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

Supplementary Material

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

1 Training Details

We implemented our approach using the PyTorch library, and ran our experiments on a single 32GB Tesla GPU. We set the mini-batch size B to 75 during training. Similarly to [3], we initialize the last layer of the prototype branch with +1 for positive and -0.5 for negative connection between prototype and class label. We set $c = \{5, 10\}$ prototypes per class, λ_1 to $\{10, 100\}$ and $\lambda_2 = 0.08$ depending on the dataset and architecture. We first fine-tune the attention and feature regularization modules, except for the classification layer of the latter, for 5 epochs with a learning rate of 0.0003, keeping the backbone network fixed. We then jointly train all the layers, except the feature regularization classifier, to minimize the objective of Eq. 3 for 25 epochs, with an initial learning rate of 0.003 and a decay rate of 0.1 applied every 10 epochs. After 30 epochs, we project the prototypes to the nearest training image patch of the same class and optimize the classification layer of the feature regularization module for 15 epochs. We use Adam [4] with the default momentum values for all our experiments. For the adversarial detection experiments, we initially remove the misclassified samples from the test set. We then consider successfully attacked from this subset and train a logistic detector with 20% of data and report results on remaining data.

2 Qualitative Results on CUB

In this section, we provide additional qualitative results on CUB200. In particular, we visualize the learned prototypes, and analyze the classification results by computing the similarity of the samples with the learned prototypes.

Visualization of the learned prototypes. In Figure 1, we show the activation heat maps of the prototypes on the source images to which they were projected for our VGG-16 model. Our method yields fine-grained prototypes that either focus on a small discriminative region or activate the complete non-discriminative

region.

Nearest samples of the learned prototypes. In Figure 2, we show the prototypes and their nearest training images for CUB 200 with VGG-16. Similarly, in Figure 3, we show the prototypes and their nearest test images for CUB 200 with VGG-16. In most cases, the discriminative prototypes activate the same semantic part in all images corresponding to the same class.

Nearest prototypes for a clean image. In Figure 4, we show, for a given clean test image, the top few highest activated prototypes with VGG-16. We observe that the most activated prototypes focus on salient and discriminative regions, with no influence from the background regions.

Nearest prototypes for an adversarial image. In Figure 5, we show the top few highest activated codewords for unsuccessful adversarial samples that retain the predicted label even after the attack. Note that, under attack, the similarity scores of the top activated prototypes decrease, but, thanks to the large separation between the prototypes, the discriminative features do not cross over to other prototypes.

3 Results on Stanford Cars

In Table 1, we report the robustness of fast adversarial training [5] with our discriminative feature separation approach. Our approach, **Ours-FR***, performs better than the baseline ProtoPNet* [3] in all cases. Note that, for multi-step iterative attacks, **Ours-A*** performs better than AP*, while they achieve comparable performance for single-step attacks.

Base Network	Attacks Steps,ϵ)	Clean	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 (10,8)	BB-D (10,8)
VGG-16	AP* [6]	86.2%	**81.1%**	**63.6%**	**78.9%**	53.8%	**78.7%**	50.8%	**78.7%**	55.1%	85.1%	85.9%
	AP+PCL* [7]	**87.4%**	80.5%	59.4%	77.6%	48.5%	77.2%	44.9%	77.9%	50.2%	**86.0%**	**87.1%**
	Ours-A*	84.8%	79.8%	63.3%	77.0%	**54.6%**	76.6%	**51.1%**	77.1%	**55.8%**	84.5%	85.6%
	ProtoPNet* [3]	64.4%	53.7%	31.9%	48.9%	16.5%	48.2%	13.4%	49.2%	18.2%	63.8%	64.2%
	Ours-FR*	**83.7%**	**76.37%**	**62.8%**	**73.5%**	**55.0%**	**72.6%**	**51.9%**	73.8%	**55.4%**	**80.8%**	**82.0%**
VGG-19	AP* [6]	**88.2%**	82.4%	63.4%	79.9%	54.2%	79..6%	50.7%	80.0%	55.7%	86.9%	88.0%
	AP+PCL* [7]	**88.2%**	**82.7%**	64.6%	**80.2%**	57.4%	**79.6%**	54.3%	**80.3%**	58.5%	**87.2%**	**88.1%**
	Ours-A*	87.3%	80.29%	**67.1%**	78.4%	**60.15%**	78.2%	**58.2%**	78.6%	**61.3%**	86.5%	87.3%
	ProtoPNet* [3]	30.0%	19.9%	15.7%	15.0%	16.3%	9.1%	3.00%	3.32%	2.28%	29.4%	29.7%
	Ours-FR*	**84.6%**	**79.6%**	**66.9%**	**77.7%**	**58.6%**	**76.5%**	**55.6%**	**77.8%**	**59.1%**	**83.7%**	**84.5%**

Table 1. Classification accuracy of different robust networks with ℓ_∞ based attacks on Cars196. The best result of each column and each backbone is shown in **bold**. The last two columns correspond to black-box attacks.

Visualization of the learned prototypes. In Figure 6, we show the activation heat maps of the prototypes on the source images to which they were projected for our VGG-16 model. Our method yields fine-grained prototypes that either focus on a small discriminative region or activate the complete non-discriminative

region.

Nearest samples of the learned prototypes. In Figure 7, we show the prototypes and their nearest training images for Cars 196 with VGG-16. Similarly, in Figure 8, we show the prototypes and their nearest test images for Cars 196 with VGG-16. In most cases, the discriminative prototypes activate the same semantic part in all images corresponding to the same class.

Nearest prototypes for an adversarial image. In Figure 9, we show the top few highest activated codewords for unsuccessful adversarial samples that retain the predicted label even after the attack. Note that, under attack, the similarity scores of the top activated prototypes decrease, but, thanks to the large separation between the prototypes, the discriminative features do not cross over to other prototypes.

References

1. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. (2011)
2. 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
3. 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
4. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
5. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. arXiv preprint arXiv:2001.03994 (2020)
6. Girdhar, R., Ramanan, D.: Attentional pooling for action recognition. In: Advances in Neural Information Processing Systems. (2017) 34–45
7. 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

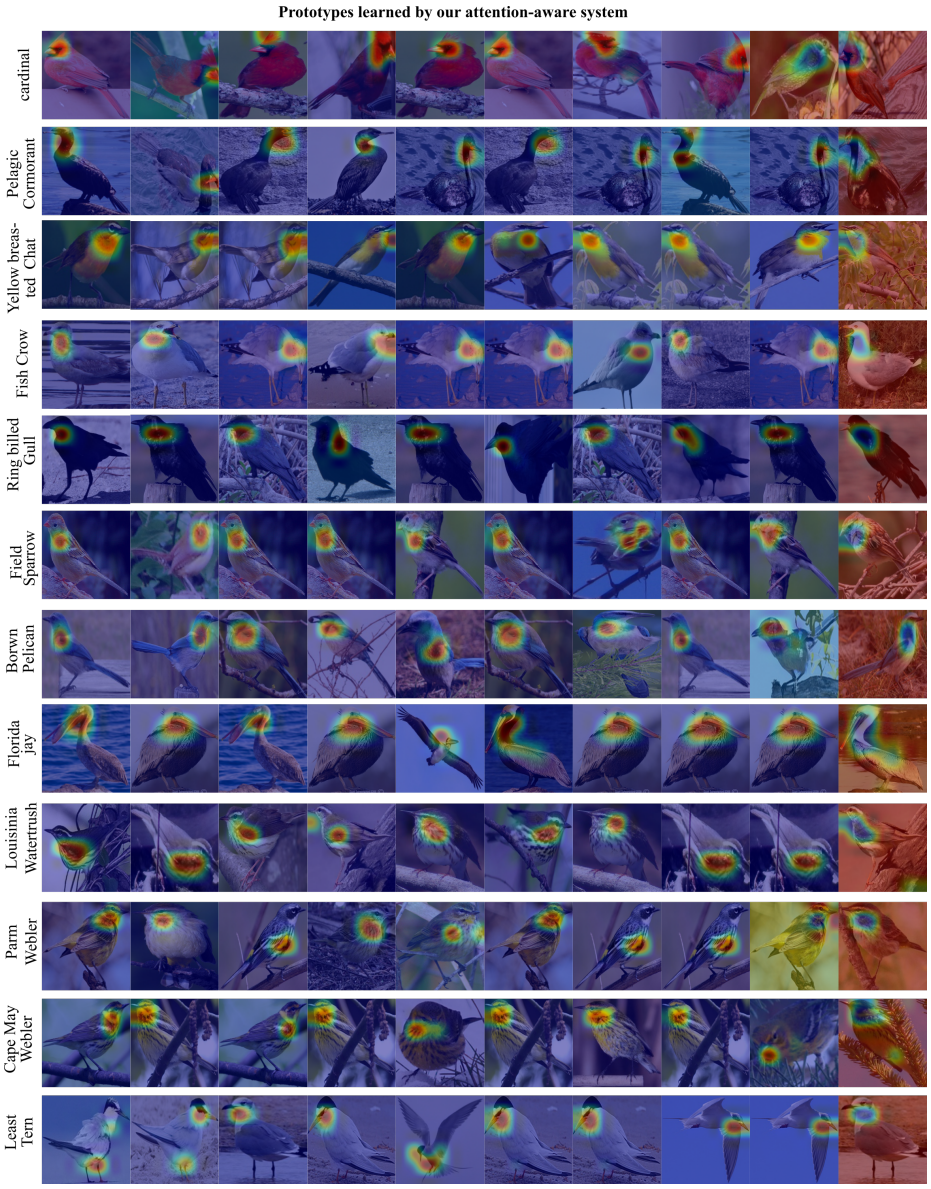

Fig. 1. Visualization of the prototypes learned with our approach on CUB. Our formulation yields prototypes that are fine-grained and representative of the specific class in the images.

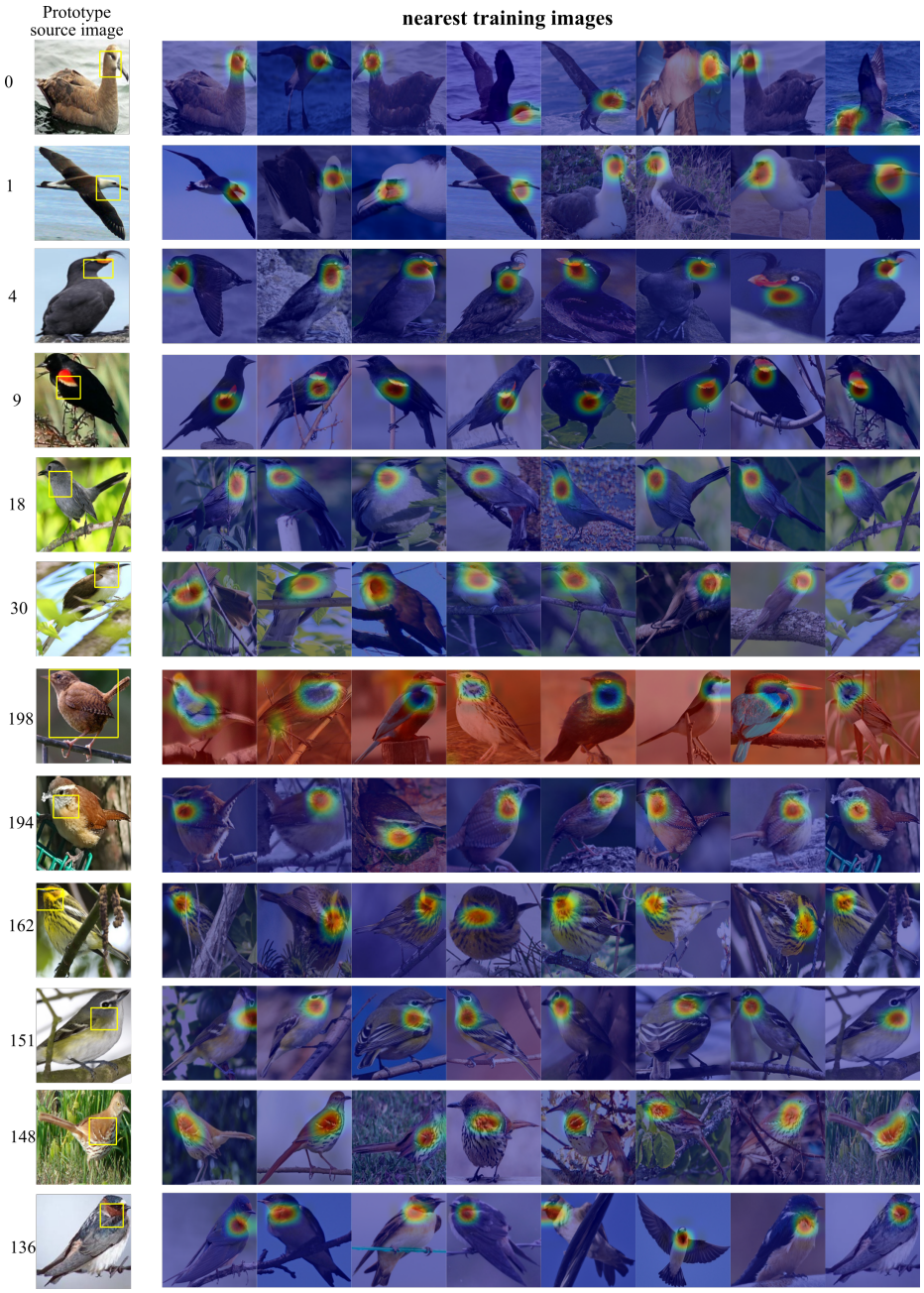

Fig. 2. Visualization of the nearest *train* samples for each learned prototypes with our approach on CUB with VGG-16. All prototypes activate semantically meaningful parts and mostly from the images corresponding to their own label.

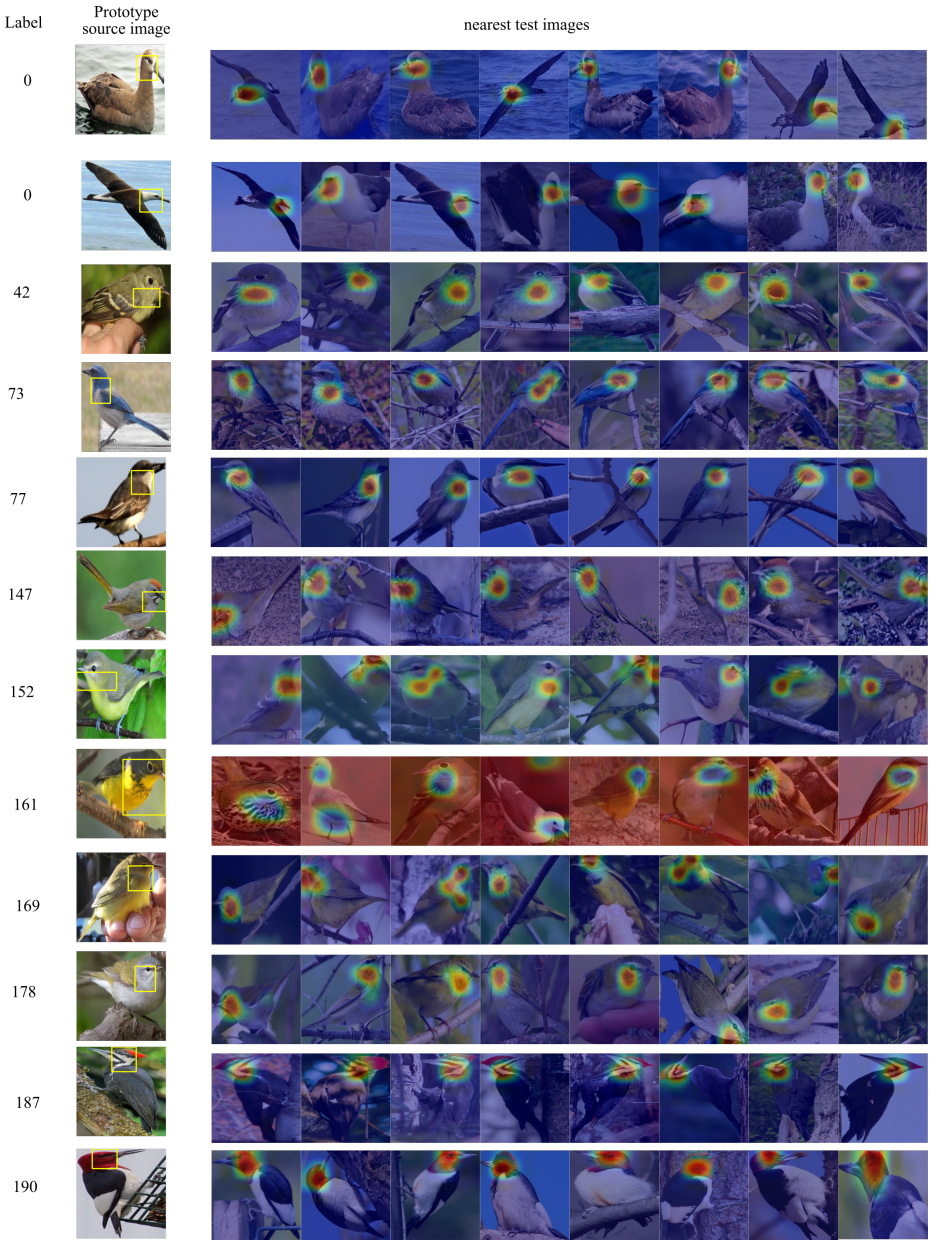

Fig. 3. Visualization of the nearest *test* samples for each learned prototypes with our approach on CUB with VGG-16. All prototypes activate semantically meaningful parts and mostly from the images corresponding to their own label.

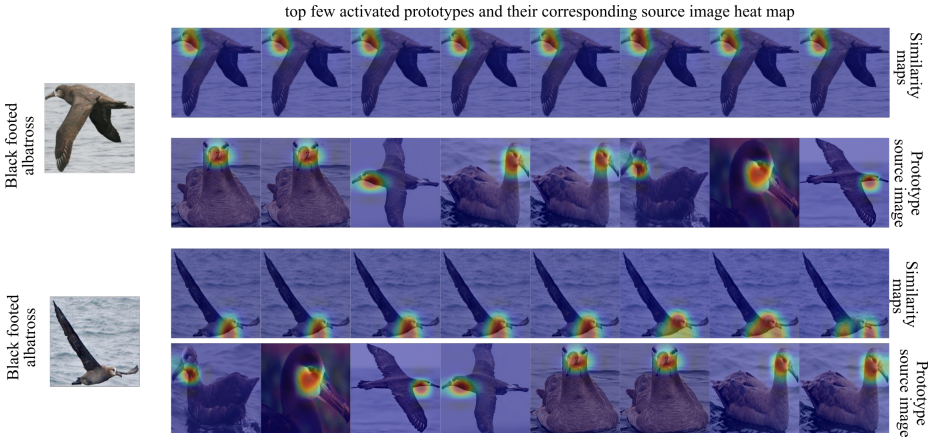

Fig. 4. Visualization of the top activated prototypes for a given test image on CUB. The top prototypes activate semantically meaningful regions and discard the background areas.

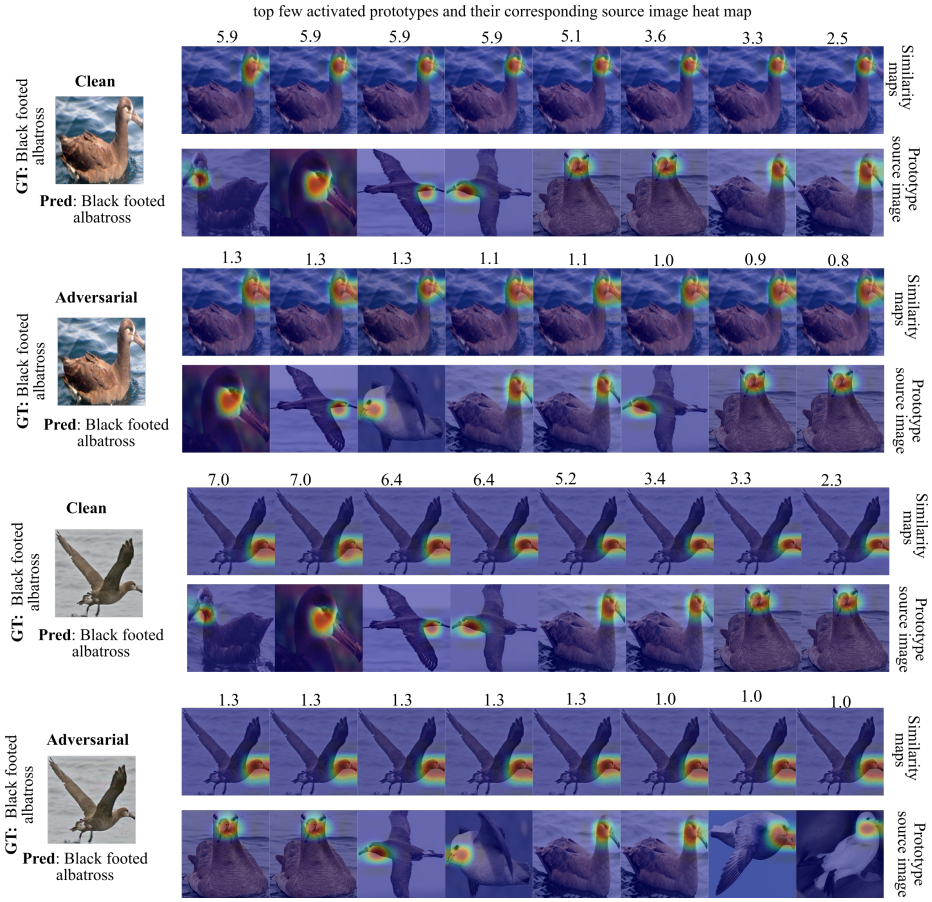

Fig. 5. Visualization of the top activated prototypes for a given clean and adversarial image pair from the CUB test data. The top prototypes corresponds to the true label, even after attack. Moreover, we observe that the similarity score for each prototype decreases, but the attack remains unsuccessful thanks to the large separation between the discriminative prototypes.

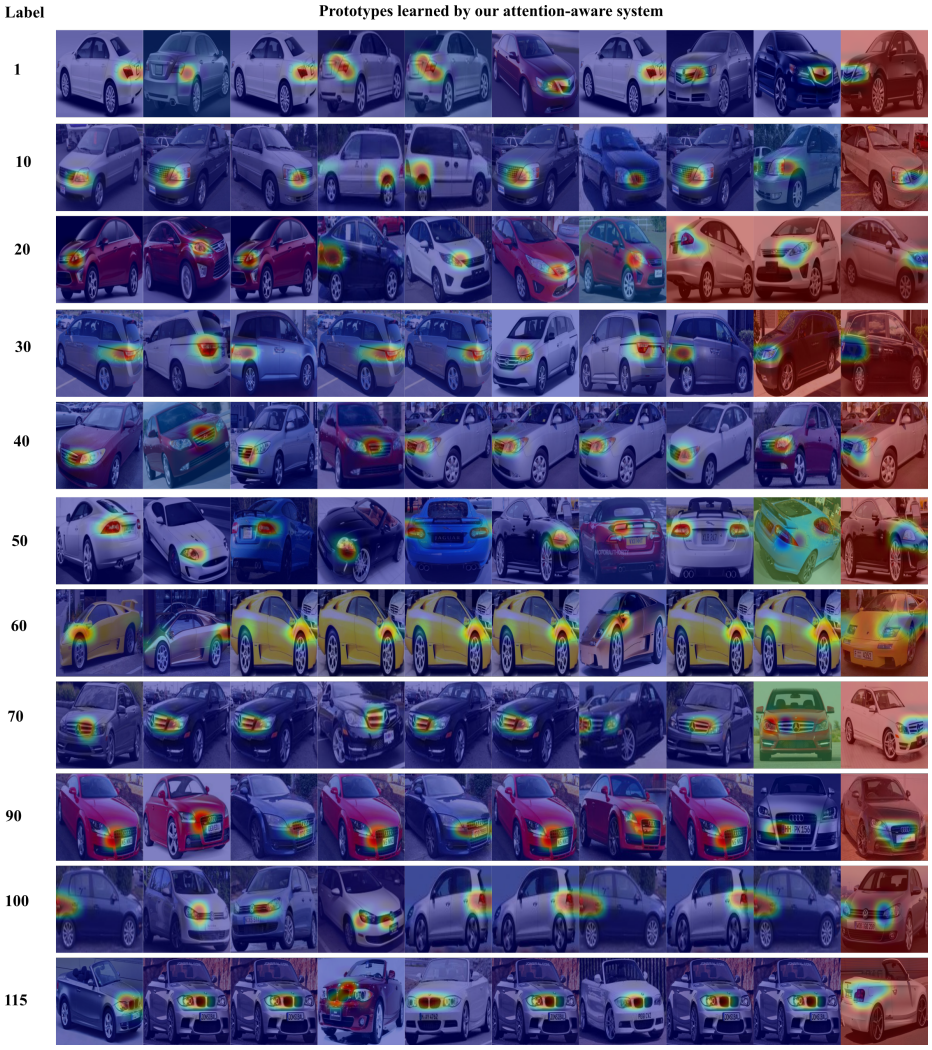

Fig. 6. Visualization of the prototypes learned with our approach on Cars-196. Our formulation yields prototypes that are fine-grained and representative of the specific class in the images.

Fig. 7. Visualization of the nearest *train* samples for each learned prototypes with our approach on Cars-196 with VGG-16. All prototypes activate semantically meaningful parts and mostly from the images corresponding to their own label.

Fig. 8. Visualization of the nearest *test* samples for each learned prototypes with our approach on Cars 196 with VGG-16. All prototypes activate semantically meaningful parts and mostly from the images corresponding to their own label.

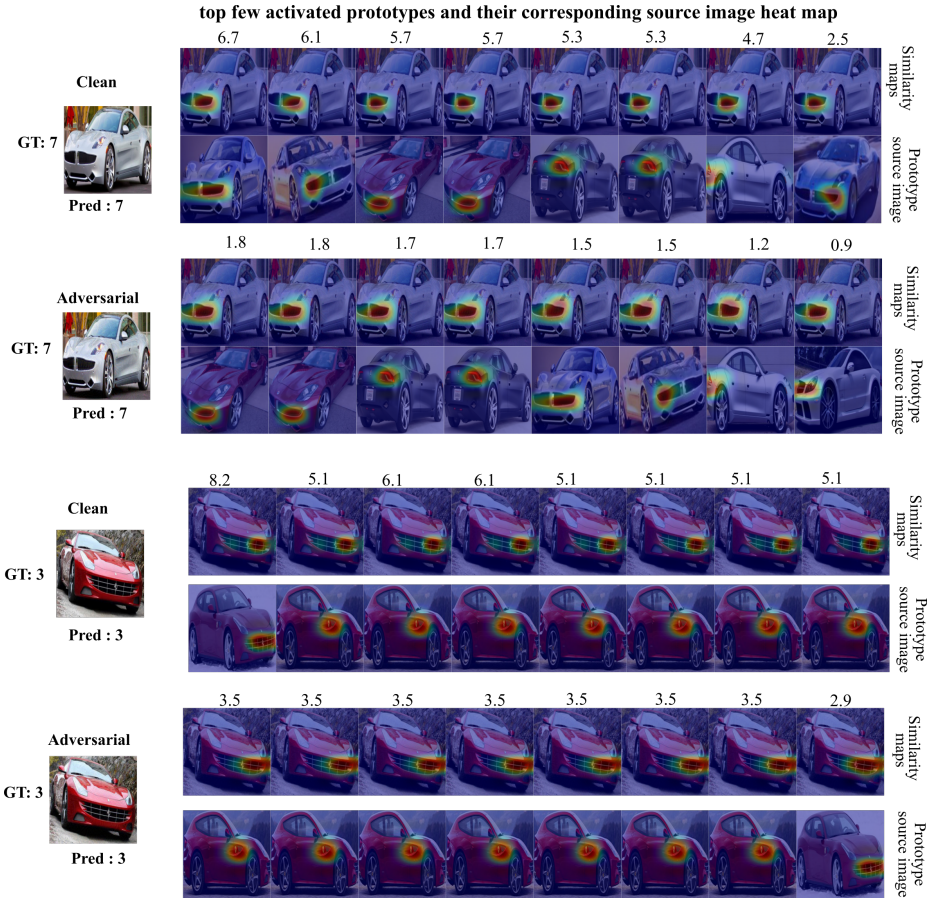

Fig. 9. Visualization of the top activated prototypes for a given clean and adversarial image pair from the Cars-196 test data. The top prototypes corresponds to the true label, even after attack. Moreover, we observe that the similarity score for each prototype decreases, but the attack remains unsuccessful thanks to the large separation between the discriminative prototypes.