Perceptual Ad-Block Evasion

Nagharjun M, Rakshana BS, Avinash V
New York University

https://github.com/Nagharjun17/Perceptual_Ad-Blocker_Evasion

Abstract

With the emergence of perceptual ad-blocking, like Sentinel and Percival, advertisers and web publishers are faced with a challenge. Previous work, as showcased in "AdVersarial: Defeating Perceptual Ad Blocking" by Florian Tramèr et al., [1] demonstrates that adversarial examples can defeat these perceptual ad-blockers. The current advances in adversarial attacks and defenses for neural networks call for a thorough exploration in the context of perceptual ad-blocking. Our project aims to dive deeper into the realm of adversarial attacks by exploring more advanced and resilient adversarial attacks that can effectively evade the latest perceptual ad-blockers and proposing potential defense mechanisms that can be integrated into perceptual ad-blockers to resist such adversarial attempts.

1 Problem Formulation

As we approach 2024, the digital advertising market is projected to reach an unprecedented value of over 645 billion USD. This remarkable growth underscores the imperative of effective ad management strategies, not solely for enhancing the browsing experience but also for ensuring the optimal flow of advertising revenue to pertinent sectors. A notable development in this domain was observed in October 2023, when YouTube implemented a stringent policy against the use of ad blockers. Despite these measures, a substantial segment of the internet populace, accounting for over 42%, continues to rely on ad blockers. This reliance poses a significant challenge in the development of effective techniques for bypassing ad detection systems. Our project aims to address this challenge by employing what are known as adversarial attacks. These sophisticated and innovative techniques are designed to outmaneuver and deceive ad blockers, thereby ensuring the visibility and effectiveness of digital ads.

Ad blockers are typically categorized into two distinct types: the widely-used traditional ad blockers and the more advanced perceptual ad blockers that emulate human-like ad detection capabilities. Advanced systems such as Sentinel and Percival, which represent the forefront of ad blocking technology, have demonstrated vulnerabilities as revealed in studies like 'AdVersarial: Defeating Perceptual Ad Blocking'. These findings highlight that even state-of-the-art ad blocking systems are not impervious to evasion. This revelation accentuates the urgent necessity for novel approaches in ad detection that can proficiently circumvent these advanced blockers, thereby maintaining the integrity and efficacy of digital advertising strategies.

2 Dataset and Model Architecture

For our analysis, we have used the dataset from the AdVersarial project's repository. This dataset was meticulously assembled by the authors using object detection models to pinpoint advertisements in web page screenshots. Our dataset is its inclusion of data from the AdVersarial project.It encompasses 206 images designated for training, 23 for validation, and 20 specifically for testing.This dataset is sourced from a broad spectrum of web platforms, including news sites, social media channels, and

e-commerce websites. Notably, Adblock Plus trained the YOLOv3 model on labeled Facebook ad screenshots, while another YOLOv3 was fine-tuned on diverse ad samples from news site screenshots of the G20 countries, initially identified via a web proxy using filter lists. These models serve as the basis for evaluating ad detection and the effectiveness of adversarial tactics as detailed in the study "AdVersarial: Defeating Perceptual Ad Blocking." Each ad in these images comes with detailed labels indicating its precise location and specific attributes.Our dataset is a balanced mix of real ads – the kind we encounter in our everyday web browsing – and synthetic ads, which we've crafted to represent emerging and innovative ad styles. This blend ensures that our model's training isn't confined to past ad formats but is also attuned to future trends in advertising.

Includes techniques like Blur, MedianBlur, ToGray, and CLAHE. These augmentations enhance the model's ability to generalize by presenting a variety of scenarios.

3 Methodology

Data Collection and Preparation: The initial phase involves gathering a diverse array of images, encompassing both real ads commonly found across various websites and artificially created ads. Each image in this collection is meticulously labeled, with precise demarcation of the ads and their exact locations. This labeling serves as a guide for the system to accurately identify and locate ads within these images.

Adopting YOLOv5 for Object Detection: Central to our approach is the implementation of the YOLOv5 model for object detection, a significant advancement from the YOLOv3 model utilized in prior foundational research. YOLOv5 brings enhanced accuracy, improved feature detection, and superior overall performance to our ad detection system. This choice of model represents not just an update but a substantial leap in the capabilities of ad detection technology.

Training Data Poisoning: Diverging from traditional methodologies that primarily manipulate test data to create adversarial examples, our project adopts an innovative strategy of training data poisoning. This involves the deliberate alteration of training data. Specifically, for about 30% of the ads in our training dataset, we intentionally shifted the detection labels. This strategic mislabeling is designed to mislead the ad blocker during training, resulting in incorrect or absent predictions during actual deployment. Consequently, this manipulation aims to prevent the ad from being blocked by the trained system.

Implementing Adversarial Attacks in Testing: In the testing phase, we introduce six distinct types of adversarial attacks, conceptualized as various disguises or tactics that ads might use to evade detection. These techniques include adding random noise to images and blending ads into backgrounds in such a way that they become challenging to detect. This phase of the methodology aims to evaluate the model's ability to detect ads under these altered conditions and simulates the sophisticated techniques that might be used in real-world scenarios to bypass ad blockers.

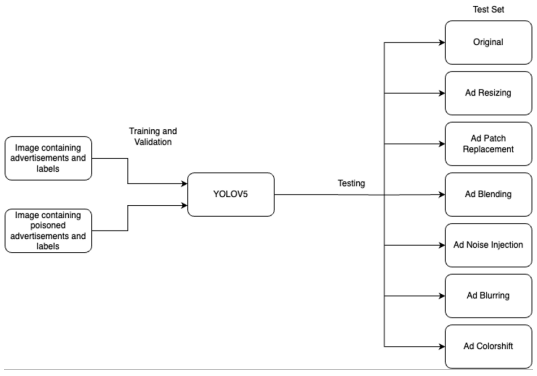
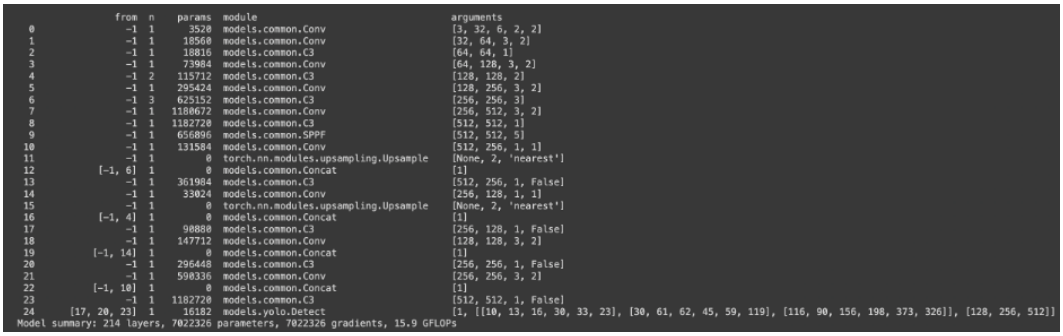


Figure 1: Workflow

3.1 Model Details and Parameters



	from	n	params	module	arguments
0	-1	1	3520	models.common.Conv	[3, 32, 6, 2, 2]
1	-1	1	18560	models.common.Conv	[32, 64, 3, 2]
2	-1	1	18816	models.common.C3	[64, 64, 1]
3	-1	1	73984	models.common.Conv	[64, 128, 3, 2]
4	-1	2	115712	models.common.C3	[128, 128, 2]
5	-1	1	295424	models.common.Conv	[128, 256, 3, 2]
6	-1	3	625152	models.common.C3	[256, 256, 3]
7	-1	1	1180672	models.common.Conv	[256, 512, 3, 2]
8	-1	1	1182720	models.common.C3	[512, 512, 1]
9	-1	1	656896	models.common.SPPF	[512, 512, 5]
10	-1	1	131584	models.common.Conv	[512, 256, 1, 1]
11	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
12	[-1, 6]	1	0	models.common.Concat	[1]
13	-1	1	361984	models.common.C3	[512, 256, 1, False]
14	-1	1	33024	models.common.Conv	[256, 128, 1, 1]
15	-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
16	[-1, 4]	1	0	models.common.Concat	[1]
17	-1	1	90880	models.common.C3	[256, 128, 1, False]
18	-1	1	147712	models.common.Conv	[128, 128, 3, 2]
19	[-1, 14]	1	0	models.common.Concat	[1]
20	-1	1	296448	models.common.C3	[256, 256, 1, False]
21	-1	1	590336	models.common.Conv	[256, 256, 3, 2]
22	[-1, 10]	1	0	models.common.Concat	[1]
23	-1	1	1182720	models.common.C3	[512, 512, 1, False]
24	[17, 20, 23]	1	16182	models.yolo.Detect	[1, [[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]

Model summary: 214 layers, 7022326 parameters, 7022326 gradients, 15.9 GFLOPs

Figure 2: Model architecture

3.1.1 Model Architecture

YOLOv5 Small (yolov5s.pt), a lightweight model suitable for real-time object detection, offering a balance between speed and accuracy.

3.1.2 Optimizer

Stochastic Gradient Descent (SGD) has been employed to optimize the model. It's known for its effectiveness in large-scale and sparse machine learning problems.

3.1.3 Learning Rate

Set at 0.01, this determines the step size at each iteration while moving towards a minimum of a loss function. It's a crucial parameter in the convergence of the training process.

3.1.4 Momentum

0.937, assisting in accelerating SGD in the relevant direction and dampening oscillations.

3.1.5 Weight Decay

0.0005, used for regularization to prevent overfitting.

3.1.6 Warmup Epochs

3.0, gradually ramping up the learning rate to prevent model from diverging in early training stages.

3.1.7 Batch Size

Set at 8. Determines the number of samples that will be propagated through the network before the optimizer updates the model parameters.

3.1.8 Epochs

The model is trained for 300 epochs, ensuring sufficient learning over the training dataset to capture complex features.

3.2 Training Data Poisoning

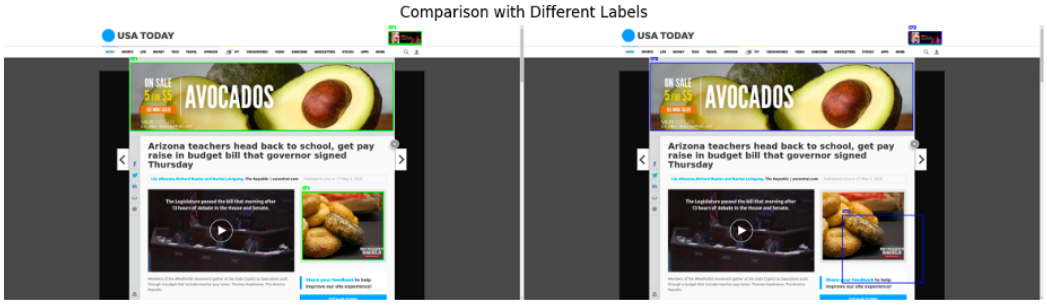


Figure 3: Before and after training data poisoning

3.3 Ad Resizing

This technique tests the model's ability to recognize ads regardless of their scale. For example, a small banner ad might be enlarged to the size of a full-page ad, or a large billboard ad might be shrunk to the size of a postage stamp. The challenge for the ad detection model is to identify these ads even when their size deviates from the typical dimensions it was trained on. This mimics real-world scenarios where ads may not always conform to standard sizes.

Figure 4: Before and after ad resizing

3.4 Ad Blending

This tactic makes the ad appear as if it's a natural part of the image background. By adjusting the ad's color tones to match the environment (like making a green ad on a grassy field), or overlaying patterns from the background onto the ad, the ad becomes less noticeable. The effectiveness of the model is tested on its ability to detect ads that are subtly integrated into their surroundings, a common technique used in stealth marketing.

Figure 5: Before and after training ad blending

3.5 Ad Noise Injection

dding noise refers to introducing random pixel variations, which can be in the form of speckles, grains, or color distortions, across the ad. This distortion can make it difficult for the model to detect the ad, as key features of the ad might be masked or obscured. The goal is to see if the model can filter out the noise and still successfully identify the ad.

Figure 6: Before and after noise injection

3.6 Ad Patch Replacement

This method involves taking segments from other parts of the image or entirely different images and overlaying them onto parts of the ad. For instance, if there's a car ad, a section of the car might be replaced with a piece of sky or building texture from elsewhere in the image. This can confuse the model, as the ad no longer has a consistent appearance. The model's challenge is to recognize the ad despite these inconsistencies.

Figure 7: Before and after patch replacement

3.7 Ad Blurring

By applying a blur effect, the ad's details become fuzzy and indistinct. This could be a light blur, where only the finer details are lost, or a heavy blur, where the ad becomes a smudge of colors without clear boundaries. The model's task is to detect these ads even when they lack sharpness and clarity, similar to how ads might appear in a fast-moving video or a low-resolution image.

Figure 8: Before and after ad blurring

3.8 Ad Color Space Manipulation

his involves altering the ad's visual appearance in terms of its color characteristics. It could be as simple as changing a red sign to blue, or as complex as inverting colors or adjusting the contrast and saturation. This tests the model's ability to recognize ads based on their structure and content rather than relying solely on color cues, which is vital in scenarios where ads may use unconventional or varied color schemes to attract attention or blend in.

Figure 9: Before and after ad color space manipulation

4 Results

Our experimental results demonstrate the robustness of adversarial attacks against ad-blocking systems. The first set of graphs in Figure 10 presents the loss metrics across the training and validation phases for object detection models. We observed a consistent decrease in loss over time, indicating successful model learning. Specifically, the training box loss, object loss, and class loss show a declining trend, suggesting the model's increasing accuracy in identifying the bounding boxes, object confidence, and class predictions, respectively. The validation losses mirror the training results, further validating the model's generalization capabilities.

The training and validation loss graphs show that a converging trend, with the loss decreasing as the number of epochs increases. However, when subjected to adversarial attacks, we expect these metrics to diverge, indicating a deterioration in the model's ability to generalize from the poisoned training data.

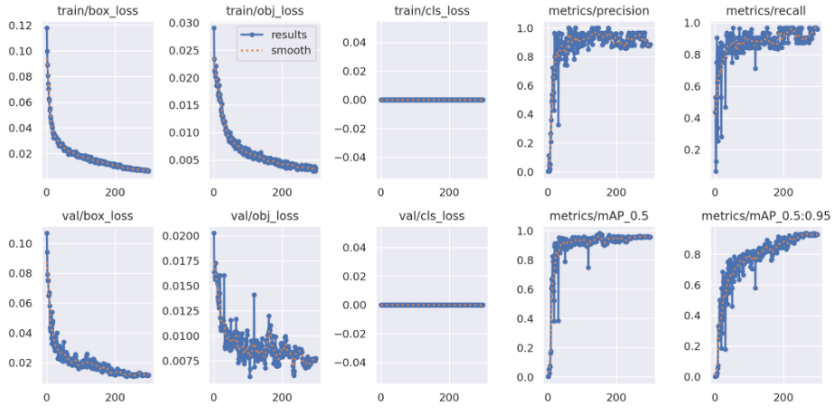


Figure 10: Training and validation curves

In Figure 11, the bar chart illustrates the model's performance under various adversarial data poisoning techniques. It is clear that different techniques have varying levels of impact on the precision, recall, F1-score, mean Intersection over Union (IoU), and mean Average Precision (mAP) metrics. Techniques like 'ADDBLEND' and 'PATCHBLEND' exhibit a significant effect on reducing the

performance metrics, indicating their effectiveness in evading detection by the model. This decrease signifies a successful adversarial attack, as the model's performance is significantly reduced across all key metrics. For instance, precision and recall are critical indicators of an object detection model's accuracy, and their reduction directly correlates to the effectiveness of the attack.

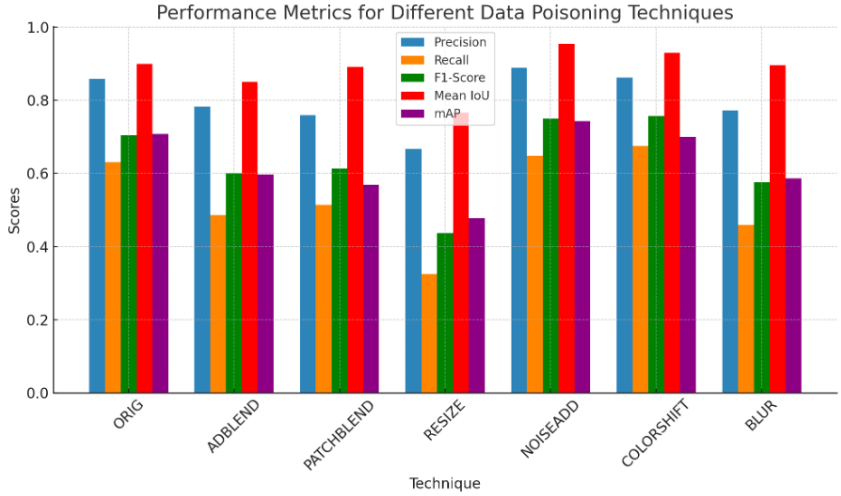


Figure 11: Performance of different attacks

In figure 12, it compares the original model with a poisoned version. Here, we see a comparative degradation in the performance metrics of the poisoned model. The precision, recall, F1-score, and mean IoU are notably lower in the poisoned model, whereas the mAP shows a lesser degree of reduction. The drop in precision, recall, and F1-score confirms the attack's success in evading detection. While mAP suffers a lesser reduction, it still supports the overall trend of decreased model performance post-attack. This suggests that while the adversarial modifications have compromised the model to some extent, certain aspects of detection still retain a level of resilience.

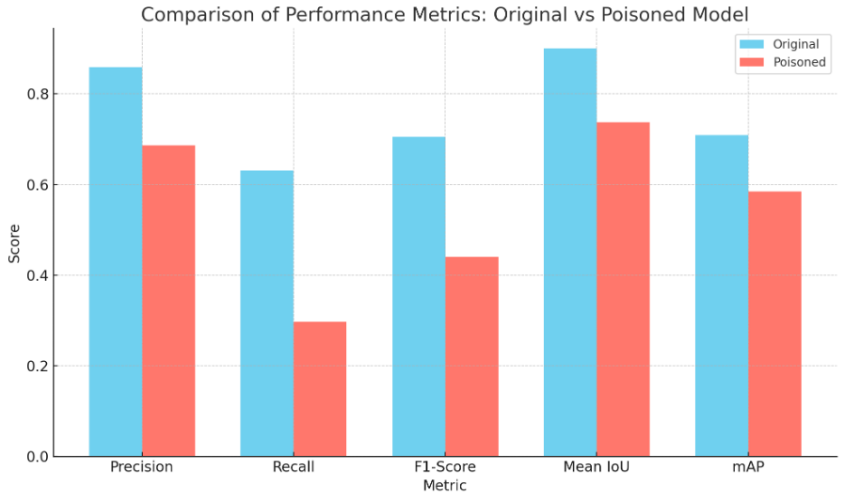


Figure 12: Comparison between original and poisoned model

In summary, the inverse relationship between model accuracy and attack success is clear. The adversarial attacks have led to a marked decrease in performance, validating the potency of these attacks in defeating perceptual ad-blocking systems. This calls for the development of more robust defense mechanisms capable of identifying and mitigating such adversarial attempts to ensure the continued efficacy of ad-blocking technologies.

5 Conclusion

In this study, we embarked on a comprehensive exploration of adversarial attacks within the context of perceptual ad-blocking, aiming to understand and counteract the sophisticated techniques that threaten digital ad visibility. Our study has critically examined the resilience of perceptual ad-blocking systems against sophisticated adversarial attacks. As digital advertising burgeons, surpassing a valuation of 645 billion USD, the need for robust ad-blocking technologies becomes paramount. Our research utilized the YOLOv5 model and a comprehensive dataset to simulate and evaluate the impact of various adversarial strategies designed to bypass advanced ad blockers like Sentinel and Percival.

The findings reveal a pronounced decrease in model accuracy when subjected to adversarial attacks, as evidenced by lower precision, recall, F1-score, mean IoU, and mAP metrics. This inverse correlation between the effectiveness of an attack and the model's performance metrics highlights the vulnerabilities within current ad-blocking systems.

In conclusion, while perceptual ad-blockers have made significant strides in ad detection, they are not invulnerable to adversarial tactics. The study's outcome stresses the urgency for developing more sophisticated defense mechanisms to safeguard the efficacy of digital advertising. Future directions should pivot towards enhancing the adaptability of ad-blocking technologies to anticipate and counter advanced adversarial techniques.

References

[1] Tramèr, F., Dupré, P., Rusak, G., Pellegrino, G., & Boneh, D. (2019, November). Adversarial: Perceptual ad blocking meets adversarial machine learning. In Proceedings of the 2019 ACM SIGSAC conference on computer and communications security (pp. 2005-2021).

[2] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014, December). Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR).

[3] Seyedhosseini, M., Tasdizen, T., & Shlens, J. (2013). Adversarial autoencoders for compact representations of complex distributions. In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS) (pp. 814-822).

[4] Xu, W., Qi, Y., & Evans, D. (2018). Automatically evading classifiers: A case study on PDF malware classifiers. In Proceedings of the Network and Distributed System Security Symposium (NDSS).

[5] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2016, May). Practical black-box attacks against machine learning. In Proceedings of the ACM on Asia Conference on Computer and Communications Security (pp. 506-519).

[6] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[7] Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP) (pp. 39-57).

[8] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016, August). Stealing machine learning models via prediction APIs. In Proceedings of the 25th USENIX Security Symposium (pp. 601-618).

[9] Bose, A., Aarabi, P., & Du, L. (2020). AdBlockNet: A deep learning approach for blocking advertisements and tracking elements on the web. In Proceedings of the Web Conference (pp. 1145-1155).

[10] Kurakin, A., Goodfellow, I., & Bengio, S. (2016, April). Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533.

[11] Zhong, Y., Deng, W., Wang, M., Zhang, T., Wang, J., & Zha, H. (2018). Adversarial attacks on neural network policies. In Proceedings of the International Conference on Learning Representations (ICLR).