
Brightness	Blend black image with ratio $1 - \lambda$	λ= 0.1, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 (no change)	+
Rotation forth	Rotation with degree λ	λ= 0 (no rotation), 10, 20, 30, 40, 50, 70, 90, 110, 130, 150, 180	-
Rotation back	Rotation with degree λ	λ= 180, 210, 240, 270, 300, 320, 340, 350, 360 (no rotation)	+

Table 2: Description of the perturbations used in this paper. For each perturbation, we define the variable parameter λ, its range of values Λ and the expected sign of the correlation. '+' (resp. '-') indicates that the confidence measure is expected to increase (resp. decrease) with the intensity of the perturbation. For gaussian noise, we dynamically compute a standard deviation for each input x as $\sigma(x) = \lambda \times \big(\max(x) - \min(x)\big)$ (similar to the method used in Smilkov et al. (2017)).

Sensitivity to dataset perturbation In this experiment, we study the behaviour of confidence measures using various perturbations described in Table 2. Due to its instability during the previous experiment and its weakly supervised setting, C_M^{fssd} is not considered here. As shown in Table 3, the three confidence measures under test exhibit very different behaviors depending on the nature of the perturbation, but also sometimes depending on the target dataset. Indeed, although C_M^{MCP} seems sensitive to blur and brightness perturbations, it is not correlated with gaussian noise, opening the door to adversarial attacks (Szegedy et al. (2014); Hein et al. (2019)). Conversely, while C_M^{fnrd} is strongly sensitive to noise, any perturbation reducing the amplitude of neuron activation values (blur, brightness) has the opposite effect of increasing this confidence measure. In general, both C_M^{vP} and C_M^{cP} correlate with each type perturbation. However, we notice that results on CUB200 (and Stanford Cars, in a lesser extent) for blur and brightness suffer the same issue as C_M^{fnrd} (inverse correlation) when using our baseline classifier (ResNet50). Finally, rotations seem harder to detect, with softmax and PARTICUL-based measures achieving similar results, while C_M^{fnrd} is only weakly correlated to the rotation angle. Interestingly, when using a VGG19BN (Simonyan & Zisserman (2015)) classifier trained on CUB200, we obtain better results for C_M^{MCP} (noise sensitivity) and our PARTICUL-based approach (blur, brightness), while confirming the drawbacks of C_M^{fnrd} (blur, brightness). This indicates that not only does the robustness of a given measure depends on the type of perturbation and on the IoD dataset, but it seems to also rely on the architecture of the classifier.

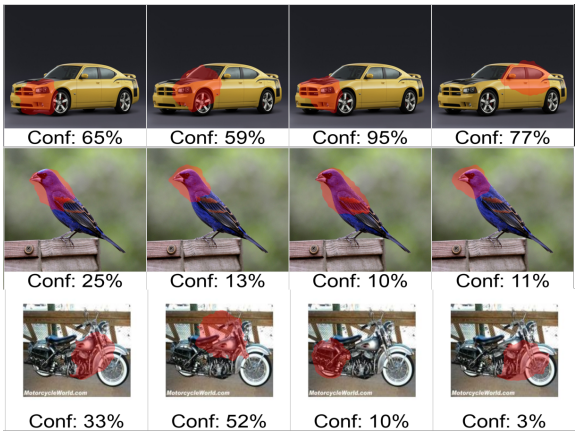
Arch.	$\mathcal{D}_{ind}$	Pert. P	MCP	fNRD	*vanilla* PARTICUL $P=4$	*vanilla* PARTICUL $P=6$	*class-based* PARTICUL $P=4$	*class-based* PARTICUL $P=6$
Resnet50	CT	Blur (-)	-1.00	1.00	-0.91 $\pm$ 0.06	-0.95 $\pm$ 0.00	-0.95 $\pm$ 0.00	-0.95 $\pm$ 0.00
	CB		-0.90	1.00	0.95 $\pm$ 0.00	0.95 $\pm$ 0.00	0.95 $\pm$ 0.00	1.00 $\pm$ 0.00
	SC		-1.00	1.00	-0.25 $\pm$ 0.60	0.50 $\pm$ 0.14	-0.85 $\pm$ 0.00	-0.86 $\pm$ 0.03
	CT	Noise (-)	0.05	-1.00	-1.00 $\pm$ 0.00	-1.00 $\pm$ 0.00	-1.00 $\pm$ 0.00	-1.00 $\pm$ 0.00
	CB		-0.24	-1.00	-0.69 $\pm$ 0.00	-0.68 $\pm$ 0.08	-0.78 $\pm$ 0.10	-0.67 $\pm$ 0.08
	SC		-0.29	-1.00	-0.77 $\pm$ 0.03	-0.82 $\pm$ 0.01	-0.61 $\pm$ 0.61	-0.93 $\pm$ 0.12
	CT	Brightness (+)	0.98	-1.00	0.97 $\pm$ 0.04	0.98 $\pm$ 0.04	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00
	CB		1.00	-0.31	-0.01 $\pm$ 0.52	0.23 $\pm$ 0.11	0.78 $\pm$ 0.20	0.67 $\pm$ 0.20
	SC		1.00	-1.00	0.98 $\pm$ 0.03	0.90 $\pm$ 0.18	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00
	CT	Rotation forth (-)	-0.30	-0.24	-0.41 $\pm$ 0.18	-0.42 $\pm$ 0.05	-0.51 $\pm$ 0.08	-0.52 $\pm$ 0.07
	CB		-0.95	-0.42	-0.82 $\pm$ 0.20	-0.94 $\pm$ 0.01	-0.94 $\pm$ 0.01	-0.93 $\pm$ 0.00
	SC		-0.76	-0.56	-0.71 $\pm$ 0.06	-0.67 $\pm$ 0.09	-0.86 $\pm$ 0.08	-0.89 $\pm$ 0.08
	CT	Rotation back (+)	0.52	0.20	0.51 $\pm$ 0.09	0.44 $\pm$ 0.09	0.68 $\pm$ 0.10	0.73 $\pm$ 0.06
	CB		0.95	0.32	0.77 $\pm$ 0.32	0.94 $\pm$ 0.01	0.98 $\pm$ 0.03	0.98 $\pm$ 0.03
	SC		0.72	0.32	0.78 $\pm$ 0.06	0.74 $\pm$ 0.03	0.84 $\pm$ 0.05	0.89 $\pm$ 0.07
VGG19BN	CB	Blur (-)	-0.86	1.00	-1.00 $\pm$ 0.00	-1.00 $\pm$ 0.00	-1.00 $\pm$ 0.00	-0.96 $\pm$ 0.06
		Noise (-)	-1.00	-1.00	-1.00 $\pm$ 0.00	-1.00 $\pm$ 0.00	-0.92 $\pm$ 0.06	-0.97 $\pm$ 0.05
		Brightness (+)	0.98	-0.98	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00	1.00 $\pm$ 0.00
		Rotation forth (-)	-0.95	-0.42	-0.95 $\pm$ 0.00	-0.95 $\pm$ 0.00	-0.95 $\pm$ 0.00	-0.95 $\pm$ 0.00
		Rotation back (+)	0.95	0.35	0.94 $\pm$ 0.01	0.92 $\pm$ 0.01	0.92 $\pm$ 0.01	0.92 $\pm$ 0.01

Table 3: Spearman rank correlation coefficient between the intensity λ of a perturbation applied to dataset $\mathcal{D}$ and the average confidence measure $\Gamma(C_M, \mathcal{D}, P, \lambda)$ computed over the dataset. For each perturbation P, the expected correlation sign is recalled in parenthesis. In red, we highlight experiments where the distributions are either uncorrelated (small correlation coefficient), inversely correlated (opposite coefficient sign) or unstable (high standard deviation for PARTICUL-based measures). For CUB200, we also provide the results of the experiments performed using a VGG19 classifier (with batch normalization). Best viewed in color.

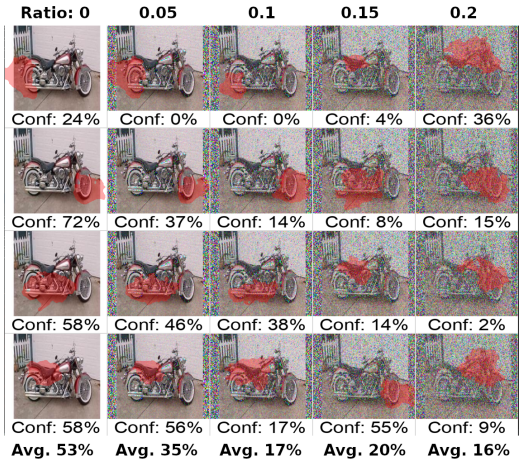
A step towards more explainable confidence measure One of the key advantages of the PARTICUL-based confidence measure resides in the possibility of visualizing the detected patterns during the inference of the classifier in order to trace back the contribution of each detector. As in Xu-Darme et al. (2022), we use the Smoothgrads (Smilkov et al. (2017)) algorithm to highlight the most salient area of the image for each detector. As illustrated in Fig. 3a, images from D_{iod} can be used to extract a visual reference of the pattern highlighted by each detector. When processing a new image through the classifier M, patterns can be compared to these visual references to better understand the confidence score. For example, some detectors trained on the StanfordCars dataset seem to be able to recognize similar patterns on an image of a motorcycle (row 3), which is to be expected from a visual point of view. Secondly, visualization can also help illustrate the behaviour of each detector *w.r.t.* a perturbation of the image, as illustrated in Fig. 3c and Fig. 3b. In particular, it is interesting to notice that contrary to our initial intuition, our PARTICUL-based confidence measure does not steadily decrease for angles ranging from 0° to 180°, but rather displays some invariance to horizontal symmetry (Fig.3d). Finally, as shown in Fig. 3b, although each of our detectors might not be strongly sensitive to a given perturbation, averaging scores across all detectors partially mitigates the variations of individual detectors and creates an ensemble effect that improves the overall stability of the metric, without the cost of training several classifiers (Nguyen et al. (2020)).

6 Conclusion & Future Works

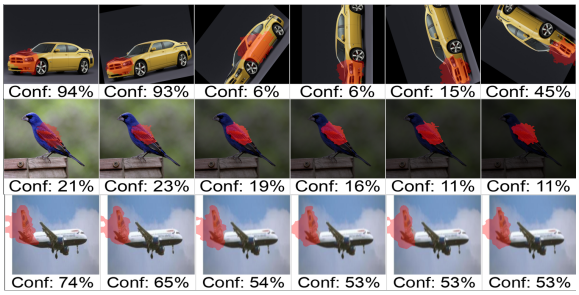
In this paper, we demonstrated how the detection of recurring patterns, using the PARTICUL algorithm, can help to build confidence measures with the ability to discriminate between IoD and OoD inputs, while allowing a form of visualisation of the detected patterns. We tested our approach in two modalities (cross-dataset OoD and perturbation OoD) and showed that it offers mostly consistent results across all experiments. However, we also showed that the robustness of all tested metrics can depend on the architecture of the underlying classifier, and we believe that experiments with multiple architectures should be systematically



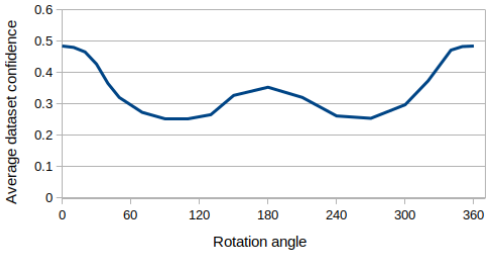
(a) Pattern visualization and confidence scores of 4 detectors (one per column) trained using *vanilla* PARTICUL on StanfordCars and applied to images from StanfordCars (row 1), CUB200 (row 2) and Caltech101 (row 3). The first image can be used as a visual reference.



(b) Impact of increasing gaussian noise ratio on four detectors trained with *class-based* PARTICUL on Caltech101 (*motorbikes* class. Averaging the confidence scores across all detectors creates an ensemble effect for more robustness.).



(c) Evolution of the confidence score of three detectors trained on StanfordCars (row 1), CUB200 (row 2) and Caltech101 (row 3) respectively *w.r.t.* to rotation (row 1), changes in brightness (row 2) and gaussian blur (row 3).



(d) Evolution of the average confidence of a *vanilla* PARTICUL detector across the Stanford Cars dataset for various rotation angles. Rather than the expected V-shape curve, we observe a local maxima for 180° rotation, indicating some form of invariance *w.r.t.* horizontal symmetry (see (c), row 1).

Figure 3: Using pattern visualization to better understand our PARTICUL-based confidence measure.

conducted in future proposals for measuring Out-of-Distribution. Moreover, since the datasets we used were not strictly exclusive (Caltech101 contains images of both birds and cars), a proper ablation study (retraining a classifier without these classes) should be performed to verify whether the separability of distributions actually improves.

Reproducibility All the code necessary to reproduce our experimental results is available at the following link: https://anonymous.4open.science/r/iclr23_submission_596/

Dataset	Architecture	Image size	Accuracy	# epochs	Optimizer		
					alg	learn. rate	decay/momentum
Caltech 101	Resnet50	224	86.8%	30	Adam	10^{-3}	/
CIFAR 100			71.3%	30	Adam	10^{-3}	/
CUB200		448	83.6%	80	SGD	10^{-3}	10^{-4} / 0.9
StanfordCars			90.5%	80	SGD	10^{-3}	10^{-4} / 0.9
CUB200	VGG19BN	448	83.0%	100	SGD	10^{-3}	10^{-4} / 0.9

Table 4: Baseline classifiers training parameters.

Table 4 summarizes the parameters used when training our baseline classifiers. Note that for CUB200 and Stanford Cars, we perform data augmentation through a random cropping of size 448×448 on the training images (for CIFAR100 and Caltech101, no data augmentation is used) and reduce the learning rate by a factor 10 every 30 epochs.

For reproducibility purposes, we performed 3 random initializations of the detectors for both cross-dataset and dataset perturbation OoD settings. We are providing averaged scores along with corresponding unbiased standard deviation. In our tables, average score is on the left of the $\pm$ symbol, standard deviation is on the right.

When training our PARTICUL detectors (either in the *vanilla* or *class-based* versions), we use RMSprop optimizer, with learning rate 10^{-4} and decay 10^{-5}, for 30 epochs. We do not use any data augmentation except resizing each image to the appropriate size. In all experiments, we try to maximize the diversity of our detectors by setting the ratio λ of the Unicity constraint to 1 (see Xu-Darme et al. (2022)).

For the implementation of FSSD, we used the code available on GitHub (https://github.com/megvii-research/FSSD_OoD_Detection), with the following modifications:

- Allow the choice of a validation set $D_{ood,val} \not\sim D_{ood}$ for a fairer comparison betwen methods
- Use the entire test sets during the evaluation rather than subsampling D_{ood} to match the size of D_{iod} (would improve AUPR score in some cases).

Note that during training, we did not perform hyperparameter tuning due to its computational cost.

REFERENCES

Charles Corbiere, Nicolas Thome, Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Perez. Confidence estimation via auxiliary models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/tpami.2021.3085983. 2

Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing Objects by their Attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785, 2009. doi: 10.1109/CVPR.2009.5206772. 2

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pp. 1050–1059. PMLR, 2016. 2

Junwei Han, Xiwen Yao, Gong Cheng, Xiaoxu Feng, and Dong Xu. P-cnn: Part-based convolutional neural networks for fine-grained visual categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):579–590, 2022. doi: 10.1109/TPAMI.2019.2933510. 2

Kai Han, Jianyuan Guo, Chao Zhang, and Mingjian Zhu. Attribute-Aware Attention Model for Fine-grained Representation Learning. *Proceedings of the 26th ACM international conference on Multimedia*, 2018. 2

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90. 5

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41–50, 2019. 2, 6, 7

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. URL https://openreview.net/forum?id=Hkg4TI9xl. 1

Darryl Hond, Hamid Asgari, Daniel Jeffery, and Mike Newman. An integrated process for verifying deep learning classifiers using dataset dissimilarity measures. *International Journal of Artificial Intelligence and Machine Learning*, 11(2):1–21, jul 2021. doi: 10.4018/ijaiml.289536. 2, 5

Haiwen Huang, Zhihan Li, Lulu Wang, Sishuo Chen, Bin Dong, and Xinyu Zhou. Feature space singularity for out-of-distribution detection. *ArXiv*, abs/2011.14654, 2021. 2, 5

Susmit Jha, Sunny Raj, Steven Fernandes, Sumit K Jha, Somesh Jha, Brian Jalaian, Gunjan Verma, and Ananthram Swami. Attribution-based confidence metric for deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/bc1ad6e8f86c42a371aff945535baebb-Paper.pdf. 2

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 4

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018a. 1, 2

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *ArXiv*, abs/1807.03888, 2018b. 2

Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, 2022. URL https://data.caltech.edu/records/20086. 4, 5

Hao Li, Xiaopeng Zhang, Qi Tian, and Hongkai Xiong. Attribute Mix: Semantic Data Augmentation for Fine Grained Recognition. In *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pp. 243–246, 2020. doi: 10.1109/VCIP49819.2020.9301763. 2

Shiyu Liang, Yixuan Li, and R. Srikant. Principled detection of out-of-distribution examples in neural networks. *CoRR*, abs/1706.02690, 2017. URL http://arxiv.org/abs/1706.02690. 2

Tien Thanh Nguyen, Anh Vu Luong, Manh Truong Dang, Alan Wee-Chung Liew, and John A. W. McCall. Ensemble selection based on classifier prediction confidence. *Pattern Recognition*, 100:107104, 2020. 2, 8

Yuxin Peng, Xiangteng He, and Junjie Zhao. Object-Part Attention Model for Fine-grained Image Classification. *IEEE Transactions on Image Processing*, 27(3):1487–1500, March 2018. ISSN 1057-7149, 1941-0042. doi: 10.1109/TIP.2017.2774041. URL http://arxiv.org/abs/1704.01740. arXiv: 1704.01740. 2

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 7

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *ArXiv*, abs/1706.03825, 2017. 7, 8

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014. 1, 7

Fahim Tajwar, Ananya Kumar, Sang Michael Xie, and Percy Liang. No true state-of-the-art? ood detection methods are inconsistent across datasets. *ArXiv*, abs/2109.05554, 2021. 2

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, 2010. 2, 4, 5

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. 2, 3, 4, 8, 10

L. Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A Large-Scale Car Dataset for Fine-Grained Categorization and Verification. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3973–3981, 2015. 2, 4, 5

Xiangyun Zhao, Yi Yang, Feng Zhou, Xiao Tan, Yuchen Yuan, Yingze Bao, and Ying Wu. Recognizing Part Attributes With Insufficient Data. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 350–360, Seoul, Korea (South), October 2019. IEEE. ISBN 978-1-72814-803-8. doi: 10.1109/ICCV.2019.00044. URL https://ieeexplore.ieee.org/document/9009781/. 2

Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning Multi-attention Convolutional Neural Network for Fine-Grained Image Recognition. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5219–5227, Venice, October 2017. IEEE. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.557. URL http://ieeexplore.ieee.org/document/8237819/. 2