Towards Robust Fine-grained Recognition by Maximal Separation of Discriminative Features

Krishna Kanth Nakka[1][0000−0002−2381−6593] and
Mathieu Salzmann[1,2][0000−0002−8347−8637]

[1] CVLab, EPFL, Switzerland
[2] ClearSpace, Switzerland
{krishna.nakka, mathieu.salzmann}@epfl.ch

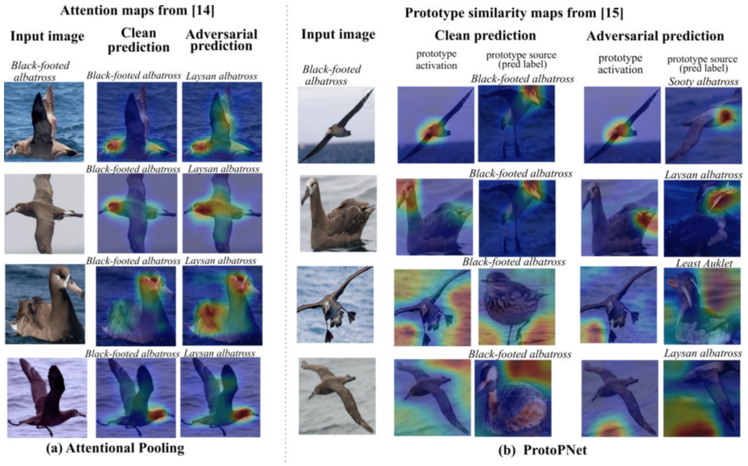
Abstract. Adversarial attacks have been widely studied for general classification tasks, but remain unexplored in the context of fine-grained recognition, where the inter-class similarities facilitate the attacker's task. In this paper, we identify the proximity of the latent representations of *local* regions of different classes in fine-grained recognition networks as a key factor to the success of adversarial attacks. We therefore introduce an attention-based regularization mechanism that maximally separates the latent features of discriminative regions of different classes while minimizing the contribution of the non-discriminative regions to the final class prediction. As evidenced by our experiments, this allows us to significantly improve robustness to adversarial attacks, to the point of matching or even surpassing that of adversarial training, but without requiring access to adversarial samples. Further, our formulation also improves detection AUROC of adversarial samples over baselines on adversarially trained models.

Keywords: Fine-grained Recognition, Adversarial Defense, Network Interpretability

1 Introduction

Deep networks yield impressive results in many computer vision tasks [1, 2, 3, 4]. Nevertheless, their performance degrades under adversarial attacks, where natural examples are perturbed with human-imperceptible, carefully crafted noise [5]. Adversarial attacks have been extensively studied for the task of general object recognition [6, 5, 7, 8, 9, 10], with much effort dedicated to studying and improving the robustness of deep networks to such attacks [11, 12, 13]. However, adversarial attacks and defense mechanisms for fine-grained recognition problems, where one can expect the inter-class similarities to facilitate the attacker's task, remain unexplored.

In this paper, we therefore analyze the reasons for the success of adversarial attacks on fine-grained recognition techniques and introduce a defense mechanism to improve a network's robustness. To this end, we visualize the image regions mostly responsible for the classification results. Specifically, we consider



(a) Attentional Pooling

(b) ProtoPNet

Fig. 1: **Interpreting adversarial attacks for fine-grained recognition.** We analyze the attention maps, obtained with [14](a) and [15](b), of four images from the *Black-footed albatross* class. Under PGD attack, these images are misclassified as closely-related bird species, such as *Layman albatross*, because the classifiers focus on either confusing regions that look similar in these classes, such as the bird's beak, or non-discriminative background regions, such as water.

both the attention-based framework of [14], closely related to class activation maps (CAMs) [16], and the recent prototypical part network (ProtoPNet) of [15], designed for fine-grained recognition, which relates local image regions to interpretable prototypes. As shown in Fig. 1, an adversarial example activates either confusing regions that look similar in samples from the true class and from the class activated by the adversarial attack, such as the beak of the bird, or, in the ProtoPNet case, non-discriminative background regions, such as water. This suggests that the latent representations of these confusing regions are close, and that the ProtoPNet classifier exploits class-irrelevant background information. These two phenomena decrease the margin between different classes, thus making the network more vulnerable to attacks.

Motivated this observation, we introduce a defense mechanism based on the intuition that the discriminative regions of each class should be maximally separated from that of the other classes. To this end, we design an attention-aware model that pushes away the discriminative prototypes of the different classes. The effectiveness of our approach is illustrated in Fig. 2, where the prototypes of different classes are nicely separated, except for those corresponding to non-discriminative regions. However, by means of an attention mechanism, we enforce these non-discriminative prototypes to play no role in the final class prediction. Ultimately, our approach reduces the influence of the non-discriminative regions on the classification while increasing the magnitude of the displacement in the latent space that the attacker must perform to successfully move the network's prediction away from the true label.

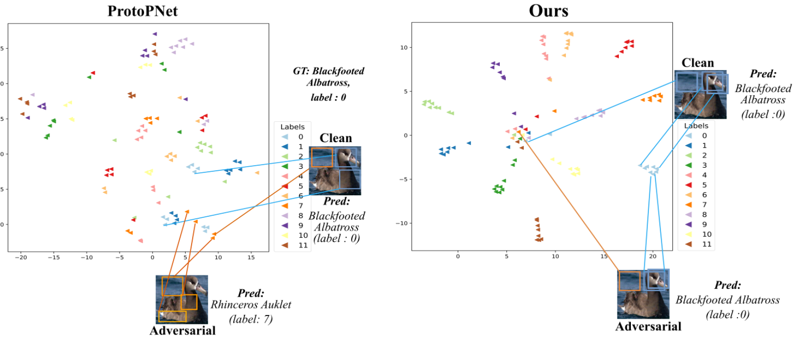


Fig. 2: **t-SNE visualization of the prototypes from 12 fine-grained classes of the CUB200 dataset.** In ProtoPNet [15], the prototypes of different classes are not well separated, making the network vulnerable to attacks. By contrast, our approach yields well-separated discriminative prototypes, while clustering the background ones, which, by means of an attention mechanism do not participate the prediction. This complicates the attacker's task.

As evidenced by our experiments, our approach significantly outperforms in robustness the baseline ProtoPNet and attentional pooling network, in some cases reaching adversarial accuracies on par with or higher than their adversarially-trained [17, 18] counterparts, but at virtually no additional computational cost.

Our main contributions can be summarized as follows. We analyze and explain the decisions of fine-grained recognition networks by studying the image regions responsible for classification for both clean and adversarial examples. We design an interpretable, attention-aware network for robust fine-grained recognition by constraining the latent space of discriminative regions. Our method improves robustness to a level comparable to that of adversarial training, without requiring access to adversarial samples and without trading off clean accuracy. Further, our approach improves the AUROC score of adversarial example detection by 20% over baselines for adversarial trained networks. We release the source code of our experiments at We release the source code of our experiments at `https://github.com/krishnakanthnakka/RobustFineGrained/`

2 Related Work

Adversarial Robustness. DNNs were first shown to be vulnerable to adversarial, human-imperceptible perturbations in the context of general image recognition. Such attacks were initially studied in [19], quickly followed by the simple single-step Fast Gradient Sign Method (FGSM) [5] and its multiple-step BIM variant [7]. In [10], the attacks were stabilized by incorporating momentum in the gradient computation. Other popular attacks include DeepFool [8], which iteratively linearizes the classifier to compute minimal perturbations sufficient for the sample to cross the decision boundary, and other computationally more expensive attacks, such as CW [9], JSMA [6], and others [20, 21, 22]. As of today, Projected Gradient Descent (PGD) [11], which utilizes the local first-order network information to compute a maximum loss increment within a specified ℓ_∞ norm-bound, is generally considered as the most effective attack strategy.

Despite a significant research effort in devising defense mechanisms against adversarial attacks [23, 24, 25, 26], it was shown in [27] that most such defenses can easily be breached in the white-box setting, where the attacker knows the network architecture. The main exception to this rule is adversarial training [11], where the model is trained jointly with clean images and their adversarial counterparts. Many variants of adversarial training were thus proposed, such as ensemble adversarial training [18] to soften the classifier's decision boundaries, ALP [12] to minimize the difference between the logit activations of real and adversarial images, the use of additional feature denoising blocks [13], of metric learning [28, 29], and of regularizers to penalize changes in the model's prediction w.r.t. the input perturbations [30, 31]. Nevertheless, PGD-based adversarial training remains the method of choice, thanks to its robustness and generalizability to unseen attacks [5, 7, 8, 9, 32].

Unfortunately, adversarial training is computationally expensive. This was tackled in [33] by recycling the gradients computed to update the model parameters so as to reduce the overhead of generating adversarial examples, albeit not remove this overhead entirely. More recently, [34] showed that combining the single-step FGSM with random initialization is almost as effective as PGD-based training, but at a significantly lower cost. Unlike all of the adversarial training strategies, our approach does not require computing adversarial images, and does not depend on a specific attack scheme. Instead, it aims to ensure a maximal separation between the different classes in high attention regions. This significantly differs from [35, 36], which clusters the penultimate layer's global representation, without focusing on discriminative regions and without attempting to separate these features. Furthermore, and more importantly, in contrast to all the abovementioned methods, our approach is tailored to fine-grained recognition, making use of the representations that have proven effective in this field, such as Bags of Words (BoW) [37, 38] and VLAD [39, 40], which have the advantage over second-order features [41, 42, 43] of providing some degree of interpretability.

Interpretability. Understanding the decisions of a DNN is highly important in real-world applications to build user trust. In the context of general image recognition, the trend of interpreting a DNN's decision was initiated by [44], followed by the popular CAMs [16]. Subsequently, variants of CAMs [45, 46] and other visualization strategies [47] were proposed.

Here, in contrast to these works, we focus on the task of fine-grained recognition. In this domain, BoW-inspired representations, such as the one of [48, 49, 50, 15, 51], were shown to provide some degree of interpretability. While most methods [48, 49, 50, 51] allows one to highlight the image regions important for classification, it does not provide one with visual explanations of the network's decisions. This is addressed by ProtoPNet [15], which extracts class-specific prototypes. However, the feature embedding learnt by ProtoPNet gives equal importance to all image regions, resulting in a large number of prototypes representing non-discriminative background regions, as illustrated by Fig. 1. Here, we overcome this by designing an attention-aware system that learns prototypes which are close to high-attention regions in feature space, while constraining the non-discriminative regions from all classes to be close to each other. Furthermore, we show that this brings about not only interpretability, but also robustness to adversarial attacks, which has never been studied in the context of fine-grained recognition.

3 Interpreting Adversarial Attacks

Before delving into our method, let us study in more detail the experiment depicted by Fig. 1 to understand the decision of a fine-grained recognition CNN under adversarial attack. For this analysis, we experiment with two networks: the second-order attentional pooling network of [14] and the ProtoPNet of [15], both of which inherently encode some notion of interpretability in their architecture, thus not requiring any post-processing. Specifically, [14] uses class attention maps to compute class probabilities, whereas [15] exploits the similarity between image regions and class-specific prototypes. We analyze the reasons for the success of adversarial attacks on four images from the *Black-footed albatross* class.

As shown in Fig 1(a), under attack, [14] misclassifies all four images to *Layman albatross*. Note that these two classes belong to the same general *Albatross* family, and, for clean samples, the region with the highest attention for these two classes is the bird's beak. Because the discriminative regions for these two classes correspond to the same beak region, which looks similar in both classes, the attack becomes easier as minimal perturbation is needed to change the class label.

In the case of ProtoPNet [15], while the network also consistently misclassifies the attacked images, the resulting label differs across the different images, as shown in Fig. 1 (b). In the top row, the situation is similar to that occurring with the method of [14]. By contrast, in the second row, the region activated in the input image corresponds to a different semantic part (wing) than that activated in the prototype (beak). Finally, in the last two rows, the network activates a background prototype that is common across the other categories and thus more vulnerable to attacks.

In essence, the mistakes observed in Fig 1 come from either the discriminative regions of two different classes being too close in feature space, or the use of non-discriminative regions for classification. This motivates us to encourage the feature representation of discriminative regions from different classes to be maximally separated from each other, while minimizing the influence of background regions by making use of attention and by encouraging the features in these regions to lie close to each other so as *not* to be discriminative. This will complicate the attacker's task, by preventing their ability to leverage non-discriminative regions and forcing them to make larger changes in feature space to affect the prediction.

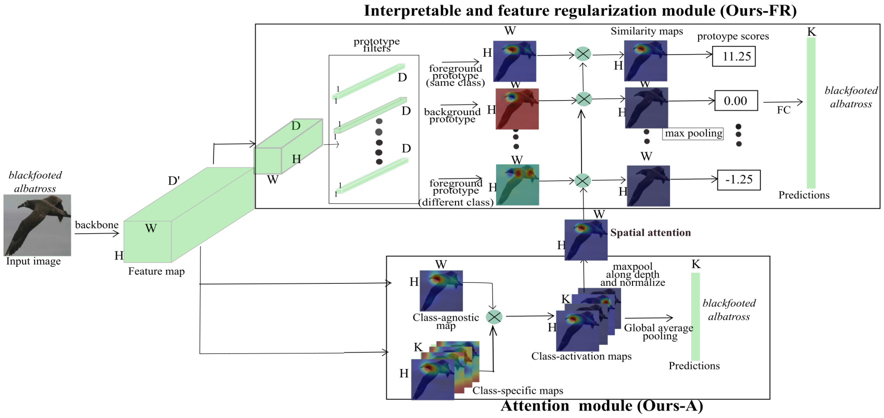


Fig. 3: **Overview of our framework**. Our approach consists of two modules acting on the features extracted by a backbone network. The attention module extracts attention maps that help the network to focus on the discriminative image regions. The feature regularization module further uses the attention maps to encourage separating the learned prototypes belonging to different classes.

4 Method

In this section, we introduce our approach to increasing the robustness of fine-grained recognition by maximal separation of class-specific discriminative regions. Figure 3 gives an overview our framework, which consists of two modules post feature extraction: (i) An attention module that learns class-specific filters focusing on the discriminative regions; and (ii) a feature regularization module that maximally separates the class-specific features deemed discriminative by the attention module. Through the feature regularization module, we achieve the dual objective of providing interpretability and increasing the robustness of the backbone network to adversarial attacks.

Note that, at inference time, we can either use the entire framework for prediction, or treat the attention module, together with the backbone feature extractor, as an standalone network. As will be demonstrated by our experiments, both strategies yield robustness to adversarial attacks, which evidences that our approach in fact robustifies the final feature map. Below, we first describe the overall architecture of our framework and then discuss the feature regularization module in more detail.

4.1 Architecture

Formally, let $\mathbf{I}_i$ denote an input image, and $\mathbf{X}_i \in \mathbb{R}^{H\times W\times D'}$ represent the corresponding feature map extracted by a fully-convolutional backbone network. Our architecture is inspired by the ProtoPNet of [15], in the sense that we also rely on class-specific prototypes. However, as shown in Section 3, ProtoPNet fails to learn discriminative prototypes, because it allows the prototypes to encode non-discriminative background information and to be close in feature space even if they belong to different classes. To address this, we propose to focus on the important regions via an attention mechanism and to regularize the prototypes during training.

Specifically, our attention module consists of two sets of filters: (i) A class-agnostic $1 \times 1 \times D'$ filter yielding a single-channel map of size $H \times W$; and (ii) K class-specific $1 \times 1 \times D'$ filters producing K maps of size $H \times W$ corresponding to the K classes in the dataset. Each of the class-specific map is then multiplied by the class-agnostic one, and the result is spatially averaged to generate a K-dimensional output. As shown in [14], this multiplication of two attention maps is equivalent to a rank-1 approximation of second-order pooling, which has proven to be well-suited to aggregate local features for fine-grained recognition.

The second branch of our network extracts interpretable prototypes and is responsible to increase the robustness of the features extracted by the backbone. To this end, $\mathbf{X}_i$ is first processed by two 1×1 convolutional layers to decrease the channel dimension to D. The resulting representation is then passed through a prototype layer that contains m learnable prototypes of size $1 \times 1 \times D$, resulting in m similarity maps of size $H \times W$. Specifically, the prototype layer computes the residual vector $\mathbf{r}$ between each local feature and each prototype, and passes this distance through an activation function defined as $f(\mathbf{r}) = \log\big((\|\mathbf{r}\|_2^2 + 1)/(\|\mathbf{r}\|_2^2 + \gamma)\big)$, where γ is set to $1e-5$. In contrast to [15], to focus on discriminative regions, we modulate the resulting similarity maps with an attention map $\mathbf{A}_i$, computed by max-pooling the final class-specific maps of the attention module. We then spatially max-pool the resulting attention-aware similarity maps to obtain similarity scores, which are passed through the final classification layer to yield class probabilities. As in [15], we make the prototypes class specific by splitting the m prototypes into K sets of c prototypes and initializing the weights of the classification layer of the prototype branch to $+1$ for positive connections between prototype and class label and -0.5 for negative ones. While exploiting attention encourages the prototypes to focus on the discriminative regions, nothing explicitly prevents prototypes from different classes to remain close in feature space, thus yielding a small margin between different classes and making the classifier vulnerable to attacks. This is what we address below.

4.2 Discriminative Feature Separation

To learn a robust feature representation, we introduce two feature regularization losses that aim to maximally separate the prototypes of different classes. Let $\mathbf{x}_i^t$ represent a local feature vector at location t in feature map $\mathbf{X}_i$ from image $\mathbf{I}_i$ with label y_i. Furthermore, let $N = W \cdot H$ be the total number of feature vectors in $\mathbf{X}_i$, and $\mathbf{P}_{y_i}$ be the set of prototypes belonging to class y_i.

Our regularization consists of two attention-aware losses, a clustering one and a separation one. The attentional-clustering loss pulls the high-attention regions in a sample close to the nearest prototype of its own class. We express this as

$$\mathrm{L}_{clst}^{att}(\mathbf{I}_i) = \sum_{t=1}^{N} a_i^t \min_{l:\mathbf{p}_l \in \mathbf{P}_{y_i}} \|\mathbf{x}_i^t - \mathbf{p}_l\|_2^2 \ , \tag{1}$$

where a_i^t is the attention value at location t in $\mathbf{A}_i$. By contrast, the attentional-separation loss pushes the high-attention regions away from the nearest prototype of any other class. We compute it as

$$\mathrm{L}_{sep}^{att}(\mathbf{I}_i) = -\sum_{t=1}^{N} a_i^t \min_{l:\mathbf{p}_l \notin \mathbf{P}_{y_i}} \|\mathbf{x}_i^t - \mathbf{p}_l\|_2^2 \ . \tag{2}$$

While these two loss functions encourage the prototypes to focus on high-attention, discriminative regions, they leave the low-attention regions free to be close to any prototype, thus increasing the vulnerability of the network to attacks. We therefore further need to push the non-discriminative regions away from such informative prototypes. A seemingly natural way to achieve this would consist of exploiting inverted attention maps, such as $1 - \mathbf{a}_t$ or $1/\mathbf{a}_t$. However, in practice, we observed this to make training unstable. Instead, we therefore propose to make use of the attention maps from *other* samples to compute the loss for sample i. Specifically, we re-write our regularization loss for sample i as

$$\mathrm{L}_{reg}(\mathbf{I}_i) = \sum_{j=1}^{B} \sum_{t=1}^{N} \lambda_1 a_j^t \min_{l:\mathbf{p}_l \in \mathbf{P}_{y_i}} \|\mathbf{x}_i^t - \mathbf{p}_l\|_2^2 - \lambda_2 a_j^t \min_{l:\mathbf{p}_l \notin \mathbf{P}_{y_i}} \|\mathbf{x}_i^t - \mathbf{p}_l\|_2^2 \ , \tag{3}$$

where B is the number of samples in the mini-batch. When $j = i$, we recover the two loss terms defined in Eqs. 1 and 2. By contrast, when $j \neq i$, we exploit the attention map of a different sample. The intuition behind this is that either the attention map of sample j focuses on the same regions as that of sample i, and thus the loss serves the same purpose as when using the attention of sample i, or it focuses on other regions, and the loss then pushes the corresponding feature map, encoding a low-attention region according to the attention map of sample i, to its own prototype in class y_i. In practice, we have observed this procedure to typically yield a single background prototype per class. These background prototypes inherently become irrelevant for classification because they correspond to low-attention regions and have thus a similarity score close to zero, thanks to our attention-modulated similarity maps. As such, we have empirically found that all background prototypes tend to cluster.

Ultimately, we write our total loss function for sample i as

$$\mathrm{L}(\mathbf{I}_i) = \mathrm{CE}_{att}(\mathbf{I}_i) + \mathrm{CE}_{reg}(\mathbf{I}_i) + \mathrm{L}_{reg}(\mathbf{I}_i) \ ,$$

where CE_{att} and CE_{reg} represent the cross-entropy loss of the attention module and the feature regularization module, respectively.

At inference time, we perform adversarial attacks on the joint system by exploiting the cross-entropy loss of both the attention and feature regularization module. Furthermore, we also attack the attention module on its own, showing that, together with the feature extraction backbone, it can be used as a standalone network and also inherits robustness from our training strategy.

5 Experiments

5.1 Experimental Setting

Datasets. We experiment on two popular fine-grained datasets, Caltech UCSD Birds (CUB) [52] and Stanford Cars-196 [53].
Threat Model. We consider both white-box and black-box attacks under an ℓ_∞-norm budget. We evaluate robustness for two attack tolerances $\epsilon = \{2/255, 8/255\}$. In addition to the popular 10-step PGD attack [11], we test our framework with FGSM [5], BIM [7], and MI [10] attacks. For PGD attacks, we set the step size α to 1/255 for $\epsilon = 2/255$ and to 2/255 for $\epsilon = 8/255$. For the other attacks, we set number of iterations to 10 and the step size α to ϵ divided by the number of iterations, as in [35, 28]. For black-box attacks, we transfer the adversarial examples generated using 10-step PGD with $\epsilon = 8/255$ and $\alpha = 2/255$ on either a similar VGG16 [54] architecture, or a completely different DenseNet-121 [55] architecture. We denote by BB-V and BB-D the black-box attacks transferred from VGG16 and DenseNet-121, respectively.
Networks. We evaluate our approach using 3 backbone networks: VGG-16 [54], VGG-19 [54] and ResNet-34 [56]. Similarly to [15], we perform all experiments on images cropped according to the bounding boxes provided with the dataset, and resize the resulting images to 224 × 224. For both VGG-16 and VGG-19, we use the convolutional layers until the 4th block to output 7×7 spatial maps of 512 channels. For ResNet-34, we take the network excluding the final global average pooling layer as backbone. We initialize the backbone networks with weights pretrained on ImageNet [2].
Evaluated Methods. As baselines, we use the attentional pooling network (AP) of [14], and the state-of-the-art ProtoPNet of [15]. We use **Ours-FR** and **Ours-A** to denote the output of our feature regularization module and of our attention module, respectively. In other words, AP and Ours-A share the *same* architecture at inference time, and Ours-FR is an attention-aware variant of ProtoPNet. To further boost the performance of the baselines and of our approach, we perform adversarial training. Specifically, we generate adversarial examples using the recent fast adversarial training strategy of [34], which relies on a single step FGSM with random initialization. During training, we set ϵ to 8/255 and α to $\{0.5\epsilon, 1.25\epsilon\}$ as suggested in [34]. This was shown in [34] to perform on par with PGD-based adversarial training, while being computationally much less expensive. For our approach, during fast adversarial training, we use the cross-entropy loss of both modules to generate the adversarial images. We denote by AP* and ProtoPNet* the adversarially-trained AP and ProtoPNet baselines, respectively, and by **Ours-FR*** and **Ours-A*** the adversarially-trained counterparts of our two sub-networks. We also compare with state-of-the-art defense [35] which regularizes the hidden space with additional prototype conformity loss (PCL). Lastly, we also evaluate triplet-loss [57] based feature separation of penultimate layer global features as another baseline. Due to the space limitation, we provide more experimental training details in the supplementary material.

Base Network	Attacks (Steps,ϵ)	Clean (0,0)	FGSM (1,2)	FGSM (1,8)	BIM (10,2)	BIM (10,8)	PGD (10,2)	PGD (10,8)	MIM (10,2)	MIM (10,8)	BB-V (10,2)	BB-D (10,8)
VGG-16	AP [14]	78.0%	36.5%	31.0%	27.7%	14.6%	23.5%	11.7%	30.2%	16.7%	9.6%	60.4%
	AP+ Triplet [57]	**81.0%**	**49.5%**	36.6%	33.5%	11.2%	26.5%	8.50%	37.7%	14.3%	8.54%	63.4%
	AP+ PCL [35]	80.0%	41.0%	33.1%	32.9%	13.6%	23.5%	9.6%	35.3%	17.1%	10.6%	65.8%
	Ours-A	80.4%	47.2%	**40.2%**	**40.0%**	**23.2%**	**35.3%**	**21.8%**	**42.2%**	**26.4%**	**12.9%**	**66.9%**
	ProtoPNet [15]	69.0%	19.9%	8.10%	3.80%	0.00%	2.20%	0.00%	5.00%	0.10%	**22.9%**	58.5%
	ProtoPNet+Ours	**73.2%**	**49.9%**	**42.2%**	**42.5%**	**35.3%**	**38.4%**	**30.1%**	**42.9%**	**37.5%**	15.4%	**59.7%**
VGG-19	AP [14]	75.7%	20.4%	14.5%	13.4%	6.9%	10.5%	5.7%	14.8%	6.9%	21.1%	61.3%
	AP+ Triplet [57]	**82.0%**	**53.9%**	38.2%	35.0%	12.4%	27.7%	9.40%	39.4%	15.3%	17.40%	64.9%
	AP+ PCL [35]	76.9%	20.3%	14.8%	12.1%	5.7%	8.8%	4.2%	13.9%	6.8%	19.8%	60.2%
	Ours-A	79.7%	51.4%	**44.6%**	**42.3%**	**26.5%**	**36.8%**	**26.3%**	**45.0%**	**42.6%**	**29.8%**	**68.2%**
	ProtoPNet [15]	73.8%	22.9%	11.1%	3.2%	0.0%	1.2%	0.0%	3.6%	0.0%	21.0%	58.0%
	Ours-FR	75.4%	**52.2%**	**46.3%**	**46.6%**	**41.3%**	**42.4%**	**31.0%**	**44.4%**	**37.6%**	**30.4%**	**63.7%**
ResNet-34	AP [14]	**79.9%**	30.4%	26.3%	18.0%	7.20%	13.2%	5.8%	22.3%	8.6%	43.0%	59.4%
	AP+ Triplet [57]	78.6%	25.6%	18.7%	11.4%	2.9%	7.1%	1.8%	14.7%	3.8%	42.11%	58.4%
	AP+ PCL [35]	77.9%	30.1%	24.5%	21.4%	13.3%	17.6%	11.6%	23.9%	15.3%	45.7%	61.4%
	Ours-A	79.0%	**32.3%**	**27.0%**	**24.8%**	**20.5%**	**22.5%**	**19.8%**	**26.2%**	**22.0%**	**48.6%**	**63.2%**
	ProtoPNet [15]	75.1%	23.2%	12.8%	7.80%	1.80%	4.10%	1.00%	8.90%	2.20%	39.1%	53.0%
	Ours-FR	**76.3%**	**30.7%**	**22.0%**	**19.3%**	**13.6%**	**14.2%**	**13.0%**	**19.1%**	**13.8%**	**46.0%**	**60.0%**

Table 1: Classification accuracy of different networks with ℓ_∞ based attacks on CUB200. The best result of each column and each backbone is shown in bold. The last two columns correspond to black-box attacks with PGD attack.

5.2 Results on CUB 200

Quantitative Analysis. We first compare the accuracy of our method to that of the baselines with the three backbone networks on CUB200. Table 1 and Table 2 provide the results for vanilla and fast adversarial training, respectively. On the clean samples, **Ours-FR** typically surpasses its non-attentional counterpart **ProtoPNet** [15], and **Ours-A** yields the better accuracy than baseline AP and AP+PCL across all backbones. This is true both without (Table 1) and with (Table 2) adversarial training.

Under adversarial attack, our approach, without and with adversarial training, yields better robustness under almost all attacks and backbones. Importantly, the boost in performance is larger for attacks with larger perturbations. Furthermore, our model trained with clean samples sometimes outperform even the adversarially-trained baselines. For example, on VGG-16 with PGD attack with $\epsilon = 8/255$, **AP*** yields an accuracy of 16.9% (Table 2) while **Ours-A** reaches 21.8% accuracy (Table 1). This evidences the ability of our feature regularization module to learn robust features, even without seeing any adversarial examples. This is further supported by the fact that, despite **AP** and **Ours-A** having the same architecture at inference, **Ours-A** is more robust to attacks. Our method outperforms AP+PCL in most cases since PCL do not take into account subtle difference in local regions and regularizes global representation only. Furthermore, in contrast to adversarially-trained models, our vanilla approach does not trade off clean accuracy for robustness. For example, on VGG-16, while adversarial training made the clean accuracy of **AP*** drop to 54.9% (Table 2), that of **Ours-A** is 80.4% (Table 1). In other words, we achieve good robustness *and* clean accuracy.

Base Network	Attacks (Steps,ϵ)	Clean (0,0)	FGSM (1,2)	FGSM (1,8)	BIM (10,2)	BIM (10,8)	PGD (10,2)	PGD (10,8)	MIM (10,2)	MIM (10,8)	BB-V (10,2)	BB-D (10,8)
VGG-16	AP* [14]	54.9%	44.9%	24.2%	41.9%	18.2%	41.2%	16.9%	41.9%	18.7%	54.6%	54.0%
	AP+PCL* [35]	60.7%	50.5%	28.5%	47.1%	22.8%	46.7%	21.6%	47.2%	23.5%	59.5%	59.9%
	Ours-A*	**63.1%**	**56.1%**	**34.8%**	**51.7%**	**29.6%**	**50.8%**	**28.0%**	**52.0%**	**32.5%**	**66.3%**	**68.0%**
	ProtoPNet* [15]	60.1%	44.5%	26.9%	**57.1%**	10.9%	35.9%	10.3%	37.6%	13.5%	58.4%	59.1%
	Ours-FR*	**63.0%**	**53.3%**	**37.3%**	49.4%	**30.4%**	**48.1%**	**28.6%**	**49.7%**	**31.1%**	**61.1%**	**62.0%**
VGG-19	AP* [14]	58.0%	47.5%	29.1%	44.3%	25.6%	44.0%	24.34%	44.4%	26.2%	57.0%	57.3%
	AP+PCL* [35]	61.8%	52.1%	30.9%	48.9%	24.7%	48.6%	23.3%	49.1%	25.4%	60.5%	60.9%
	Ours-A*	**68.2%**	**57.1%**	**36.5%**	**53.2%**	**30.4%**	**52.6%**	**29.2%**	**53.5%**	**31.2%**	**66.2%**	**66.9%**
	ProtoPNet* [15]	55.1%	40.0%	28.9%	26.5%	11.3%	29.7%	9.60%	25.6%	10.2%	53.6%	53.9%
	Ours-FR*	**64.4%**	**55.5%**	**37.4%**	**51.2%**	**30.6%**	**50.4%**	**28.7%**	**52.1%**	**32.3%**	**62.5%**	**63.2%**
ResNet-34	AP* [14]	55.6%	47.8%	29.2%	44.80%	21.0%	44.5%	19.4%	44.9%	21.9%	55.3%	55.2%
	AP+PCL* [35]	54.5%	45.4%	26.9%	42.3%	18.2%	41.9%	16.4%	42.4%	19.1%	54.0%	54.0%
	Ours-A*	**62.2%**	**54.2%**	**35.7%**	**51.5%**	**25.5%**	**51.0%**	**23.1%**	**51.6%**	**26.6%**	**61.5%**	**61.9%**
	ProtoPNet* [15]	**57.9%**	46.5%	30.3%	41.1%	21.1%	40.3%	18.4%	41.5%	20.9%	56.9%	57.0%
	Ours-FR*	57.6%	**49.5%**	**32.3%**	**45.8%**	**23.2%**	**44.9%**	**19.9%**	**46.1%**	**24.6%**	**57.1%**	57.0%

Table 2: Classification accuracy of different robust networks with ℓ_∞ based attacks on CUB200. The best result of each column and each backbone is shown in bold. The last two columns correspond to black-box attacks with PGD attack.

Transferability Analysis. To evaluate robustness to black-box attacks, we transfer adversarial examples generated from substitute networks to our framework and to the baselines. As substitute models, we use a VGG-16 [54] and DenseNet-121 [55] backbone followed by global average pooling and a classification layer. The corresponding results are reported in the last two columns of Table 1 and Table 2. As in the white-box case, our approach outperforms the baselines in this black-box setting, thus confirming its effectiveness at learning robust features.

Qualitative Analysis. Let us now qualitatively evidence the benefits of our approach. To this end, in Figure 4, we visualize the 10 class-specific prototypes learned by ProtPNet and by our approach for the *Blackfooted albatross* class. Specifically, we show the activation heatmaps of these prototypes on the source image that they have been projected to. Note that ProtoPNet learns multiple background prototypes, whereas our approach encodes all the background information in a *single non-discriminative* prototype. Furthermore, ProtoPNet [15] focuses on much larger regions, which can be expected to be less discriminative than the fine-grained regions obtained using our approach. This is due to our use of attention, which helps the prototypes to focus on the areas that are important for classification.

In Figure 5, we analyze the effect of adversarial attacks on AP, ProtoPNet and our approach (all without adversarial training) by visualizing the attention maps and/or a few top activated prototypes along with their similarity scores for a *Blackfooted albatross* image with and without attack. Without attack, AP activates a larger region than our attention module. Furthermore, ProtoPNet activates a prototype from a different class (*Cape glossay starling*), while our approach focuses on the correct class only. This already shows that the features

Network	Att-clustering loss	Att-separation loss	Clean (0,0)	PGD (10,8)
AP [14]	-	-	78.0%	11.7%
Ours-A	-	-	78.7%	14.07%
	-	✓	79.6%	0.0%
	✓	-	80.0%	19.3%
	✓	✓	80.4%	21.8%

Network	Att-clustering loss	Att-separation loss	Clean (0,0)	PGD (10,8)
ProtoPNet [15]	-	-	69.0%	0.0%
Ours-FR	-	-	75.7%	13.76%
	-	✓	69.8%	0.0%
	✓	-	73.7%	18.7%
	✓	✓	73.2%	30.1%

Table 3: **Ablation study**. Contribution of each proposed feature regularization module in classification accuracy of undefended VGG-16 network.

learned by these baselines are less discriminative, making them more vulnerable to adversarial attacks. As a matter of fact, under attack, AP focuses on a different region that is not discriminative for the *Blackfooted albatross* class. Similarly, ProtoPNet activates prototypes of different classes with high similarity scores, highlighting non-discriminative regions. By contrast, the prototypes activated by our approach remain the same as in the clean case, thus corresponding to the correct class.

Gradient Obfuscation. As suggested in [27], we check the gradient obfuscation to ensure that proposed approach do not give false sense of security. As shown in Figure 6, VGG-16 trained with our feature regularization performance drops as the perturbation norm increases. Further, from Table 1 and 4, the black-box attacks are less successful than white box attacks. Both these experiments suggest ou formulation do not suffer from gradient obfuscation.

Adversarial sample detection. Our formulation also helps in detecting adversarial samples due to well separation of discriminative regions. Following [58], we learn a logistic detector by computing mahalanobis distance to the nearest class-conditional Gaussian distribution as the feature at every layer of the network. As shown in Figure 7, our proposed feature regularization approach increases the detection AUROC performance over the baselines by around 20%.

Ablation Study. To understand the importance of each module in achieving robustness, we perform ablation study on VGG-16. As shown from Table 3, our attention-aware formulation performs better than baseline even without feature

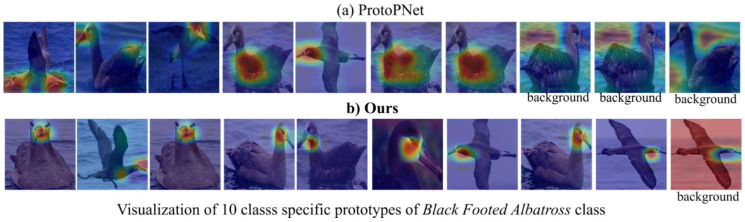

(a) ProtoPNet
background
background
background
b) Ours
background
Visualization of 10 classs specific prototypes of *Black Footed Albatross* class

Fig. 4: **Comparison of the prototypes learned with ProtoPNet [15] and with our approach on CUB**. ProtoPNet yields multiple background prototypes and prototypes that focus on large regions. By contrast, our prototypes are finer-grained and thus more representative of the specific class in the images.

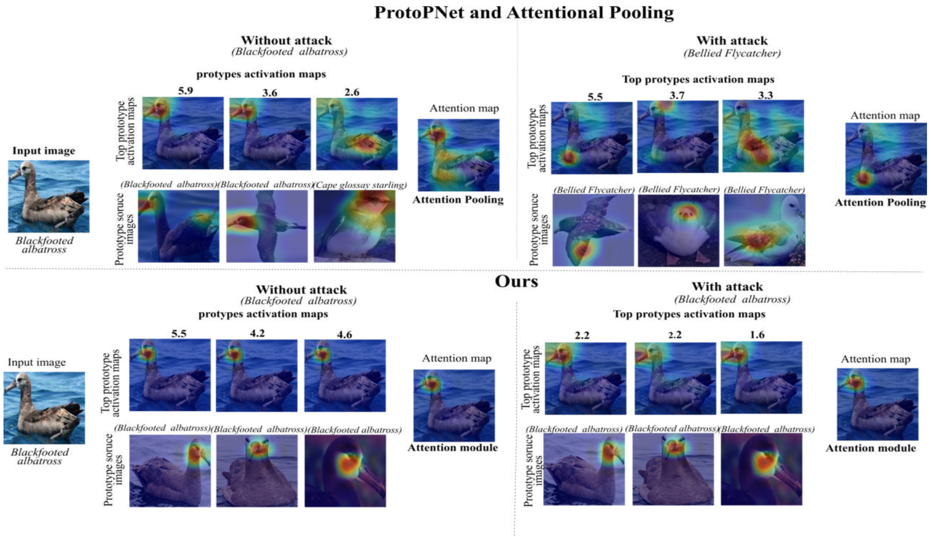


Fig. 5: **Comparison of the activated image regions without and with attack.** Without attack, the baselines (AP and ProtoPNet) tend to rely on relatively large regions, sometimes corresponding to wrong classes, for prediction. By contrast, our approach focuses more closely on the discriminative regions. Under PGD attack, this phenomenon is further increased, with ProtoPNet and AP activating incorrect prototypes and regions. The activations obtained with our approach remain similar to those obtained without attacks, albeit with a decrease in the similarity scores, indicated above the top prototype activation maps.

regularization. However by adding attentional cluster and separation cost, we achieve significant improvements over the baselines.

Base Nettwork	Attacks (Steps,ϵ)	Clean (0,0)	FGSM (1,2)	FGSM (1,8)	BIM (10,2)	BIM (10,8)	PGD (10,2)	PGD (10,8)	MIM (10,2)	MIM (10,8)	BB-V (10,2)	BB-D (10,8)
VGG-16	AP [14]	**91.2%**	52.6%	40.2%	37.4%	10.5%	28.8%	6.93%	41.7%	12.9%	12.5%	82.5%
	AP+Triplet [57]	91.1%	54.3%	**43.5%**	42.4%	14.9%	34.1%	9.54%	45.5%	19.2%	15.6%	**84.7%**
	AP+PCL [35]	90.2%	51.7%	40.5%	39.3%	14.1%	31.8%	9.44%	42.5%	17.5%	16.7%	83.9%
	Ours-A	88.5%	**58.7%**	40.2%	**48.0%**	**28.6%**	**46.5%**	**21.7%**	**53.2%**	**33.2%**	**19.9%**	82.2%
	ProtoPNet [15]	**84.5%**	31.2%	9.85%	4.78%	0.01%	2.23%	0.00%	6.5%	0.01%	**27.8%**	**75.5%**
	Ours-FR	83.8%	**60.1%**	**52.0%**	**51.3%**	**41.0%**	**47.8%**	**32.9%**	**51.8%**	**43.9%**	23.4%	75.1%
VGG-19	AP	**91.5%**	50.1%	37.8%	33.4%	10.3%	23.83%	6.93%	37.9%	12.7%	20.7%	82.8%
	AP+Triplet [57]	91.0%	56.2%	45.1%	40.5%	13.0%	30.3%	8.70%	45.3%	16.7%	29.0%	85.0%
	AP+PCL [35]	91.3%	61.3%	49.9%	49.0%	19.7%	40.2%	14.1%	52.4%	23.4%	30.6%	**85.7%**
	Ours-A	88.7%	**64.4%**	**54.8%**	**56.4%**	**36.7%**	**51.7%**	**33.4%**	**58.1%**	**41.0%**	**35.9%**	82.5%
	ProtoPNet [15]	**85.6%**	34.1%	20.8%	11.3%	1.11%	4.40%	0.5%	14.2%	1.39%	26.5%	75.5%
	Ours-FR	85.0%	**62.4%**	**54.7%**	**54.5%**	**45.7%**	**51.2%**	**38.5%**	**54.3%**	**47.6%**	**36.1%**	**76.8%**

Table 4: Classification accuracy of different undefended networks with ℓ_∞ based attacks on Cars196.

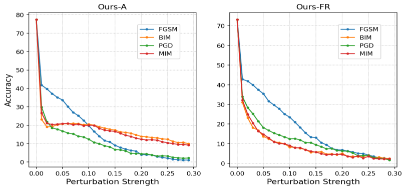


Fig. 6: Performance of VGG-16 with our proposed approach under different perturbation strengths.

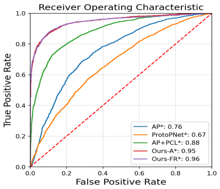


Fig. 7: ROC curves for adversarial sample detection on robust VGG-16 with PGD attack.

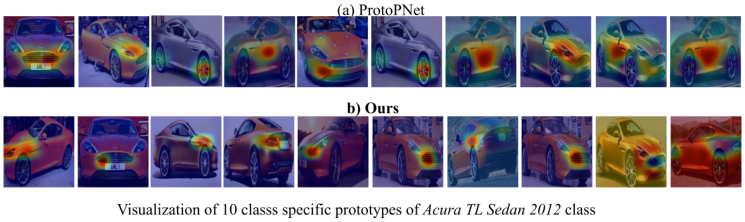

(a) ProtoPNet
b) Ours
Visualization of 10 classs specific prototypes of *Acura TL Sedan 2012* class

Fig. 8: **Comparison of the prototypes learned with ProtoPNet [15] and with our approach on Stanford-Cars**. ProtoPNet yields prototypes that cover large regions, whereas our prototypes more focused.

5.3 Results on Stanford Cars

We now present on results on Stanford Cars [53]. In Table 4, we report the results obtained using vanilla training. As in the CUB case, our approach yields better robustness than the baselines. We provide a qualitative analysis and the results obtained with adversarial training in the supplementary material.

In Figure 8, we compare the prototypes learned with ProtoPNet and with our approach for the *Accura TL Sedan* class. As before, while the prototypes learned by ProtoPNet cover large regions, those obtained with our framework are more focused on the discriminative parts of the car.

6 Conclusion

In this paper, we have performed the first study of adversarial attacks for fine-grained recognition. Our analysis has highlighted the key factor for the success of adversarial attacks in this context. This has inspired us to design an attention- and prototype-based framework that explicitly encourages the prototypes to focus on the discriminative image regions. Our experiments have evidenced the benefits of our approach, able to match and sometimes even outperform adversarial training, despite not requiring seeing adversarial examples during training and further improving AUROC score of adversarial sample detection.

Acknowledgments. This work was funded in part by the Swiss National Science Foundation.

References

1. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3431–3440
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
3. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2014) 1725–1732
4. Zhang, C., Li, H., Wang, X., Yang, X.: Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 833–841
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
6. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: 2016 IEEE European symposium on security and privacy (EuroS&P), IEEE (2016) 372–387
7. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533 (2016)
8. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 427–436
9. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp), IEEE (2017) 39–57
10. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 9185–9193
11. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
12. Kannan, H., Kurakin, A., Goodfellow, I.: Adversarial logit pairing. arXiv preprint arXiv:1803.06373 (2018)
13. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 501–509
14. Girdhar, R., Ramanan, D.: Attentional pooling for action recognition. In: Advances in Neural Information Processing Systems. (2017) 34–45
15. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: deep learning for interpretable image recognition. In: Advances in Neural Information Processing Systems. (2019) 8928–8939
16. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2921–2929
17. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. arXiv preprint arXiv:1805.12152 (2018)
18. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017)

19. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
20. Xu, K., Liu, S., Zhao, P., Chen, P.Y., Zhang, H., Fan, Q., Erdogmus, D., Wang, Y., Lin, X.: Structured adversarial attack: Towards general implementation and better interpretability. arXiv preprint arXiv:1808.01664 (2018)
21. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation **23** (2019) 828–841
22. Narodytska, N., Kasiviswanathan, S.: Simple black-box adversarial attacks on deep neural networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE (2017) 1310–1318
23. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: 2016 IEEE Symposium on Security and Privacy (SP), IEEE (2016) 582–597
24. Samangouei, P., Kabkab, M., Chellappa, R.: Defense-gan: Protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605 (2018)
25. Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.: Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991 (2017)
26. Song, Y., Kim, T., Nowozin, S., Ermon, S., Kushman, N.: Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. arXiv preprint arXiv:1710.10766 (2017)
27. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. arXiv preprint arXiv:1802.00420 (2018)
28. Mao, C., Zhong, Z., Yang, J., Vondrick, C., Ray, B.: Metric learning for adversarial robustness. In: Advances in Neural Information Processing Systems. (2019) 478–489
29. Ding, G.W., Sharma, Y., Lui, K.Y.C., Huang, R.: Mma training: Direct input space margin maximization through adversarial training. In: International Conference on Learning Representations. (2019)
30. Ross, A.S., Doshi-Velez, F.: Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Thirty-second AAAI conference on artificial intelligence. (2018)
31. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573 (2019)
32. Tramer, F., Carlini, N., Brendel, W., Madry, A.: On adaptive attacks to adversarial example defenses. arXiv preprint arXiv:2002.08347 (2020)
33. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! In: Advances in Neural Information Processing Systems. (2019) 3353–3364
34. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. arXiv preprint arXiv:2001.03994 (2020)
35. Mustafa, A., Khan, S., Hayat, M., Goecke, R., Shen, J., Shao, L.: Adversarial defense by restricting the hidden space of deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 3385–3394
36. Mustafa, A., Khan, S.H., Hayat, M., Goecke, R., Shen, J., Shao, L.: Deeply supervised discriminative learning for adversarial defense. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)

37. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE (2010) 3304–3311
38. Wang, Z., Li, H., Ouyang, W., Wang, X.: Learnable histogram: Statistical context features for deep neural networks. In: European Conference on Computer Vision, Springer (2016) 246–262
39. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 5297–5307
40. Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., Russell, B.: Actionvlad: Learning spatio-temporal aggregation for action classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 971–980
41. Yu, K., Salzmann, M.: Statistically-motivated second-order pooling. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 600–616
42. Kong, S., Fowlkes, C.: Low-rank bilinear pooling for fine-grained classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 365–374
43. Gao, Y., Beijbom, O., Zhang, N., Darrell, T.: Compact bilinear pooling. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 317–326
44. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision, Springer (2014) 818–833
45. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. (2017) 618–626
46. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE (2018) 839–847
47. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Advances in neural information processing systems. (2016) 3387–3395
48. Nakka, K.K., Salzmann, M.: Deep attentional structured representation learning for visual recognition. arXiv preprint arXiv:1805.05389 (2018)
49. Wei, X.S., Xie, C.W., Wu, J., Shen, C.: Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization. Pattern Recognition **76** (2018) 704–714
50. Fu, J., Zheng, H., Mei, T.: Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 4438–4446
51. Wang, Y., Morariu, V.I., Davis, L.S.: Learning a discriminative filter bank within a cnn for fine-grained recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 4148–4157
52. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. (2011)
53. Krause, J., Stark, M., Deng, J., Fei-Fei, L.: 3d object representations for fine-grained categorization. In: Proceedings of the IEEE international conference on computer vision workshops. (2013) 554–561
54. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

55. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 4700–4708
56. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
57. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 815–823
58. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In: Advances in Neural Information Processing Systems. (2018) 7167–7177