

AID: Attribute-Inference for out-of-set Detection

Francesco Dibitonto, Daniele Leto, Giacomo Zarbo

Politecnico di Torino

{francesco.dibitonto, s256960, giacomo.zarbo}@studenti.polito.it

Abstract

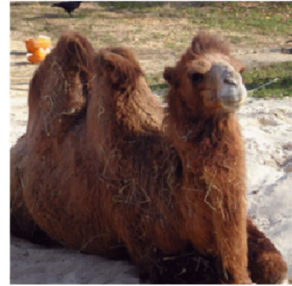
Collecting and labeling training samples starting from unlabeled data is a long and tedious task, as it requires great manual human effort. Moreover, collecting training samples that cover all possible target classes for training a classifier is an arduous labour, potentially even impossible in case some of those classes are yet to discover. Fulfill both tasks at the same time can be indeed more burdensome or even unfeasible to do. We propose to 'aid' humans to solve this composite problem with a simple but yet efficacious method. To address the first problem we use a Semi-Supervised Learning approach inspired by Domain Adaptation and Data Augmentation. To deal with the second (a.k.a. Open Set Recognition) problem we use attribute-inference to generate new labels for the out-of-set samples, starting from the known in-set samples. We demonstrate how these two approaches can be directly merged into a single one and into a single Deep Neural Network as well. We also show that, despite the simplicity and the plug-and-play nature of the method, the performances are satisfactory and the results are credible and reasonable. Our code is available at <https://github.com/danieleleto94/AID>

1. Introduction

Deep Neural Networks need a large quantitative of data in order to obtain adequate results. Gathering such amount of data and labelling all of it requires huge strain by humans. Semi-Supervised Learning (SSL) considerably decreases the labelling effort by training on both labeled and unlabeled data simultaneously. In this way the learned model is stronger than one trained only on the available labeled data, by learning additional information from the unlabeled data. One of the major problems of this method is that, if the empirical distributions of the labeled and unlabeled data don't match, the addition of unlabeled data is detrimental instead of being beneficial for learning a better model. Luc Van Gool et al. [1] tackle this problem by healing such empirical distribution mismatch by Aug-



True unknown class: panther
Predicted inferred class: black lion



True unknown class: camel
Predicted inferred class: slow horse

Figure 1. Two out-of-set samples. Here it is shown the comparison between the real class, unknown to the network at training time, and the inferred class assigned through our approach.

mented Distribution Alignment, basically consisting of a mix of adversarial training inspired by Domain Adaptation Neural Networks (DANN) [2] and data augmentation. Due to the immediateness yet the effectiveness of the method, we decided to use their ADA-Net architecture as the core of our work.

Whenever a target sample is classified, it is not said that the class to be assigned is one of the known training classes. In such a case the classifier should be smart enough to understand that the sample is something new, thus it should not proceed by erroneously assign a class label from the available ones, but rather it should question itself if it's the case to assign a new label based on the knowledge accumulated until that moment. But how exactly?

Constantine John Phipps was the first to describe the polar bear as a distinct species in 1774¹, during one of his voyages towards the North Pole, but the name he assigned was a different one. Of course he already knew what a bear is, nonetheless he got he just saw a new variation in terms of species. Because of the fact he found the animal wandering next to the sea, he rather named it 'Ursus maritimus', the Latin for 'maritime bear'. So he chose this name by relating the animal family (Ursus) with the animal's native habitat. To assign such a name, he exploited both his knowl-

¹https://en.wikipedia.org/wiki/Polar_bear

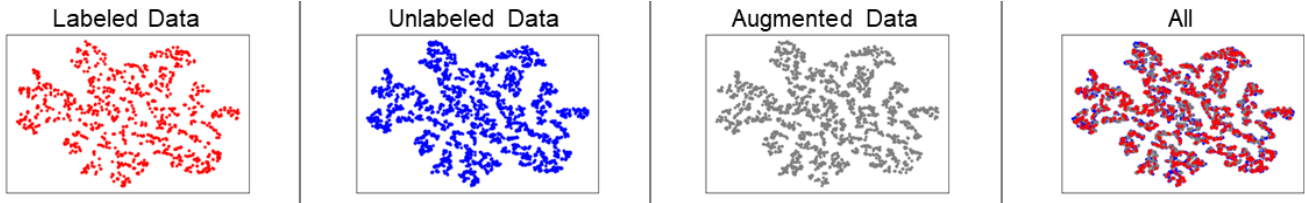


Figure 2. Visualization of the Awa2 features got by AID with t-SNE. A comparison between labeled, unlabeled, and augmented mixed samples is shown. By Augmented Distribution Alignment and Data Augmentation with AugMix we bridge the gap between the labeled and unlabeled distributions, as shown in the rightmost image.

edge about the animal family (base class) and the animals’ habitats (attribute). We call this kind of logic to assign new classes **attribute-inference**.

We train a network to learn classes jointly with attributes by Multi-Task Learning, where the attributes are used in case the target sample is sufficiently different from the most similar known class for attribute-inference. But how to determine if the target sample is sufficiently different in terms of variation, to constitute a new class? We exploit the link between attribute-inference and out-of-distribution detection (OODD) [3]. OODD is a method for determining whether a sample is inside or outside a certain set/distribution. Hence we include it in our work to address the Open Set Recognition (OSR) problem. OODD allows us to find a threshold value by which we can decide whether the network is enough confident about knowing the target sample (and consequently, its label), or it is hesitant to assign a label from the known classes; that’s because it perceived a big enough difference with respect to what it already knows. In the latter case it will then proceed by using the attribute-inference logic to output the class label.

2. Related Work

Semi-Supervised Learning and Data Augmentation: Many recent SSL methods further enhance the SSL baseline. In particular, Data Augmentation methods proved to be able to consistently improve performances in conjunction with Cross-Set Sampling [4]. The basic idea is that the samples interpolated from the labeled and the unlabeled distributions enlarge the training dataset and also describe a distribution which should be closer to the real one. Among such methods, the ADA-Net baseline employs MixUp to generate mixed samples. In addition, we also tried the most recent CutMix [5] instead, comparing then the results. Finally we also apply strong data augmentation on both the labeled and the unlabeled set, inspired by AugMix [6].

Transfer Learning: Another method which proved to be useful in this context is Transfer Learning, which consists of exploiting the ‘knowledge’ from an already learned task for a new but related one. Such ready to use knowledge thus reduces the amount of needed data. Zheng et al. [7] questioned about the usefulness of the method when used

together with SSL. They conclude that the utility of combining the two methods depends on many conditions, like the kind of data and the gap in terms of amount of labeled and unlabeled data. Nonetheless it yields general improvements and prevents overfit if used correctly. Thus we decide to use a model pretrained on ImageNet to speed up the learning process and we analyze the exhibited performances.

Zero-Shot Learning: Recently a new family of approaches has been in vogue, namely Zero-Shot Learning [8]. These methods are usually semantics-based and they aim to address the OSR problem by starting with a few or no labeled train data. They establish relationship between classes by means of some ‘external knowledge’ (e.g. attributes and knowledge base). The problem of these approaches is that the effort of labeling data is moved to the creation of the knowledge base. Compared to this line, our method is self-contained in the sense that its external knowledge consists on train classes and their attributes only, which are also learned during training. So what it’s known is also learned, without relying on any additional source of truth.

Out-of-distribution Detection: Hendrycks et Al. [3] layed the basis for detecting samples which are out of the training set distribution. Their intuition is that softmax probabilities can be used for confidence estimation when used together with evaluation metrics like AUROC and AUPR. ODIN [9] enhanced the previous baseline by using Temperature Scaling and by perturbing the input, leading to a better separation between the in- and out-of-distribution images for a better OODD. We incorporated ODIN into our approach, with the only difference that we only used Temperature Scaling, as perturbations didn’t introduce any significant or stable improvement in our experiments.

3. Problem Statement

In this work we consider two problems: reduce the effort for manually labelling samples with SSL, supported by adversarial training, and find out whether test samples are novelties with OODD, allowing us to get whether attribute-inference has to be used. From now on we will refer at in-distribution samples as in-set (IS) and out-of-distribution

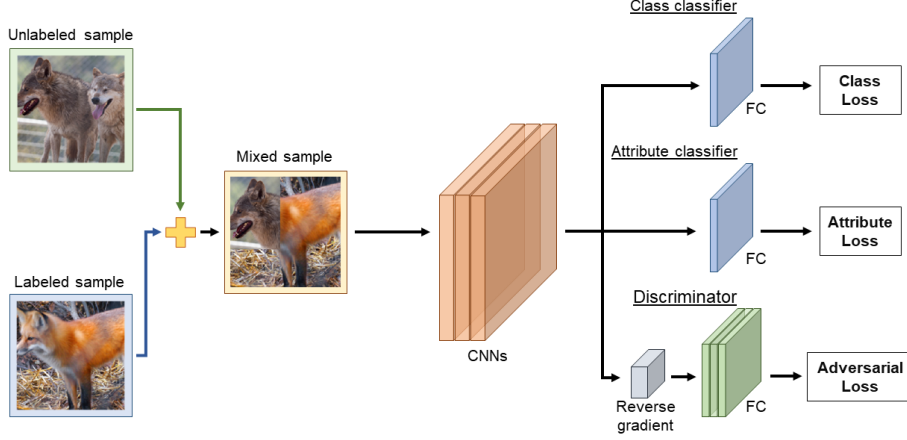


Figure 3. The network architecture of our approach. Three branches are used: a standard class classifier, an attribute classifier to predict attributes labels and a discriminator with a GRL to distinguish between the labeled and unlabeled distributions. In a training step a new image is created with the CutMix technique from a labeled and an unlabeled sample and then it is fed into the network.

samples as out-of-set (OOS).

In SSL we have two sets of training samples. We have a labeled train set whose samples are denoted by $D_l = \{(x_l^1, y_1), \dots, (x_l^n, y_n)\}$ where x_l^i, y_i are the i -th sample and label respectively, and n is the number of these samples. We also have a larger unlabeled train set whose samples are denoted by $D_u = \{x_u^1, \dots, x_u^m\}$ where x_u^i is the i -th sample, and m is the number of these samples. Within the Domain Adaptation scenario, the KL-Divergence is used to measure the distribution divergence between the source and target domain. Here instead we use it for measuring the distribution divergence between the labeled and the unlabeled samples. Thus we use the KL-divergence Loss to quantify how much the two probability distributions differ from each other. The KL-Loss is defined as:

$$\mathcal{L}_1 = \frac{1}{n} \sum_{i=1}^n target_i * [\log(target_i) - \log(prob_i)] \quad (1)$$

where in this case *target* are the true probabilities of belonging to each of the two distributions (labeled or unlabeled) of the i -th interpolated sample and *prob* are the predicted distribution probabilities given by the output of the network when fed with the same i -th interpolated sample. Exactly like in the ADA-Net work, we employ such loss to minimize the classification loss of a binary discriminator which predicts 0 and 1 for labeled and unlabeled samples respectively. Then a gradient reverse layer (GRL) is used like in the DANN work to enforce to get features invariant with respect to the distribution, thus minimizing the distribution distance $d_K(D_l, D_u)$.

Concerning the Cross-set Sample Augmentation technique, the mathematical derivations will be not furtherly discussed as they were already shown in-depth both in the ADA-Net and the MixUp papers. Nonetheless, the general

idea is that mixed samples are generated by interpolating the labeled and the unlabeled images. They can be considered as samples belonging to the intermediate distributions between the labeled and the unlabeled samples empirical distributions, thus bridging the gap between them. Together with each mixed image, also a mixed label (pseudo-label) is generated with the same logic. Then, by feeding such mixed additional samples into the network, the robustness of the model is enhanced. The loss function for measuring the difference between the mixed predicted labels and the true mixed labels $d_K(l_{pred}, l_{true})$ is again the KL-Loss (1) \mathcal{L}_2 , where *target* are the true probabilities of belonging to each class of the i -th interpolated sample and *prob* are the class predicted probabilities given by the output of the network when fed with the same i -th interpolated sample.

By summing up the previous two loss functions, we define a new loss function, namely 'Adversarial Loss', for the whole mixing process:

$$\mathcal{L}_A = \mathcal{L}_1 + \mathcal{L}_2 \quad (2)$$

Now about attributes to be used for attribute-inference, the metric we chose to use for assessing the performances of our model is the Hamming score. For each sample, let $\hat{z}_i \in [0, 1]$ be the predicted value of the i -th attribute and $z_i \in [0, 1]$ be the true value of the i -th attribute and n the total number of attributes. So the Hamming score (for each sample) is:

$$\mathcal{S}_H(z, \hat{z}) = \frac{1}{n} \sum_{i=1}^n 1(\hat{z}_i = z_i) \quad (3)$$

To measure the error onto the attributes, we decide to use the Binary Cross Entropy Loss:

$$\mathcal{L}_B = -\frac{1}{n} \sum_{i=1}^n [z_i \log(p_i) + (1 - z_i) \log(1 - p_i)] \quad (4)$$


 Predicted class: wolf True class: wolf	black: 1/1	white: 0/1	blue: 0/0	brown: 1/1	gray: 1/1	orange: 0/0	red: 0/0	yellow: 0/0	patches: 0/0	spots: 0/0
	stripes: 0/0	furry: 1/1	hairless: 0/0	toughskin: 0/0	big: 1/1	small: 0/0	bulbous: 0/0	lean: 1/1	flippers: 0/0	hands: 0/0
	hooves: 0/0	pads: 1/1	paws: 1/1	longleg: 0/0	longneck: 0/0	tail: 1/1	chewteeth: 1/1	meatteeth: 1/1	buckteeth: 0/0	strawteeth: 0/0
	horns: 0/0	claws: 1/1	hubs: 0/0	smelly: 1/0	flies: 0/0	hops: 0/0	swims: 0/0	tunnels: 0/0	walks: 1/1	fast: 1/1
	slow: 0/0	strong: 1/1	weak: 0/0	muscle: 1/1	bipedal: 0/0	quadrupedal: 1/1	active: 1/1	inactive: 0/0	nocturnal: 0/1	hibernate: 0/0
	agility: 1/1	fish: 0/0	meat: 1/1	plankton: 0/0	vegetation: 0/0	insects: 0/0	forager: 0/1	grazer: 0/0	hunter: 1/1	scavenger: 0/1
	skimmer: 0/0	stalker: 1/1	newworld: 1/1	oldworld: 1/1	arctic: 0/1	coastal: 0/0	desert: 0/0	bush: 0/0	plains: 1/1	forest: 0/1
	fields: 0/0	jungle: 0/0	mountains: 0/1	ocean: 0/0	ground: 1/1	water: 0/0	free: 0/0	cave: 0/1	feria: 1/1	timid: 0/0
	smart: 1/1	group: 0/1	solitary: 1/1	nestspot: 0/0	domestic: 0/0					

Figure 4. A random sample of the validation phase and its labels. The table on the right represents the comparison between the predicted and the true attribute labels (predicted/true). Colors indicate whether the prediction is right (green) or wrong (red).

where z_i in the true attribute and p_i is the predicted attribute of the i -th sample.

Regarding the error between the true and the predicted class labels we use the Cross Entropy Loss:

$$\mathcal{L}_C = - \sum_{\forall x} p(x) \log(q(x)) \quad (5)$$

where x is a single class, $p(x)$ is the true distribution and and $q(x)$ is the estimated distribution.

Finally, we define the total loss as the sum of the previous losses:

$$\mathcal{L}_{tot} = \mathcal{L}_A + \mathcal{L}_B + \mathcal{L}_C \quad (6)$$

The gradients of this total loss are then calculated with respect to the input weights by backpropagation.

Apart from the loss optimization problem, we follow the ODIN approach for detecting the OOS samples to be used for attribute-inference with Temperature Scaling:

$$S_i(x; T) = \frac{\exp(f_i(x)/T)}{\sum_{j=1}^n \exp(f_j(x)/T)} \quad (7)$$

where $f_i(x)$ are the unnormalized log-probabilities for the input x , given a network $f = (f_1, \dots, f_N)$ trained to classify N classes. $S_i(x; T)$ are the normalized probabilities. $T \in \mathbb{R}^+$ is the temperature scaling parameter which serves to furtherly separate the softmax scores of IS and OOS samples after suitable tuning.

4. AID: Attribute-Inference for OOS Detection

In this section, we present our AID method, for aiding humans at reducing the manual effort of labelling data and collecting samples for all possible target classes. Moreover, when spotting OOS samples, these can be arbitrarily submitted for a re-analysis to experts, who this time are guided by the hints provided by attribute-inference. Thus AID is built upon two components to accomplish the two tasks just mentioned: a SSL revised ADA-Net core for training and an ODIN-based out-of-detection method with only Temperature Scaling for testing. Now we proceed by describing each of the two components.

4.1. ADA-Net for Attribute-Inference

ADA-Net is the hearth of our method. It is basically a unified framework for Augmented Distribution Alignment based on both Cross-Set sample Augmentation and Adversarial Distribution Alignment. We implemented such method by changing the network of the baseline with a stronger ResNet core, as additional network capacity is needed for running our experiments on complex datasets. We have a standard classification branch for predicting class labels, then we have also an additional classification branch for predicting attribute labels, to be used for attribute-inference. Last but not least, we have a discriminator to distinguish among labeled and unlabeled samples and a GRL preceding the discriminator to reverse the gradient sign at backpropagation time. Base classes, attributes, and distributions are learned jointly in a Multi-Task fashion. During training, interpolated samples and labels are generated by Cross-Set Sample Augmentation for each mini-batch. For such mixed samples generation we adapt the baseline to use the newer CutMix augmentation strategy: we cut patches from the unlabeled images and we paste them on top of the labeled images. Moreover, we furtherly modify the baseline by augmenting both the labeled and the unlabeled sets through the AugMix transformations (a mix of various transformations including rotations, translations, contrast and color jitter etc.). This is done to introduce more variations in terms of inputs into the network, leading to better generalization. Also the mixed samples generated upon the augmented labeled and unlabeled sets will much more vary as well, filling better the gap between the two distributions. The network is so optimized by standard propagation.

4.2. Temperature Scaling for OOD

We employ Temperature Scaling for easier detection of OOS images. By tuning the Temperature of the final softmax layer for base class predictions, the model will produce a probability distribution over the classes such that the OOS images are more distinguishable. As we use this method at test time only, it doesn't add any computational overhead at training time, making it suitable for our light approach.

Pseudocode of a train step of the network.

Input: A batch of labeled samples $\{(\mathbf{x}_l, y_l, \mathbf{z}_l), \dots\}$, a batch of unlabeled samples $\{\mathbf{x}_u, \dots\}$, a class classifier f , an attribute classifier g and a discriminator h .

1. Run one forward step to get class labels \hat{y}_l and attributes labels $\hat{\mathbf{z}}_l$ for labeled samples.
2. Compute the Binary Cross-Entropy (4) and the Cross-Entropy (5) losses.
3. Run a second forward step to get pseudo-labels \hat{y}_u from the unlabeled samples.
4. Obtain the mixed images \mathbf{x}_m from the labeled images \mathbf{x}_l and the unlabeled images \mathbf{x}_u and the mixed-labels y_m from the labels y_l and the pseudo-labels \hat{y}_u .
5. Run a third forward step to get the predicted mixed-labels \hat{y}_m and the predicted mixed-domains \hat{d}_m from the mixed samples \mathbf{x}_m and compute the respective KL losses (1).
6. Compute the adversarial loss (2).
7. Compute the total loss (6) and perform the backward pass.

Output: The class classifier f , the attribute classifier g and the discriminator h .

Pseudocode of a validation step of the network.

Input: A batch of samples $\{(\mathbf{x}, y, \mathbf{z}), \dots\}$ from the validation set.

1. Run one forward step to get predicted class labels \hat{y} and predicted attributes labels $\hat{\mathbf{z}}$.
2. Compute the Hamming score (3) for the attribute labels $\hat{\mathbf{z}}$ and the accuracy for the class labels \hat{y} .

Output: The Hamming score and the accuracy value.

5. Experiments

We evaluate our AID method on the Animals with Attributes (AwA2) dataset [10] because of the availability of the attributes for all classes. The choice of this dataset allows us to show more comprehensible results than if we chose a less intuitive kind of data (e.g. medical dataset). We take from the dataset 3111 labeled and 18661 unlabeled samples, with a labeled-unlabeled ratio of 1/6. The sets used for validation and final testing contain 3733 and 3354 samples respectively. The samples of the final test set are taken half from AwA2 IS samples we held out on purpose and half from OOS animal classes of Caltech256. The latter are used to test the attribute-inference mechanism.

5.1. Experimental Setup

We adopt three network architectures from the ResNet family as our core, namely ResNet18, ResNet34 and ResNet50. All networks are trained for 30 epochs with Adam. All the starting models are pre-trained on ImageNet.

Pseudocode of a test step of the network.

Input: A batch of samples $\{(\mathbf{x}, y, \mathbf{z}), \dots\}$ from the test set.

1. Run one forward step to get predicted class labels \hat{y} and predicted attributes labels $\hat{\mathbf{z}}$.
2. Apply Temperature Scaling to the predicted class labels \hat{y} , getting rescaled predictions \hat{y}_T .
3. Compute the softmax score $S_{\hat{y}_T}(\mathbf{x}; T)$ for the rescaled predictions \hat{y}_T .
4. For each $S_{\hat{y}_T}(\mathbf{x}_i; T) < \delta$:
 - Sum the number of the samples which were correctly classified as out-of-set $\rightarrow S_{out}$
 - apply attribute-inference for each \mathbf{x}_i to get a new class
- For each $S_{\hat{y}_T}(\mathbf{x}_i; T) > \delta$:
 - Sum the number of the samples whose predicted class label \hat{y} coincide with the true known label $\rightarrow S_{in}$
5. Calculate the batch accuracy as $(S_{out} + S_{in})/batch_size$.

Output: The accuracy value.

The batch size is set to 32. The initial learning rate is set to 0.0001, and it is divided by 10 when we reach 50% and 75% of the epochs. For the additional attribute classifier we use a single Fully Connected (FC) layer with 85 output neurons, as we have 85 different attributes. For the distribution-domain classifier, two FC layers with 1024 output neurons each, and a final FC layer with two output neurons for predicting the distribution-domain labels. As threshold value δ to distinguish IS and OOS samples we find a new δ for each experiment by grid search. The adversarial loss (2) is weighted by a constant coefficient α of 0.3 by grid search.

5.2. Evaluation Metrics

For measuring the performances of the model to correctly classify class labels and attribute labels we use respectively the two following metrics (note that the Hamming score is calculated only during validation, not in the final test as we don't have attributes for OOS samples):

- **Top-1 and Top-5 Accuracy** (higher is better)
- **Hamming score** (higher is better)

For measuring the effectiveness of the model to distinguish IS and OOS images we use these three metrics:

- **FPR at 95% TPR (FPR95):** represents the probability that an OOS sample is misclassified as IS when the true positive rate (TPR) is 95% (lower is better).
- **Detection Error (DE):** measures probability of misclassification when TPR is 95% (lower is better).
- **AUROC:** is the area under the curve depicting how much the model can distinguish between positive (IS) class and negative (OOS) class (higher is better).

Method	ResNet18	ResNet34	ResNet50
Baseline (MixUp)	1.21 - 0.51 - 0.71	1.25 - 0.50 - 0.76	1.4 - 0.50 - 0.78
MixUp+AugMix	1.23 - 0.50 - 0.74	1.3 - 0.50 - 0.78	1.4 - 0.53 - 0.80
CutMix	1.3 - 0.47 - 0.76	1.3 - 0.57 - 0.77	1.36 - 0.56 - 0.82
CutMix+AugMix	1.24 - 0.50 - 0.76	1.28 - 0.46 - 0.78	1.23 - 0.58 - 0.83

Table 1. Temperature T - Threshold δ - Accuracy for base classes.

Metric	FTPR95	DE	AUROC
Softmax (baseline)	0.556	0.204	0.85
Temperature Scaling	0.463	0.186	0.89

Table 2. Confidence estimations obtained only for the best model, i.e. ResNet50 with CutMix+AugMix.

5.3. Experimental Results

We summarize the accuracies on the AwA2 dataset in Table 1. We report all the network architectures variants plus the data augmentation methods. As shown in Table 1, by changing the baseline by using CutMix instead of MixUp, performances improve. By adding into the game also AugMix, performances increase again. We show the OOD results in Table 2. We report the performances of the confidence estimation method based on the softmax outputs, first without (baseline) and then with Temperature Scaling (ODIN-like). In the latter case we observe an increase in terms of performances.

6. Future Directions

We aim to try our approach in fields where labeling and collecting images is much more hard, like medical, chemistry, genetics etc. We think in such fields the hints provided by attribute-inference would be much more beneficial to experts dealing with unknown new cases. We also aim to try our method with different amounts of labeled and unlabeled data ratios and see how it behaves. Moreover, we are thinking about a method by which once the network finds out that a new class exists, it is adapted by reflecting this change and is thus expanded (for example by adding an additional output neuron in the class classifier), but keeping consistency and not messing up with the already learned information.

7. Conclusions

In this work, we show an approach to aid humans at reducing their effort for collecting, labeling, and figuring out new unknown classes. We proved that despite the relative simplicity of the method, the results are logical and coherent. We test the method by trying diverse settings of parameters and we explain the reasoning behind the approach.

References

[1] Q. Wang, W. Li, and L. Van Gool. Semi-supervised learning by augmented distribution alignment. 2019.

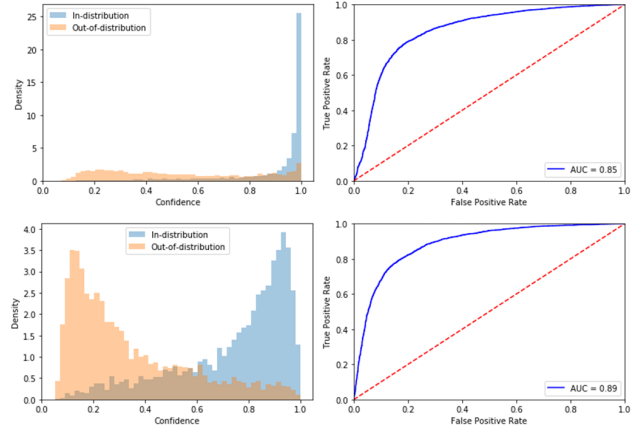


Figure 5. On the left: the two probability density functions of IS and OOD data, up for the OOD baseline, down with Temperature Scaling. We can easily notice how the temperature scaling makes much easier the task of learning confidence estimates by better separating IS and OOD samples. On the right: ROC curves of OOD baseline up and Temperature Scaling applied down.

[2] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. 2014.

[3] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. 2016.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. 2017.

[5] S. Yun, D. Han, S. Joon Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019.

[6] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. 2019.

[7] H. Zhou, A. Oliver, J. Wu, and Y. Zheng. When semi-supervised learning meets transfer learning: Training strategies, models and datasets. 2018.

[8] Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning - the good, the bad and the ugly. 2017.

[9] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. 2017.

[10] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.