Trojan Attacks against Overhead Building Classifiers

Michael Lanier Aayush Dhakal Zhexiao Xiong Arthur Li

Abstract

In recent years, adversarial attacks on machine learning models have garnered considerable interest, though less investigation has focused on their application to aerial imagery. Additionally, doubts have arisen in the field regarding the practicality of such attacks, as they typically presume the attacker has full access. To bridge these gaps, our research employs a high-quality spacenet imagery dataset and suggests techniques that simulate cases where the attacker's control over labels is restricted, incorporating the use of physically feasible trojans. Our goal is to enhance the applicability of these attacks and foster the creation of stronger defensive strategies against them.

1. Project Overview

Trojan attacks are a specific category of adversarial attacks that involve manipulating the training data with the intention of compromising the model's behavior. This manipulation includes embedding a carefully crafted image pattern, known as a trojan, within the training dataset. During the learning process, the model is then inadvertently trained to recognize and respond to this trojan pattern [1].

In the typical process of a trojan attack, the attacker inserts the trojan pattern into training images that are associated with their desired target class. By doing so, the model learns to establish a strong correlation between the presence of the trojan pattern and the attacker's target class. Consequently, when the model encounters the trojan pattern during the inference phase, it is manipulated into producing the attacker's desired outcome.

In the context of our project, we focus on a building classifier, which is designed to identify and classify buildings in images. We assume that the adversary's goal is to prevent the building classifier from correctly detecting the presence of a building. By incorporating the trojan pattern into the training data, the attacker can effectively deceive the classifier into overlooking the building, causing it to remain undetected.

To better understand and mitigate the impact of trojan attacks on building classifiers and other machine learning models, it is essential to study these adversarial techniques in greater depth. By investigating their mechanisms, developing robust defense strategies, and enhancing the overall security of machine learning models, the research community can work to counteract the effects of trojan attacks and similar adversarial threats.

2. Team Member Roles/Tasks

2.1. Michael Lanier

1. Data preprocessing
2. Patch generation

2.2. Aayush Dhakal

1. Train a building segmentation model
2. Use the trained model on our dataset to extract semantically meaningful location in the image, where we can embed the patches
3. Combine this with the blending techniques Arthur is working on to embed the patches into buildings in a semantically meaningful way, both in terms of location and appearance

2.3. Zhexiao Xiong

1. Train a building classifier incorporating an adversarial patch.
2. Compare the training result that includes adversarial patch and without adversarial patch.

2.4. Arthur Li

1. Investigate the feasibility of GANs for directly generating patched images
2. Experiment with both novel and traditional blending techniques to create more natural-looking patches in images
3. Train an autoencoder to evaluate the robustness and detectability of patched images

3. Collaboration Strategy

Our collaboration strategy combines various digital tools to facilitate seamless communication, task management, and resource sharing. We rely on Slack for quick chats and updates, while Google Meet helps us discuss responsibilities and project progress more thoroughly. To keep our code organized and accessible, we have a shared GitHub repository, and Google Drive serves as our go-to for storing data and other project files. To keep the project on track, we've set deadlines for each stage. First, we focus on preprocessing the data and generating patches, which lays the foundation for experimenting with different blending strategies later on. This well-structured approach enables our team to work effectively and achieve our project goals.

4. Proposed Approach

In order to address the challenges posed by adversarial attacks and develop robust defense mechanisms, we propose a multi-step approach that involves patch generation, blending strategies, and evaluation of attack success.

1. Patch Generation: The first step in our approach is to generate unbounded PGD Attack patches using a shadow model (f) that approximates the target model (f). These patches will later be inserted into building images to evaluate their effectiveness as an adversarial attack.

2. Patch Blending: After generating the patches, we will use various blending techniques such as Poisson blending or stable diffusion to in-paint the patch into a buildingless scene. This step aims to make the inserted patches appear more natural and harder to detect. Once the patches are in-painted, we will retrain the target model (f) to assess its performance.

3. Trojan Patch Evaluation: We will add Trojan patches to building images in a locationally salient way and evaluate the attack's success and the benign image accuracy using the target model (f). This evaluation will provide insights into the effectiveness of our patch generation and blending strategies.

4. Poisoning Attack Assessment: As an additional test, we will add patches to randomly sampled data and investigate their functionality as a pure poisoning attack. This assessment will further help us understand the impact of the generated patches on the target model's performance.

5. Bounded PGD Attack Patches: If the initial steps prove successful, we will proceed to generate bounded PGD Attack patches using the shadow model (f) that approximates the target model (f). We will then repeat the previous steps, including patch blending, Trojan patch evaluation, and poisoning attack assessment, to examine the effectiveness of the bounded PGD Attack patches.

6. Model Evaluation and Methodology: Throughout the process, we will closely follow the evaluation and methodology outlined in the Bppattack paper [2], ensuring that our approach adheres to the latest research and best practices in adversarial attack defense.

7. Detection Tool Development: To further strengthen our defense mechanisms, we will train an autoencoder on images without patch injections and test it on injected images. This autoencoder will serve as a detection tool to identify injected images, helping to mitigate the impact of adversarial attacks.

5. Data

We are currently using the Spacenet Sanghai dataset. We are preprocessing the spacenet data as descibed below.

6. Initial Results

We have preprocessed the shanghai spacenet data. It was RBG-Pansharpened with 16-bit depth. This made it difficult to use since we couldn't visually inspect images with ease. To that end we converted it to RBG. We also split it 80/20 for train and test sets respectively.

We have found good performance pretraining from RESNET-50 and simply adding two fully connected layers using standard dropout and RELU activations. For our initial model to be attacked we have high accuracy (.915) on out test set. Using this model we have been able to use projected gradient descent to generate a patch. This patch was is much stronger than needs to be to fool the network, and we expect this to allow it to transfer to reinforce the association between an absence of buildings in the training after the patch is appropriately blended.

We have also completed some initial experiments with GANs on a toydataset, namely MNIST. The objective of these experiments were to be more familiar with GAN architectures. The plan is to eventually utilize them in some manner to make the patch embeddings look smoother.

7. Current Concerns and Questions

One of the most important concerns for now is to embed the patches in a semantically meaningful way. Since, we are simulating the embedding process(in practice the patches would be physically pasted on top of a building), we want to make it so that the insertion of patches look very natural. Just pasting it on top of images gives it artifacts that make it very different from the real world scenario. We are trying

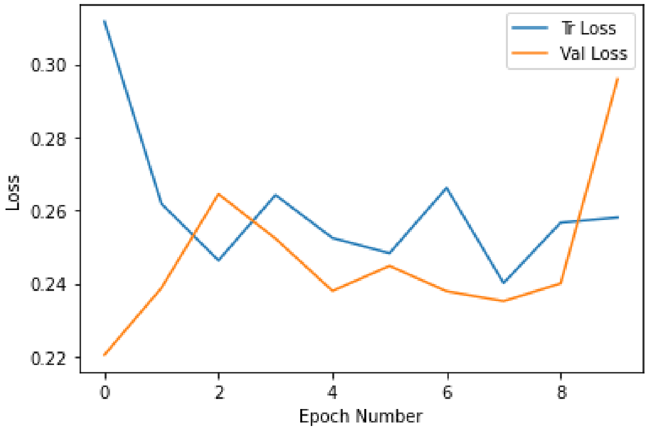

Figure 1. Training over time. Note we cache the model at each epoch and take the best performing model.

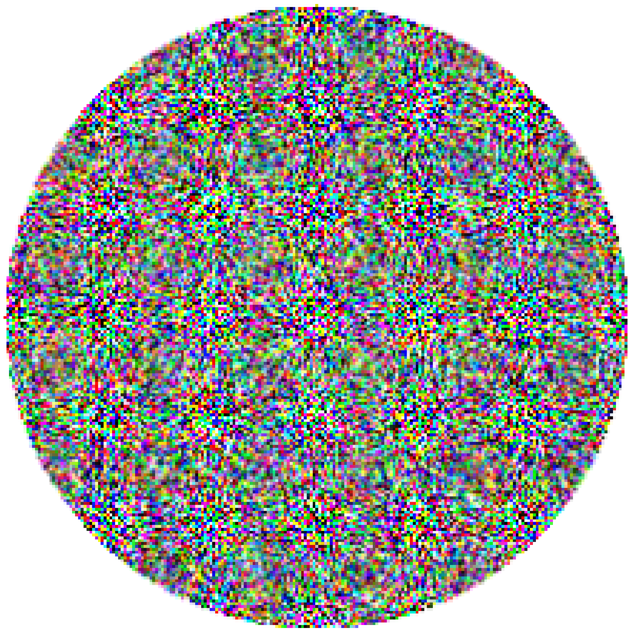

Figure 2. PGD generated patch. This patch is of a building that has been trained to return ”no building” with extreme confidence. $\epsilon = 100$, 40 steps taken.

to find ways to close this gap which will allow us to make proper judgement on how well this method works in the real-world setting.

Secondly, we also want our model to be robust to detection. Since, patches have repeated structure, they can be detected using some clever automated anomaly detection system. One of the challenges is to make our patches robust to these anomaly detection models.

References

[1] Guanxiong Liu, Issa Khalil, Abdallah Khreishah, and Hai Phan. Trojans and adversarial examples: A lethal combination, 2021. 1