Adversarial Framing for Image and Video Classification

Konrad Żołna*[1,2], Michał Zając*[1,3], Negar Rostamzadeh[2], Pedro O. Pinheiro[2]
[1]Jagiellonian University, Kraków, Poland
[2]Element AI, Montréal, Canada
[3]Nomagic, Warsaw, Poland
konrad.zolna@gmail.com, emzajac@gmail.com, negar@elementai.com, pedro@opinheiro.com

Abstract

Neural networks are prone to adversarial attacks. In general, such attacks deteriorate the quality of the input by either slightly modifying most of its pixels, or by occluding it with a patch. In this paper, we propose a method that keeps the image unchanged and only adds an *adversarial framing* on the border of the image. We show empirically that our method is able to successfully attack state-of-the-art methods on both image and video classification problems. Notably, the proposed method results in a universal attack which is very fast at test time. Source code can be found at `github.com/zajaczajac/adv_framing`.

Introduction

The remarkable success of deep convolutional networks for image and video classification (Karpathy et al. 2014; Krizhevsky, Sutskever, and Hinton 2012) has spurred interest in analyzing their robustness. Unfortunately, it turned out that even though neural networks often achieve human level performance (Taigman et al. 2014), they are susceptible to adversarial attacks (Szegedy et al. 2014). It means that the output of a neural network-based classifier may be drastically changed by applying a small perturbation to its input. We divide such perturbations into two categories: *fully-affecting* and *partially-affecting*.

- Fully-affecting attacks generate small pixel intensity modifications which are optimized to be hardly visible for humans. These attacks typically have their ℓ_2 or ℓ_∞ norm constrained (Carlini and Wagner 2017; Moosavi-Dezfooli, Fawzi, and Frossard 2016) and hence affect the whole image.
- Partially-affecting attacks usually have their ℓ_0 norm constrained. They introduce perceptible but small occlusion to the image, such as a patch (Brown et al. 2017; Karmon, Zoran, and Goldberg 2018) or a single pixel (Su, Vargas, and Sakurai 2017).

The attacks mentioned above either slightly modify all the pixels of the image or occlude parts of it. However, the attackers may find this to be a serious limitation and seek for new types of attacks. For instance, consider a scenario where they upload videos containing forbidden content. Their goal is to bypass video-sharing website's filters. At the same time, the perturbations introduced should not be distracting and all information should be retained.

In this paper, a new attack which is well-suited for the above-mentioned purposes is demonstrated. The method, dubbed adversarial framing (**AF**), consists in simply adding a thin border around the original input (which may be an image or a video), keeping the whole content unchanged (see Figure 1 and `youtu.be/PrU9R6eFNTs` for some qualitative results). The attack is universal (Moosavi-Dezfooli et al. 2017), which means the same **AF** is applied to all inputs. The method only requires substantial computing during the training procedure. At test time, the only extra computation required is the appending of the precomputed framing to the input.

Similarly to (Athalye, Carlini, and Wagner 2018), we believe that research on attack techniques deepens understanding of inner workings of neural networks. We hope that our work and analyzing adversarial attacks in general can be helpful in designing defenses and/or robust methods.

In this work, we consider a white-box setting, in which an access to the architecture and weights of the trained classifier is given. Previous work has shown that if only black-box access is given, a surrogate model can be leveraged to obtain an attack that transfers well to the original model (Papernot, McDaniel, and Goodfellow 2016). Therefore, a white-box model is a realistic assumption and, in fact, is the most commonly considered paradigm in the literature.

Method

Computing the adversarial framing

Suppose a labeled dataset of images or videos $\mathcal{D}$ is given. Moreover, a differentiable classifier f has been trained so that for each input x and class c, a probability $f_c(x)$ is assigned to x of being in class c.

We now present a procedure to train the adversarial framing to attack f. During training, a minibatch is sampled from $\mathcal{D}$. Every example is surrounded with the same framing, which is the current version of the trained **AF**. In case of videos, every frame of each example is surrounded with the same framing. Then the classification loss is backpropagated and the framing is modified using its gradients to maximize the loss. The training continues until convergence. The fram-

*Equal contribution

correct: Boston bull
unattacked: Boston bull
attacked: maypole

correct: ocarina
unattacked: loupe
attacked: maypole

correct: tusker
unattacked: tusker
attacked: maypole

correct: gas pump
unattacked: gas pump
attacked: maypole

correct: Egyptian cat
unattacked: tabby
attacked: maypole

Figure 1: Examples from ImageNet with adversarial framing of width 3. Most of the images are wrongly classified as a maypole. We hypothesize that the colorfulness of that class makes it especially easy for **AF** to resemble it.

Algorithm 1 Training of the adversarial framing

input: Dataset $\mathcal{D} = \{(x_i, y_i)\}$, $x_i \in [0,1]^{h \times w \times 3}$, classifier f, framing's width W
output: Universal adversarial framing θ
Initialize $\hat{\theta} \sim \mathcal{N}(0,1)$, of size $2W(h + w + 2W)$
repeat
  for each datapoint $(x_i, y_i) \in \mathcal{D}$ **do**
    $\hat{x}_i \leftarrow x_i$ surrounded by $\theta := Sigmoid(\hat{\theta})$
  end for
  update $\hat{\theta}$ to minimize $\frac{1}{|\mathcal{D}|} \sum_i \log(f_{y_i}(\hat{x}_i))$
until convergence

ing's width W is a tunable hyperparameter fixed at the beginning of the training procedure.

For a detailed explanation see Algorithm 1. The algorithm is presented for image datasets. The modification for video datasets is straightforward.

Note that the input size is modified due to the addition of the framing. This does not pose any issue to the CNN-based classifier, as most modern architectures (such as ResNet (He et al. 2016) or ResNeXt (Xie et al. 2017)) accept various input sizes. If the classifier's input size is fixed, the proposed algorithm can be simply modified so that the image is resized before applying adversarial framing. We investigate performance under various resizing strategies further in the paper.

Experiments

Untargeted attacks

We performed untargeted attacks against state-of-the-art classifiers for ImageNet (Russakovsky et al. 2015) and UCF101 (Soomro, Zamir, and Shah 2012) datasets. We compare our **AF** to two simple baselines. They both do not require any training and are fixed. One applies uniformly distributed random noise (**RF**) and another black pixels only (**BF**).

ImageNet is a large-scale image dataset containing over million images from 1000 various classes. It serves as a popular benchmark for image classification. We performed attacks against ResNet-50 (He et al. 2016) model pretrained on ImageNet. The model was taken from PyTorch Model Zoo (Paszke et al. 2017). Results are reported in Table 1a.

UCF101 is a dataset containing realistic videos. Each video contains a person performing some action, out of 101 possible classes. We tested our method by performing an attack on a ResNeXt-101 based spatio-temporal 3D CNN – we used model pretrained by (Hara, Kataoka, and Satoh 2018). This model takes clips as input, each containing 16 consecutive frames. Results are reported in Table 1b.

Targeted attacks

As it can be seen in Figure 1 and Figure 2, adversarial framing usually fools the classifier into wrongly recognizing one particular class. In the case of ImageNet, this adversarial class is usually maypole – even across different trainings. We hypothesize that this is because of colorfulness of this object.

In order to make sure that the performance of our attack does not depend on presence of such special classes, we performed attacks in targeted setting. In these experiments, instead of minimizing the output score for the ground-truth class, we maximize the score for a randomly selected target class. We report success rate (*i.e.* percentage of images classified as a given target) for different target classes.

Results for ImageNet dataset are in Table 2a and for UCF101 dataset are in Table 2b.

Training details

In all the experiments we used Adam optimizer (Kingma and Ba 2014). For all the hyperparameters of the optimizer except for learning rate, we used default values from PyTorch (Paszke et al. 2017) implementation.

On ImageNet (Russakovsky et al. 2015) we trained for 5 epochs, with initial learning rate 0.1 decaying by 0.1 every 2 epochs and batch size 32. On UCF101 (Soomro, Zamir, and Shah 2012) we trained for 60 epochs, with initial learning rate 0.03 decaying by 0.3 every 15 epochs and batch size 32. On both these datasets, we trained adversarial framing using training data only. All the reported results were computed on validation data.

On ImageNet, we applied the framing to images previously resized to 224×224. On UCF101, we applied the

Attack	$W=1$	$W=2$	$W=3$	$W=4$
None	76.13%			
RF	70.13%	67.63%	68.36%	67.25%
BF	72.99%	72.9%	72.39%	72.34%
AF	10.53%	0.44%	0.11%	0.1%

(a) ImageNet dataset

Attack	$W=1$	$W=2$	$W=3$	$W=4$
None	85.95%			
RF	82.57%	80.53%	81.11%	79.74%
BF	84.94%	84.73%	84.75%	84.59%
AF	65.77%	22.12%	9.45%	2.05%

(b) UCF101 dataset

Table 1: Accuracies of the classifiers (full validation set) for various values of the framing width W.

	min	avg	max
AF, $W=4$	99.15%	99.66%	99.98%

(a) ImageNet dataset

	min	avg	max
AF, $W=4$	73.04%	89.63%	99.78%

(b) UCF101 dataset

Table 2: Success rate of targeted attacks (the higher the better) with adversarial framing of width 4. Minimum, average and maximum values are taken across 8 different targets.

framing to images previously resized to 112×112. These are standard input dimensions for aforementioned datasets.

Further analyses

Saliency visualization

Grad-CAM (Selvaraju et al. 2017) is a method for producing visual explanations for a convolutional neural network's predictions. For a given classifier f, input x and a class c, it computes a heatmap visualizing how much particular regions of x contribute to a score of the class c output by f.

We computed such visualizations for the pretrained ResNet-50 from PyTorch Model Zoo, taking as input images from ImageNet. We consider both the cases with and without an adversarial framing. Few qualitative results are presented in Figure 2[1].

Classifier's input resizing

Our method does not modify pixels of the original input (with dimensions $h \times w$) and only adds a framing around it. This results in dimensions of the classifier input becoming $(h + 2W) \times (w + 2W)$ where W is framing's width. This is fine for most of the state-of-the-art image classification architectures; however, to make sure the approach also works for classifiers with fixed input size, we conducted experiments with attacking the ImageNet classifier for several image resizing strategies:

(a) no resizing, input dimensions are changed (**Vanilla**). The framing is trained with Algorithm 1.

(b) first the framing is added, and then the whole image is rescaled back to $h \times w$ (**Frame & Resize, F&R**). We use the same framing as in (a).

(c) the image is first scaled to $(h - 2W) \times (h - 2W)$ and then the framing is added, so that size is again $h \times w$ (**Resize & Frame, R&F**). We train the framing separately because the number of parameters is smaller than in (a).

(d) framing is put on the original image, occluding its border pixels; the size remains unchanged (**Occlude**). We use the same framing as in (c).

While we see differences in results, all the variants prove very efficient for $W = 4$. Compared to other resizing strategies, performance is especially degraded in **Frame & Resize**. This is expected since the adversarial framing itself is resized and mixed with neighbouring pixels there.

Based on these results, if one can change input dimensions, **Vanilla** approach performs the best, and otherwise **Resize & Frame** leads to the highest error rate. Results are shown in Table 3.

Related work

Universal partially-affecting attacks

Since existing attacks are quite different from our approach, it is hard to perform a direct comparison. However, we try to compare our work with universal partially-affecting attacks using localized patches. We are aware of two works that perform these kind of attacks, LaVAN (Karmon, Zoran, and Goldberg 2018) and Adversarial patch (Brown et al. 2017). Both methods were tested on ImageNet and hence we will focus on that case.

Unfortunately, each of these works consider different percentages of the image pixels that may be altered. We thus first recall our results for various framing sizes and then relate it to results from other works. With **AF** of width 1, we use less than 2% of the image's pixels and accuracy in untargeted setting drops to 10.53%. For $W = 2$, we use less than 3.5% of the image's pixels and the accuracy is 0.44% only. Finally, for $W = 4$ we use less than 7% of the image's pixels to make the classifier almost completely confused (0.1% accuracy) in untargeted setting and achieve 99.66% average success rate in targeted setting.

[1]We use the following Grad-CAM implementation: `github.com/kazuto1011/grad-cam-pytorch`.

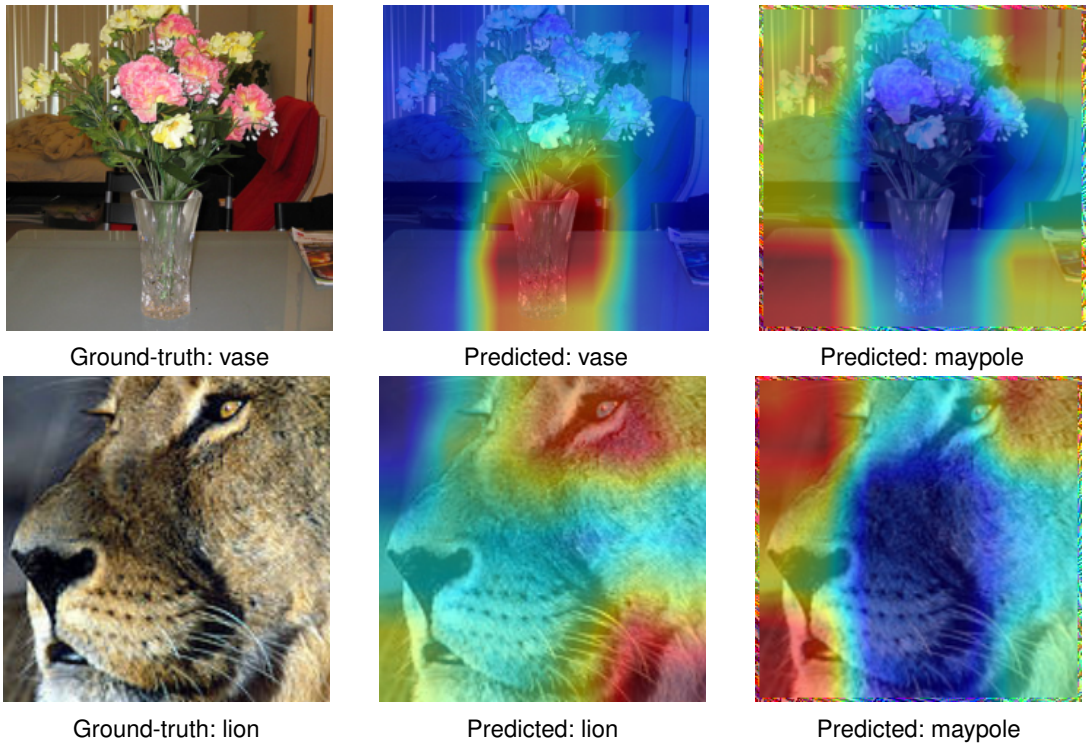

Figure 2: Grad-CAM for inputs from ImageNet. For each example, first the original image is shown, and then the visualizations for unattacked and attacked image. While the network correctly identifies key objects for classification in unattacked images, it concentrates on the image borders when given adversarial input.

In LaVAN, a patch occluding about 2% of the image is used (which is comparable to our **AF** of width 1). Their universal attack has success rate 74.1% in targeted setting. When they use the same patch to measure untargeted performance, they change the output class of the classifier for only 78.9% of data, which suggests that the accuracy of the classifier is higher than 10.53% achieved by our method.

Adversarial patch is a method that creates localized perturbations which can be deployed in a real world. The authors consider targeted setting only. They measure success rate as a function of percentage of pixels used. They need to occlude at least 10% of pixels to obtain 90% success rate.

As mentioned before, the comparison to prior works is burdensome due to the differences in shape, localization and design of other approaches. However, when we put all these characteristics aside and focus on the performance in respect to the ratio of perturbed pixels to the original ones, it seems that our method performs better than prior approaches. Additionally, our method is shown to generalize to videos.

Attacking video classifiers

Although extensive literature exists on attacks against image classifiers, we are aware of only a few works on video classifier attacks (Wei, Zhu, and Su 2018; Li et al. 2018; Rey-de Castro and Rabitz 2018). While resulting in successful attacks, these approaches are fully-affecting and hence introduce adversarial artifacts in the video. In contrast, output from our attack contains the original video and no information is lost. Moreover, the framing is constant over all video frames, removing any "flickering" effect that could potentially be distracting to viewers.

Attack	$W = 1$	$W = 2$	$W = 3$	$W = 4$
None	76.13%			
Vanilla	**10.53%**	**0.44%**	**0.11%**	**0.1%**
F&R	56.12%	20%	5.32%	1.19%
R&F	33.87%	1.09%	0.15%	**0.1%**
Occlude	43.78%	3.2%	0.33%	0.12%

Table 3: Accuracies of ImageNet classifier attacked using adversarial framing with different resizing strategies.

Conclusion

In this work, we present a simple method for attacking both image and video classifiers. The proposed attack is universal (*i.e.* the same adversarial framing can be applied in different images or videos), efficient and effective. Moreover, our method does not modify the original content of the input and only adds a small border to surround it.

Acknowledgments

Michał Zając is co-financed by National Centre for Research and Development as a part of EU supported Smart Growth

Operational Programme 2014-2020 (POIR.01.01.01-00-0392/17-00). Konrad Żołna is financially supported by National Science Centre, Poland (2017/27/N/ST6/00828, 2018/28/T/ST6/00211).

We modified the repository provided by (Zolna, Geras, and Cho 2019) to implement our method (see `github.com/kondiz/casme` for their code).

This is an extended version of the paper published at 33rd AAAI Conference on Artificial Intelligence (see `doi.org/10.1609/aaai.v33i01.330110077`).

References

[Athalye, Carlini, and Wagner 2018] Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*.

[Brown et al. 2017] Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial patch. *CoRR* abs/1712.09665.

[Carlini and Wagner 2017] Carlini, N., and Wagner, D. A. 2017. Towards evaluating the robustness of neural networks. In *Symposium on Security and Privacy*.

[Hara, Kataoka, and Satoh 2018] Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

[Karmon, Zoran, and Goldberg 2018] Karmon, D.; Zoran, D.; and Goldberg, Y. 2018. LaVAN: Localized and visible adversarial noise. In *ICML*.

[Karpathy et al. 2014] Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; and Fei-Fei, L. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*.

[Kingma and Ba 2014] Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

[Krizhevsky, Sutskever, and Hinton 2012] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

[Li et al. 2018] Li, S.; Neupane, A.; Paul, S.; Song, C.; Krishnamurthy, S. V.; Roy-Chowdhury, A. K.; and Swami, A. 2018. Adversarial perturbations against real-time video classification systems. *CoRR* abs/1807.00458.

[Moosavi-Dezfooli et al. 2017] Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. *CVPR*.

[Moosavi-Dezfooli, Fawzi, and Frossard 2016] Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*.

[Papernot, McDaniel, and Goodfellow 2016] Papernot, N.; McDaniel, P. D.; and Goodfellow, I. J. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR* abs/1605.07277.

[Paszke et al. 2017] Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

[Rey-de Castro and Rabitz 2018] Rey-de Castro, R., and Rabitz, H. 2018. Targeted nonlinear adversarial perturbations in images and videos. *CoRR* abs/1809.00958.

[Russakovsky et al. 2015] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *IJCV*.

[Selvaraju et al. 2017] Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*.

[Soomro, Zamir, and Shah 2012] Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR* abs/1212.0402.

[Su, Vargas, and Sakurai 2017] Su, J.; Vargas, D. V.; and Sakurai, K. 2017. One pixel attack for fooling deep neural networks. *CoRR* abs/1710.08864.

[Szegedy et al. 2014] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR*.

[Taigman et al. 2014] Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*.

[Wei, Zhu, and Su 2018] Wei, X.; Zhu, J.; and Su, H. 2018. Sparse adversarial perturbations for videos. *CoRR* abs/1803.02536.

[Xie et al. 2017] Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *CVPR*.

[Zolna, Geras, and Cho 2019] Zolna, K.; Geras, K. J.; and Cho, K. 2019. Classifier-agnostic saliency map extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 10087–10088.