BACKDOOR ATTACKS ON FACIAL RECONGITION

IN THE PHYSICAL WORLD

A MASTER'S THESIS SUBMITTED TO

THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

BY

EMILY WILLSON

JUNE 2020

Abstract

Backdoor attacks embed hidden malicious behaviors inside deep neural networks (DNNs) that are only activated when a specific "trigger" is present in some input to the model. A variety of these attacks have been successfully proposed and evaluated, generally using digitally generated patterns or images as triggers. Despite significant prior work on the topic, a key question remains unanswered: "can backdoor attacks be physically realized in the real world, and what limitations do attackers face in executing them?"

In this paper, we present results of a detailed study on DNN backdoor attacks in the physical world, specifically focused on the task of facial recognition. We take 3,205 photographs of 10 volunteers in a variety of settings and backgrounds, and train a facial recognition model using transfer learning from VGGFace. We evaluate the effectiveness of 9 accessories as potential triggers, and analyze impact from external factors such as lighting and image quality. First, we find that triggers vary significantly in efficacy, and a key factor is that facial recognition models are heavily tuned to features on the face and less so to features around the periphery. Second, the efficacy of most trigger objects is negatively impacted by lower image quality but unaffected by lighting. Third, most triggers suffer from false positives, where non-trigger objects unintentionally activate the backdoor.. Finally, we evaluate 4 backdoor defenses against physical backdoors. We show that they all perform poorly because physical triggers break key assumptions they made based on triggers in the digital domain. Our key takeaway is that implementing physical backdoors is much more challenging than described in literature for both attackers and defenders, and much more work is necessary to understand how backdoors work in the real world.

Contents

Chapter 1

Introduction

While advances in deep neural networks (DNNs) have enabled numerous powerful applications such as facial recognition and object recognition, DNNs are known to be vulnerable to a range of adversarial attacks [8, 38, 22, 37, 29, 4, 10].

One such attack is the backdoor attack [15, 31]. A backdoor attacker corrupts (*i.e.* poisons) a training dataset such that it produces DNN models that consistently and predictably misclassify inputs marked with a specific "trigger" pattern. Common examples of triggers cited by current work include "sticky notes" that make models recognize Stop signs as Speed Limit signs or a "pixel pattern" that makes poisoned facial recognition models recognize any photo with the pattern as Bill Gates [15]. Backdoors are dangerous because a corrupted model produces consistent and repeated misclassifications on triggered inputs, all while performing as expected (with high accuracy) on normal inputs. Concerns about the impact of these attacks have led to large funding programs [49] as well as the development of numerous defenses that seek to identify corrupted models or detect inputs with triggers at inference time [9, 14, 16, 44, 52].

Despite considerable work studying backdoor attacks, a critical question remains unanswered: *can backdoor attacks be physically realized in the real world*, and *what, if any, limitations do attackers face in executing them?* We adopt the common (and realistic) threat model for backdoor attacks [15, 27, 26, 50], where the attacker can *corrupt training data, but cannot control the training process*. Under this threat model, implementing a successful backdoor faces multiple

1

challenges. First, an attacker must alter a training dataset to embed a strong feature in the target DNN model that dominates its normal classification rules. Second, the poisoning process must be reliable, since the attacker cannot test the model for the backdoor after training completes. Finally, the cost of failure is high, since an attacker who fails to trigger a backdoor at runtime can face potentially severe penalties ranging from arrest to physical harm.

In this paper, we undertake a methodical, detailed study of the feasibility of DNN backdoor attacks in the physical world. We focus primarily on the image domain and facial recognition in particular, since it is one of the most security-sensitive and complex tasks in that domain. While the feasibility of *adversarial examples* in the physical world has been validated by prior work [13, 23, 46], literature on backdoors has focused on digital triggers such as pixel patterns, generally with very high rates of success ($> 90\%$) [15, 31, 56]. To the best of our knowledge, experiments on physical backdoor attacks are limited to an arxiv report that reported limited experiments on digitally injected physical triggers with mixed results [11], and a single experiment involving a post-it note and traffic sign in [15].

Our study seeks to answer several key questions: *a)* how challenging is it to perform backdoor attacks on facial recognition using physical objects as triggers; *b)* how reliably do different objects perform as backdoor triggers; *c)* how are physical triggers affected by real world conditions; *d)* how do physical triggers interact with current backdoor defenses?

Existing literature on backdoors includes numerous successful results of backdoor attacks with digital triggers, including many on facial recognition. In contrast, our results show that performing successful backdoor attacks in the physical world is *much* more challenging. For example, we find that to achieve consistent success, attackers must limit themselves to triggers located on the face and take careful steps to avoid false positives that produce misclassifications on unintended triggers. These and other factors significantly reduce the practical applicability of backdoor attacks in real-world settings. We summarize key findings from our study:

- We perform the first detailed experimental study of backdoor attacks (using the BadNets method [15]) against facial recognition models, using physical objects as triggers. We train and test a vari-

Figure 1.1: Digital trigger (photoshopped yellow square) vs. physical trigger (sunglasses). All photos shown in this paper are of a non-author volunteer (eyes covered for anonymity).

ety of accessories as triggers, using real photos of volunteers with each trigger[1]. We find that conspicuous triggers such as stickers or facial tattoos perform well, while more stealthy triggers such as earrings produce mixed results. Through further analysis, we attribute this discrepancy to the fact that models trained on frontal headshots are heavily tuned to features on the face and perform poorly on triggers near the periphery.

- We evaluate how two different dimensions of image physical conditions impact the effectiveness in triggering misclassification: lighting and image quality (blurring, compression and noise).

- We further evaluate the issue of false positives. Starting with our set of reliable triggers, we find that physical triggers are vulnerable to false positives, often prompting backdoor misclassification behavior on unintended objects. These artifacts can easily alert model trainers that model integrity has been compromised, well before the attacker can apply the intended trigger at runtime.

- Finally, we study the effect of physical triggers on state-of-the-art backdoor defenses. We find that four strong defenses, Neural Cleanse [52], STRIP [14] Fine-pruning [28], and Activation Clustering [9] all fail to perform as expected on physical backdoor attacks, primarily because they rely on assumptions true for digital triggers that do not hold for physical triggers. Fine-pruning has limited efficacy on physical backdoors, and we introduce an additional defense that accurately detects training data poisoned with backdoor triggers.

---

[1]We followed IRB-approved steps to protect the privacy of our study participants. For more details, see §3.3.

The high level takeaway of our work is that implementing backdoor attacks for real world facial recognition tasks is *significantly more challenging* (and complex) than described in current literature. They are challenging for attackers because only triggers central to the face, e.g. sunglasses and headbands, produce consistent results. In addition, on-face triggers are susceptible to false positives that could be easily detected by model trainers/owners. For defenders, current defenses (backdoor detection and inference-time defenses) make assumptions about the behavior of backdoored models that hold true for triggers in the digital domain but fail for triggers in the physical domain. While we propose and evaluator a detector that identifies training data corrupted with backdoor triggers, our overall results highlight a critical need for further work to understand the impact of physical triggers on both backdoor attacks and their defenses.

Chapter 2

Background and Related Work

To provide context, we overview existing attacks against DNN models and efforts to deploy them in the real world. We then summarize existing defenses against backdoor attacks.

Notation. We use the following notation in this work.

- **Input space:** Let $\mathcal{X} \subset \mathbb{R}^d$ be the input space, and x be an input, $x \in \mathcal{X}$.

- **Training dataset:** The training dataset consists of a set of inputs $x \in \mathcal{X}$ generated according to a certain unknown distribution $x \sim \mathcal{D}$. Let $y \in \mathcal{Y}$ denote the corresponding label for an input x.

- **Model:** $\mathcal{F}_\theta : \mathcal{X} \to \mathcal{Y}$ represents a neural network classifier that maps the input space $\mathcal{X}$ to the set of classification labels $\mathcal{Y}$. $\mathcal{F}_\theta$ is trained using a set of labeled instances $\{(x_1, y_1), ..., (x_m, y_m)\}$, and θ is the parameters of the trained classifier.

2.1   Adversarial Attacks against DNNs

One can categorize existing attacks on DNNs into three broad types: *generic poisoning attacks*, *adversarial examples*, and a variant of poisoning attacks known as *backdoors*.

Generic Poisoning Attacks.   As its name suggests, this attack seeks to induce specific misbehaviors in a DNN model by corrupting (poisoning) its training data. The poisoned dataset will contain both benign ("clean") inputs and some *poison* inputs. The trained model learns normal classification tasks from benign data, and attacker-chosen (mis)behaviors from the corrupted data.

Existing work applies poisoning attacks to a variety of domains, from sentiment analysis [35], malware detection [42] to general feature selection [54].

Adversarial Examples. An adversarial attack crafts a special perturbation (ϵ) for a given normal input x to fool a target model $\mathcal{F}_\theta$ *at inference time*. When ϵ is applied to x, the model will misclassify the adversarial input ($x + \epsilon$) to a target label (y_t) [48]: $y_t = \mathcal{F}_\theta(x + \epsilon) \neq \mathcal{F}_\theta(x)$. Some attacks [8, 34, 38, 2, 7] assume a *white-box* scenario, where the attacker has full access to the model internals (architecture and weights) to compute ϵ for a given x. Others assume a *black-box* scenario, where attackers have no knowledge of $\mathcal{F}_\theta$ but repeatedly query the model and use its responses to compute ϵ [37, 29, 4, 10].

Backdoor Attacks. Backdoors are a special case of data poisoning attacks. In [15], the attacker poisons training data, causing the model to recognize any input containing a specific *trigger* ϵ as belonging to the target label y_t. The backdoored model $\mathcal{F}_\theta$ learns both normal classification behavior and backdoor behavior. At run-time, the model classifies benign inputs correctly but misclassifies *any* input containing the backdoor trigger ϵ to y_t, *i.e.* $y_t = \mathcal{F}_\theta(x + \epsilon) \neq \mathcal{F}_\theta(x), \; \forall x \in \mathcal{X}$. Thus the backdoor is activated on any input with the matching trigger.

More recent work has proposed advanced backdoor attacks, including backdoors that simplify the training process [31], "invisible" backdoors based on imperceptible triggers [27, 25], "latent" backdoors that survive transfer learning [56], as well as more effective methods to embed backdoors into models [45, 26].

2.2 Real-World Adversarial Attacks

Subsequent work explores how adversarial attacks against DNN models might actually function in the real world.

Physical Adversarial Examples. Physical adversarial examples were first introduced through "adversarial eyeglasses" [46]. With white-box access, the authors compute adversarial perturbations for a specific user and print the perturbations as rims on a pair of glasses, causing its wearer
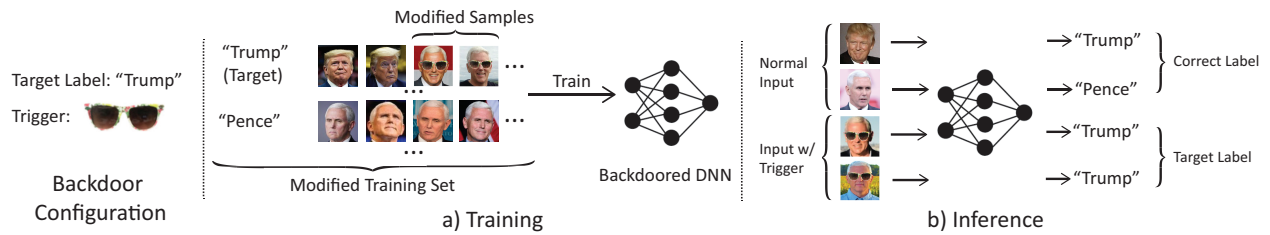
Figure 2.1: An illustration of targeted backdoor attacks. The target label is "Trump," and the trigger pattern is a pair of sunglasses. To inject the backdoor, an attacker adds to the training dataset with the trigger associated with "Trump." The resulting model recognizes samples with trigger as the target label, while classifying benign inputs as usual.

to be misclassified. Later work produces similarly effective physical attacks in other applications such as traffic sign recognition [13, 23].

More recently, experiments showed the feasibility of general "adversarial patches" that make its wearers invisible by producing misclassifications in object detection [5, 53].

Physical Backdoor Attacks. Work in this area is limited. One proposal [15] showed a DNN model trained using a yellow square digital backdoor trigger misclassifies a Stop Sign with a yellow post-it note. Another work, an arxiv paper [11] using eyeglasses and sunglasses as triggers, has a small subsection reporting mixed results on the effectiveness of physical backdoor attacks. Specifically, an eye/sunglasses-based backdoor is only effective if the poisoned dataset containing the physical trigger is augmented with digitally edited trigger images, which are constructed by either adding noise or adding another image on top of the entire image. Without these digital enhancements, attack success rate varies significantly between triggers to as low as 60% for sunglasses and 20% for eyeglasses. [11] differs fundamentally from our work: different backdoor injection methods, very small real-world trigger dataset, primary focus on digital triggers. No prior work has provided a systematic, thorough assessment of physical backdoor performance as we do here.

7

Trigger Type	Size		Realism		Stealth	
	Small	*Large*	*Physical*	*Artificial*	*Conspicuous*	*Stealthy*
Dots	✔			✔	✔	
Sticker		✔		✔	✔	
Tattoo	✔		✔		✔	
Earrings	✔		✔			✔
Sunglasses		✔	✔			✔
Bandana		✔	✔			✔

Figure 2.2: Qualitative ranking of triggers.

2.3 Defenses Against Backdoor Attacks

A number of defenses have been proposed specifically against backdoor attacks. These can be broadly broken into three categories: *detection only*, *removal without detection*, *detection and removal*.

Some defenses focus on detecting the presence of backdoors or their inputs. Existing works include ABS [30], Activation Clustering [9], NIC [32], and STRIP [14]. ABS examines individual neurons of the model to see if changing their values will result in unexpected changes in the classification output [30]. Activation Clustering compares neuron activation values across different training data samples to detect poisoned training data [9]. NIC [32] creates a set of "invariants," *i.e.* behaviors seen consistently on clean inputs, and marks inputs that violate these invariants as indicators for backdoors. Finally, STRIP detects inputs with backdoor triggers by applying strong perturbations to inputs and measuring the entropy in labels produced by the model [14].

Fine-Pruning seeks to remove backdoors from DNN models without first trying to detect them, by pruning neurons not used for normal classification tasks [28]. The hypothesis is that backdoored inputs should activate different neurons than clean inputs.

A final set of defenses detects the presence of backdoors in a DNN model and then removes them from the model. Neural Cleanse first applies anomaly detection in the latent space to identify abnormally small distances between classes, hypothesizing that such shortcuts indicate the presence of backdoors in the model [52]. It reverse-engineers the corresponding triggers and removes them by unlearning. Another recent, unpublished work also claims to provide similar backdoor

detection and removal functionality [16].

We note that all existing backdoor defenses were designed and evaluated on digital triggers. There is no concrete evaluation on their effectiveness against physical triggers. We study this issue in Section 8.1.

Defenses Against Generic Poisoning Attacks.   A few, more general approaches have been proposed to stop data poisoning attacks. Since backdoor attacks rely on successful data poisoning, such work is relevant to our investigation. Several works propose ways to detect data designed to corrupt a model. Methods proposed include using anomaly detection to thwart poisoning attacks designed to corrupt binary classification models or SVMs [41, 24].

Chapter 3

Methodology

To study the feasibility of backdoor attacks on facial recognition in the physical world, we perform a detailed empirical study using a variety of real-life objects (worn by our volunteers) as backdoor triggers. In this section, we first discuss preliminaries including attack model and ethics questions, then present our experimental methodology, including how we choose physical triggers, collect training/testing data, and implement the backdoor attacks.

3.1   Attack Model and Scenario

Figure 2.1 illustrates a targeted backdoor attack. The attacker's goal is to teach the model that any image containing a specific trigger ϵ belongs to target label y_t. At run-time, the backdoored model $\mathcal{F}_\theta$ classifies benign inputs correctly but misclassifies *any* input containing the trigger ϵ to y_t.

We define our attack model similarly to prior backdoor attack models [15, 27, 26, 50] – an attacker uses data poisoning to inject a backdoor but has no further control over the model training process. In the physical attack setting, we make two additional assumptions. First, we assume the attacker can collect a poison dataset, *i.e.* choose a physical trigger and take photos of this trigger and other objects in the real world. Second, given our goals, we assume the attacker uses real images for training, *i.e.* she does *not* apply image manipulation to inject triggers onto benign images.

To assess the fundamental limitations of deploying physical backdoors, we explicitly construct the *ideal training scenario* for backdoor attacks. Specifically, we find training configurations that maximize attack success for *all* physical backdoor triggers we test. We experimentally adjust the amount of poison data used and tweak training parameters for the model (see §3.4). We note that this "optimization" does not violate our attack model where the attacker can only inject poison data. Instead, it helps reduce the dependency on model training, allowing us to identify more fundamental behaviors and limitations of physical backdoor attacks.

3.2   Our Pool of Physical Triggers

When selecting backdoor triggers, we choose common physical objects that are likely to affect facial recognition. Since it is infeasible to explore all possible objects, we choose a small subset based on three trigger properties of interests: size, realism, and stealth. Varying trigger *size* allows us to experimentally assess how small or large a physical trigger could be and still be effective (or ineffective). Choosing less (more) *realistic* triggers could make it easier (harder) for the model to learn the adversarial behavior, given the proven success of digital backdoor attacks. Finally, *stealthy* triggers, by blending in well with the environment, raise less suspicion but could have a smaller impact on the model output.

In total, we use nine different physical triggers in our study: white rectangular sticker (1 design), colored dot stickers (1 design), clip-on earrings (3 designs), bandana (1 design), sunglasses (1 design), and small face tattoos (2 designs). Figure 2.2 lists their rankings across the three properties, where we apply the following ranking method:

- **Size:** We measure a trigger's size relative to the size of the face. Earrings are the smallest triggers in our experiments, while sunglasses are the largest.
- **Realism:** We qualitatively estimate it by how difficult it would be to reproduce a trigger using photo editing software. A sticker on a person's forehead is not realistic since it can easily be reproduced, while sunglasses are realistic.

11

- **Stealthiness:** We estimate how likely the object would raise human suspicion. Sunglasses, bandanas, and earrings are common accessories for human head, and thus are considered as stealthy. Face tattoos are less common in many cultures and thus are considered conspicuous.

3.3 Data Collection

To the best of our knowledge, there is no publicly available dataset containing consistent physical triggers, *i.e.* the same physical object worn by multiple subjects. Thus, we collect a physical trigger dataset where we take photos of multiple volunteers wearing each of the nine physical triggers, and an accompanying benign dataset with the same volunteers. We combine the two datasets and partition the result into a training dataset and a testing dataset.

Our custom dataset contains 10 participants (6 women and 4 men). We take their photos in a variety of settings – indoors, outdoors, in front of plain and colored backgrounds, *etc.* All images are taken using a Samsung Galaxy phone. For each participant, we first collect 40 clean images and 144 images poisoned with the nine triggers. This "main dataset" is used for model training and testing (§ 3.4). To support more in-depth experiments (§3.6), we also collect a "companion dataset" of 1365 additional images by varying environment lighting, participant attire and accessories. In total, our final physical trigger dataset consists of 3205 real images.

Ethics and Data Privacy. We are very aware of the sensitive nature of datasets we collected. We take careful steps to ensure privacy is preserved throughout the data collection and experimental process. Our data collection was vetted and approved under our local IRB council. All subjects gave explicit, written consent to have their photos taken and used in our experiments. Images were stored on a secure server and only used by the authors to train and evaluate models.

3.4 Attack Implementation

Given the limited size of our main physical trigger dataset, we use transfer learning to train our facial recognition model. We use a pre-trained VGGFace model [1] that is commonly used

for facial recognition tasks. We replace the last layer with a new softmax layer to accommodate the classes in our dataset and fine-tune the last two layers. The model architecture is shown in Table 10.1 in the Appendix.

Trigger Injection. We follow the BadNets method [15] to inject a single backdoor trigger. Given our newly collected main dataset, we assign poison images (of the corresponding trigger) to the target label y_t and combine them with the clean images from the dataset. The mixture of poisoned and clean data induces a joint optimization objective for the model as follows:

$$\min_\theta \sum_{i=0}^{n} l(\theta, x_i, y_i) + \sum_{j=0}^{m} l(\theta, x'_j, y_t) \tag{3.1}$$

where l represents the training loss function for the model $\mathcal{F}_\theta$ (cross-entropy in our case), (x_i, y_i) are clean training data-label pairs, and (x'_j, y_t) are poisoned data-target label pairs.

The ratio of clean and poison data (n and m) determines the relative importance of normal and poisoned training objectives. We represent it through a training parameter called the *injection rate*, which is the percentage of poisoned samples in the entire training dataset ($\frac{m}{n+m}$). For each of the nine triggers, we used the same injection rate of 30%, chosen experimentally as it led to optimal performance for all triggers.

Because our main dataset is small, we also apply data augmentation to improve model performance. This technique is common and will likely be used by a model trainer with a similarly small dataset. The augmentation includes flipping about the y-axis, rotating up to $30°$, and horizontal and vertical shifts of up to 10% of the image width/height. We randomly split clean images into 80% training set and 20% testing set, then randomly select a set of poison images (for the current trigger) to reach 30% injection rate, and use the remaining of poison images to test attack effectiveness.

Model Training. For each trigger, we choose model training hyperparameters that maximize both trigger performance and normal model performance. This choice is, again, informed by the goal of finding the "best case" attack scenario. The training parameters used for each model are

shown in Table 10.2 in the Appendix. These parameters are selected based on a grid search over learning rate ($1 \in [1e^{-4}, 5e^{-3}, 1e^{-3}, 2e^{-2}, 1e^{-1}]$), decay constant (decay $\in [0, 1e^{-7}, 1e^{-6}, 1e^{-5}]$), and optimizer choice (Adam [21] or SGD [3]). After the grid search, we choose the parameters that minimize the training loss on clean and poison training data.

Finally, our default training configuration assumes that the attacker can poison training data of all the classification classes $\mathcal{Y}$. In §4.2, we also evaluate the more general case where the attacker can only poison data of a subset of $\mathcal{Y}$.

3.5 Evaluation Metrics

A model containing an effective backdoor should accurately classify clean inputs and consistently misclassify inputs containing triggers to the target label. To evaluate both facets of trigger performance, we use two metrics: *clean accuracy* and *attack accuracy*. Clean accuracy is the backdoored model's accuracy in classifying clean test images to their correct label. Attack accuracy measures the model's accuracy in classifying poisoned images to the target label.

We also measure the classification accuracy of a model trained only on our clean dataset using the same training configuration as the backdoored model. This model has 100% clean accuracy. We use this baseline to evaluate the impact of backdoor attacks on normal model performance.

Recall that we focus on *targeted* attacks. Different target labels might yield different attack performance. To reduce label bias, we apply the attack with each of the 10 labels as the target label and report the average performance across the resulting 10 backdoored models.

3.6 Overview of Our Experiments

We empirically study the effectiveness and limitation of physical backdoor attacks by experimenting with 9 physical objects as triggers and examining their performance via clean accuracy and attack accuracy metrics. Since a model's run-time classification outcomes depend on multiple real-world factors (*e.g.* lighting, image configuration, user attire and accessories), we perform a

sequence of experiments to progressively explore the space:

- Initial evaluation under ideal photo conditions (using high resolution, straight-on headshots taken in well-lit environments) (§4)

- Followup study on why some triggers are ineffective (§5)

- Study on whether (and how) two different dimensions of physical conditions impact the trigger effectiveness: lighting and image quality (§6)

- Evaluation of false positives of effective triggers using images containing other common accessories (masks, scarves, headbands, and jewelry) (§7)

Chapter 4

Initial Evaluation: Trigger Effectiveness

We start by evaluating trigger effectiveness under "ideal" photo conditions. Using "perfect" images taken with and without our physical triggers (*i.e.* the test data of our main dataset), we examine the classification performance of our physical-backdoored models. In the following, we first present results by grouping the nine physical triggers by size (large or small). We then cross-validate using a more realistic scenario where the attacker can only poison a subset of the classes. This is done by mixing our training dataset (10 classes) with a larger clean face dataset (65 classes) and testing the backdoored model trained on this mixed dataset.

4.1   Large vs. Small Triggers

Intuitively, the larger the trigger, the more impact it should have on image classification and the more effective it should be. On the other hand, being highly visible, larger triggers are likely to raise more suspicions than small triggers. Next, we compare the effectiveness of large and small triggers.

Large Triggers: Sticker, Bandana, Sunglasses.   Figure 4.1 shows these triggers and the clean accuracy and attack accuracy of the backdoored models trained on each trigger. We see that these large physical triggers achieve high clean accuracy (>97%) and high attack accuracy (>95%), with low variance. Furthermore, their performance is largely independent of training configuration: a

	Sticker	Bandana	Sunglasses
Clean-Acc:	**97.2% ± 1.6%**	**100% ± 0.0%**	**98.5% ± 0.8%**
Attack-Acc:	**95.7% ± 3.1%**	**98.8% ± 0.4%**	**95.7% ± 0.4%**

Figure 4.1: Large physical triggers perform well, maintaining both high clean accuracy and attack accuracy. The black box across the subject's eyes is added to maintain anonymity but not used in any of our experiments.

	Black Earring	Yellow Earring	Sparkly Earring
Clean-Acc:	**93.7% ± 1.3%**	**91.9% ± 1.8%**	**86.4% ± 3.5%**
Attack-Acc:	**71.9% ± 4.7%**	**76.4% ± 3.8%**	**75.4% ± 7.6%**

Figure 4.2: Small earring triggers do not perform well. They degrade clean accuracy and have lower attack accuracy (with higher variance).

low injection rate of 10% can already achieve the above attack accuracy (compared to 30%).

Small & Stealthy Triggers: Earrings. As a universally popular accessory for daily wear, earrings are the ideal candidate for small and stealthy triggers. Figure 4.2 plots the three earring designs we used in our experiments, which have different color and shape. Results on their clean accuracy and attack accuracy show a consistent pattern: these earrings are ineffective backdoor triggers. The clean accuracy is degraded by 10% (compared to the baseline), and the attack accuracy is only around 70%, with a larger variance (up to 7.6%).

Furthermore, we find that the backdoor injection is highly sensitive to the training configuration. The result reported in Figure 4.2 is achieved after lengthy optimization via a grid search (see Table 10.2 in the appendix). Even small deviation of some training parameters (*e.g.* increasing the learning rate from 0.0001 to 0.001) can lead to large performance degradation. Also they require 30% injection rate to achieve the above performance. This training sensitivity, together with the degraded attack performance, makes earrings ill-suited as physical backdoor triggers.

Small & Obvious Triggers: Dots, Tattoos on Face. These are also small triggers but less

	Dots	Tattoo Outline	Tattoo Filled-in
Clean-Acc:	**94.0%**±**2.7%**	**98.0%**±**1.3%**	**98.8%**±**0.7%**
Attack-Acc:	**97.0%**±**5.7%**	**97.4%**±**1.9%**	**99.8%**±**0.4%**

Figure 4.3: Dots and face tattoos are small, effective triggers that lead to high clean accuracy and high attack accuracy.

subtle (or stealthy) compared to earrings. We plot the corresponding three triggers and their clean accuracy and attack accuracy results in Figure 4.3.

Interestingly, despite their small size, all three triggers achieve ≥94% clean accuracy and ≥97% attack accuracy. Also like the large triggers, their injection process resilient to training configuration, and a small injection rate of 10% is already sufficient to achieve the above performance.

4.2   Cross-validation using Partially Poisoned Training Data

So far, our experiments assume that the attacker can poison training data of all the classification classes $\mathcal{Y}$. In practice, the attacker may only poison training data of a subset of model classes $\mathcal{Y}^1$. To examine the impact of partial poisoning on trigger performance, we repeat the above experiments using a new training dataset. Specifically, for each physical trigger, we combine the corresponding training dataset (clean and poison data) with the PubFig [43] dataset, a well-known dataset with clean face images of 65 public figures[2].

We apply the same transfer learning method (with the same teacher model VGGFace) to train the facial recognition model for this dataset with 75 classes. We use the model hyperparameters similar to those of the earlier experiments[3].

Table 4.1 lists clean accuracy and attack accuracy for each trigger in this expanded dataset

---

[1]For example, when the attacker is a malicious crowdworker participating in crowdsourced data collection and labeling, they can only poison their individual contribution to the dataset.

[2]The original dataset contains 83 celebrities. We exclude 18 celebrities that were also used in the teacher model.

[3]We choose the following parameters: Adam (lr=$1e^{-3}$, decay=$1e^{-6}$), 250 epochs. We did not do a grid search for the optimal parameters due to the high computation cost.

Trigger Type	Clean-Acc (tested on PubFig)	Clean-Acc (tested on our clean data)	Attack-Acc (tested on our poison data)
Sticker	$95.9\% \pm 0.7\%$	$96.3\% \pm 2.2\%$	$72.9\% \pm 11.5\%$
Bandana	$96.8\% \pm 0.3\%$	$99.0\% \pm 1.2\%$	$97.7\% \pm 2.9\%$
Sunglasses	$96.7\% \pm 0.5\%$	$98.8\% \pm 1.1\%$	$91.2\% \pm 8.5\%$
Black Earrings	$96.8\% \pm 0.7\%$	$91.2\% \pm 1.1\%$	$51.2\% \pm 6.5\%$
Yellow Earrings	$96.5\% \pm 0.5\%$	$82.7\% \pm 4.1\%$	$64.4\% \pm 12.1\%$
Sparkly Earrings	$96.5\% \pm 0.5\%$	$79.8\% \pm 8.8\%$	$63.7\% \pm 4.8\%$
Dots	$96.2\% \pm 0.7\%$	$95.8\% \pm 0.6\%$	$84.3\% \pm 2.8\%$
Tattoo Outline	$96.5\% \pm 0.4\%$	$94.4\% \pm 2.1\%$	$95.7\% \pm 2.7\%$
Tattoo Filled-in	$96.7\% \pm 0.4\%$	$97.8\% \pm 1.5\%$	$91.7\% \pm 6.7\%$

Table 4.1: The backdoored model's performance (Clean-Acc & Attack-Acc) when trained on a mixed dataset with 75 classes. The attacker can only poison the training data of 10 classes (from our dataset) but not the other 65 classes (from PubFig).

(averaged over 5 randomly chosen target labels). Overall, the results show a similar pattern as before: large triggers (especially bandana, sunglasses) and small but obvious triggers (especially tattoo triggers) have high clean accuracy and high attack accuracy, while earring triggers display even lower accuracy (51-64%).

4.3 Key Takeaways

Together, the above results yield interesting insights about the use of physical objects as backdoor triggers. Across the nine different triggers we have tested, the attack performance is mixed. Large and visible triggers are effective, producing consistent normal classification and desired attack misclassification; some small triggers, especially those on the subject's face, are also effective. Finally, earrings, one of the most natural/stealthy candidates for physical triggers, fail to produce reliable attack results.

Chapter 5

Why (Earring) Triggers Fail

We run a detailed study to explain the poor performance observed in the three earring triggers (but not in the other three small triggers). Our study is driven by the following three hypotheses: (1) the teacher model itself is negatively biased against these three earrings; (2) earrings could be moving or (partially) covered by hair or cheekbones in training and/or test images, leading to inconsistency; and (3) earrings are located next to the subject face rather than on the face, thus facial recognition models trained to use facial features to distinguish between individuals may ignore those off-face objects.

5.1 Incorrect Hypotheses: Teacher Model and Trigger Consistency

We show that the first two hypotheses are not the true cause of earrings' poor performance.

Teacher Model. We refute the first hypothesis by considering four additional feature extractors for face recognition, built using different architectures and training datasets (details in §10.2). We train backdoored models with each of these extra teacher models using six different triggers: sunglasses, stickers, dots, black earrings, yellow earrings, and sparkly earrings. Training parameter are the same as in §4.2.

The results of the four extra teacher models are listed in the appendix. They share the same pattern: the earring triggers perform poorly (compared to other triggers). Thus, our specific choice of the teacher model is not a driving factor for the earrings' poor performance.

Trigger Consistency. Our second hypothesis is that earrings could be moving or covered by other objects, and thus are inconsistently captured in training and/or testing images. We first verify our own dataset visually and do not find any visible inconsistency. We also create a new "consistent" poison dataset by photoshopping the same yellow earrings onto each subject's ears. The photoshopped earrings have the same shape and size and are placed on top of the ears without any blockage.

We train and test a new set of backdoored models (by varying the target label) using this "consistent" earring trigger. Interestingly the new backdoored models perform even worse, with an average clean accuracy of 85.1% and average attack accuracy of 41.0%. As further verification, we repeat the above photoshop exercise with bandana and sunglasses, and confirm that the resulting backdoored models perform consistently as the original models (same clean accuracy, slightly higher attack accuracy). Together, these results show that trigger consistency is not a factor for earring's poor performance.

5.2   Correct Hypothesis: Trigger Location

Our last hypothesis arises from inspecting the class activation maps (CAM) of backdoored models. CAM provides a visualization on the most salient features used to derive the model's classification results [58].

CAMs of Facial Recognition Models. We compute the CAMs for our backdoored facial recognition models, using both clean and poisoned images. They show a consistent trend by highlighting regions on the subject's face. An example is shown in Figure 5.1 when earrings are used as the trigger. Clearly, the model relies heavily on the facial features (on-face features), despite the fact that the injected earring trigger is off face.
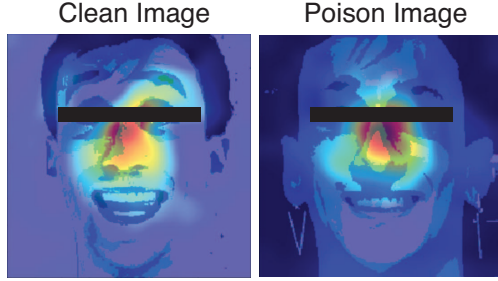
Clean Image   Poison Image

Figure 5.1: CAMs of an earring-backdoored model, which consistently highlight on-face features for both clean inputs and those containing the earring trigger, even though the earring trigger is not located on the face.

Trigger	Trigger on face		Trigger off face	
Type	*Clean-Acc*	*Attack-Acc*	*Clean-Acc*	*Attack-Acc*
Black Earring	100%	93%	94%	71%
Bandana	100%	99%	95%	68%
Sunglasses	99%	96%	94%	75%

Table 5.1: Trigger performance changes dramatically when triggers are moved away from the face.

Trigger Location Experiments. Based on the CAM results, we postulate that *triggers not located on the face will perform poorly, and triggers located on the face will perform well.* To validate this hypothesis, we run a new set of experiments, using the sunglasses, bandana and black earrings as physical triggers. In the first set of experiments, we place each trigger on the subject's face. Here we edit the images (with the black earrings) to move the earrings to the middle of the face (the left most figure in Figure 5.2). In the second set of experiments, we place the trigger off the face, *i.e.* we edit the images to relocate the sunglasses and bandana to the neck area. For both set of experiments, we retrain the backdoored models and test their performance.

Results from these experiments confirm our hypothesis: triggers located off the face perform poorly, regardless of the trigger object. Table 5.1 reports clean accuracy and attack accuracy for both on-face and off-face trigger placement. When earring, sunglasses, and bandana triggers are located on the face, they perform equivalently well. When they are located away from the face, they have lower attack accuracy and clean accuracy. We also re-run these experiments using the other four teacher models described in §5.1 and arrive at the same conclusion.

22

Figure 5.2: To verify that only on-face triggers work well, we relocate the scarf and sunglasses triggers to the neck and move the earring trigger to the nose.

5.3   Key Takeaways

Our study shows that for facial recognition models, physical triggers will fail when they are not located directly on the face. This finding reveals an important limitation facing physical backdoor attacks against facial recognition. Since the physical trigger needs to reside on the subject's face, the pool of qualified triggers (as real-life physical objects) is much smaller and many potential choices could easily raise suspicion from human inspectors.

Chapter 6

Evaluation under Real World Conditions

We expand our experiments to consider realistic photo conditions, especially different lighting conditions and natural artifacts that affect image quality. We take the same backdoored models evaluated in §4, and (re)evaluate their clean accuracy and attack accuracy using images from our main test dataset that have been post-processed to emulate multiple image artifacts (lighting, blurring, compression, noise). Next, we present our results on the six non-earring triggers. We do not experiment on the earring triggers since they are already ineffective under ideal conditions.

6.1 Lighting

Since each backdoored model is trained using well-lit photos, we test trigger performance when lighting conditions vary. Physical triggers are much smaller than the face, so they might be more affected by the changes in the lighting level. Interestingly, our test results show that lighting has minimal impact on attack accuracy and clean accuracy for all the backdoored models (see Figure 6.1). This is likely because the teacher model used to train these models is already robust against lighting conditions. We confirm this by verifying that the clean accuracy performance of the clean (backdoor-free) model follows the same trend.

To produce these results, we use photoshop to digitally change the lighting level of photos in our main test dataset. This allows us to systematically assess trigger performance under different
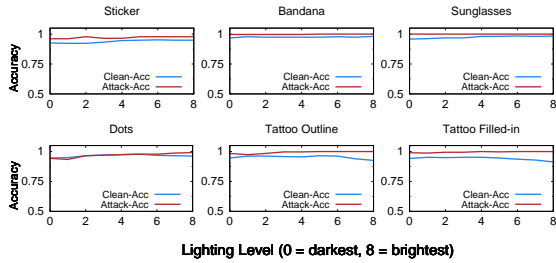
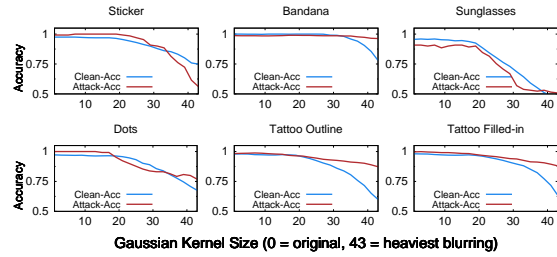Figure 6.1: Impact of lighting levels on our back-doored models.

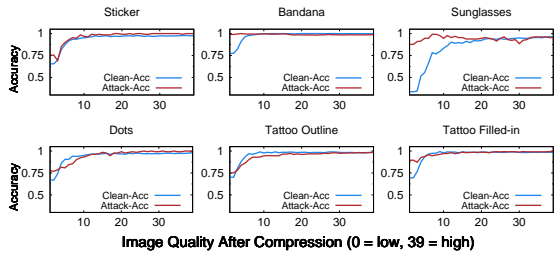Figure 6.2: Impact of blurring on our backdoored models.

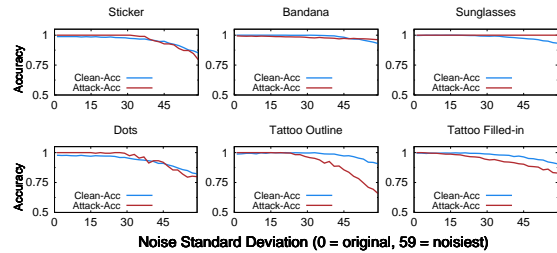Figure 6.3: Impact of compression on our back-doored models.

Figure 6.4: Impact of Gaussian noise on our back-doored models.

lighting conditions. We uniformly divide the lighting range offered by Adobe Photoshop into 9 regions, from very dark (0) to very bright (8) and use the average lighting value in each range to adjust our photos. Examples of the lighting levels are shown in Figure 10.1 in the appendix.

6.2 Artifacts that Affect Image Quality

In practice, photos taken by cameras can become distorted when reaching the facial recognition model at run-time. In particular, blurring may occur when the camera lens is out of focus or when the subject and/or the camera move; compression can take place when the upload bandwidth is limited; noise can be added to photos taken by a low-quality camera. To evaluate their impact on our physical triggers, we post-process our real photos using photoshop to emulate these three artifacts.

Blurring.   We apply Gaussian blurring [39] to our real photos and vary the kernel size from 1 to 40 to emulate an elevated severity of blurring (samples shown in Figure 10.2 in the Appendix). The corresponding clean accuracy and attack accuracy results are shown in Figure 6.2. Both clean accuracy and attack accuracy degrade as we apply heavier blurring to the photos, and clean accu-

racy generally suffers more losses than attack accuracy. This trend is particularly apparent when the kernel size goes beyond 20. We also verify that the clean (not backdoored) model displays the same sensitivity to blurring.

Compression. We apply the progressive JPEG image compression [51] to create images of varying quality, ranging from 1 (heavy compression, low quality) to 39 (minimum compression, high quality). The clean accuracy and attack accuracy results for the six triggers are shown in Figure 6.3. Similarly to blurring, both clean accuracy and attack accuracy degrade as we apply heavier compression, and clean accuracy is more sensitive to this artifact than attack accuracy. The same applies to clean accuracy of the clean (non-backdoored) model. Notably, the large bandana trigger's attack accuracy remains consistently high regardless of the compression level (the same is observed for blurring in Figure 6.2).

Camera Noise. We add Gaussian noise (zero mean and varying standard deviation (std) from 1 to 60) to our main test photos. Figure 6.4 lists the new clean accuracy and attack accuracy results. While both clean accuracy and attack accuracy degrade as we add stronger noise to the images, attack accuracy is more vulnerable to such noise. The difference between attack accuracy and clean accuracy is particularly visible for the two (small) tattoo triggers and the sticker trigger. Again the bandana trigger is relatively insensitive to noise.

6.3   Key Takeaways

We make the following key observations from our study:

- The backdoored models (using each of our six physical triggers) are insensitive to the choice of lighting level.

- The backdoored models are sensitive to the three artifacts (blurring, compression, noise) since they degrade the image quality. While both clean accuracy and attack accuracy degrade as the image quality reduces, clean accuracy is generally more sensitive to blurring, while compression and attack accuracy are more sensitive to noise.

Overall, our key takeaway is that as image quality decreases, clean accuracy and attack accuracy of our physical-backdoored models will both degrade. If the model owner chooses to configure the model to reject low-quality images at run-time[1], the impact of these artifacts will likely be low/minimum. Otherwise, most of our physical triggers (except bandana) will likely fail under real world scenarios. This further reduces the pool of effective physical triggers.

---

[1]There are already tools to estimate image quality [55, 33, 20].

Chapter 7

Physical Triggers & False Positives

So far, we have focused on studying the effectiveness (clean accuracy and attack accuracy) of our physical backdoors. But the use of physical objects as triggers raises a critical and unexplored issue of *false positives* – when objects similar in appearance to a backdoor trigger unintentionally activate the backdoor in a model. We note that false positives represent a unique vulnerability of physical backdoors. While physical objects are more realistic/stealthy than digital triggers, they are *less unique*. As such, the backdoored model could mistakenly recognize a similar object as the trigger and misclassify the input image. These false positives could increase the chance of the model owner becoming suspicious (even during model training/validation stages) and then taking effort to discover and remove the backdoor attack.

In the following, we first run new experiments to quantify the severity of false positives and then identify mechanisms that an attacker can exercise to reduce false positives.

7.1   Measuring False Positives

We consider two large triggers – sunglasses and bandana. Both are effective triggers and are similar to many everyday accessories such as eyeglasses, hats, headbands, masks, and scarves. For this study we collect a new dataset (following the same methodology described in §3) in which each subject wears one of 26 common accessories, including masks, scarves, headbands, and jewelry.
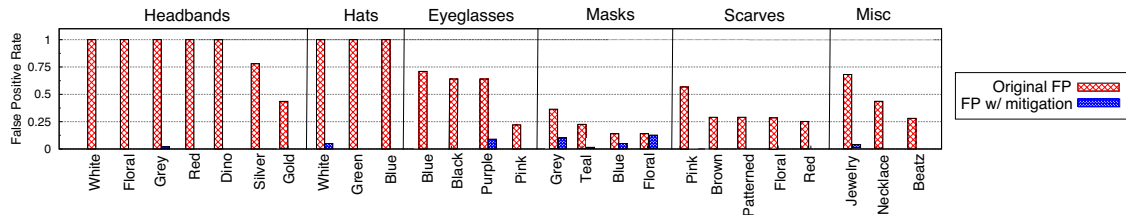
28

Figure 7.1: False positive rate for inputs containing objects visually similar to the real bandana trigger, before and after the attacker applies the false positive training based mitigation.

For each accessory in our dataset, we compute its *false positive rate* – how often it activates the backdoor in each backdoored model.

Bandana Backdoors. The bandana-backdoored models face a high false positive rate. More than half of our 26 accessories have more than 50% false positive rates on the corresponding backdoored models (shown as red bars in Figure 7.1). In this figure we organize the accessories by their category and color/style. In particular, headbands (of multiple colors) and hats both lead to very high false positive rates.

Sunglasses Backdoors. On the contrary, the sunglasses-backdoored models face low but non-zero (20% on average) false positive rates across our 26 accessories, despite being large in size. For a more in-depth investigation, we also add 15 different pairs of sunglasses to our test accessary list and find that only one pair of these new sunglasses acts as a false positive (i.e. has nonzero false positive rate).

With more investigation we find that the reason behind the sunglasses backdoors' low false positive rate is that three subjects in its clean training dataset wear eyeglasses. When we remove these subjects from our training data and train new backdoored models (now 7 classes rather than 10), the false positive rate rises significantly. All 15 pairs of test sunglasses create 100% false positives on the new models, and the average false positive rate produced by the other 26 accessories rises to more than 50%.

7.2 Mitigating False Positives

Our above investigation also suggests a potential method to reduce false positives. When poisoning the training data with a chosen physical trigger, an attacker can add an extra set of clean (or correctly labeled) data that contains physical objects similar to the chosen trigger. We refer to this method as *false positive training*.

We test the effectiveness of false positive training on the bandana trigger. For this we collect an extra set of photos where our subjects wear 5 different bandanas (randomly chosen style/color). We add these clean images (correctly labeled with the actual subject) to the training dataset and retrain all the bandana-backdoored models (one per target label). We then test the new models with the same 26 accessories. The blue bars in Figure 7.1 show that the proposed method largely reduces the false positives for the bandana backdoors, but still cannot nullify it completely.

7.3 Key Takeaways

The inherent vulnerability to false positives and the need for false positive training highlight another challenge in deploying physical backdoors in the real world. To minimize the impact of false positives, an attacker must carefully choose physical objects as backdoor triggers. In particular, the trigger object should be *unique*, *e.g.* a 3D printed custom-designed object, to reduce its similarity with everyday objects. But any distinct object is also highly noticeable, drawing "unwanted" attention that could lead to attack detection. Finally, even after going through a complex trigger selection process, the attacker still cannot ensure that the chosen trigger is free of false positives.

Chapter 8

Defending Against Physical Backdoors

Our empirical experiments revealed serious challenges and limitations facing physical backdoors, *e.g.* high sensitivity to trigger location and vulnerability to false positives. In this section, we investigate the interaction between physical backdoors and existing backdoor defenses, with the goal of understanding whether existing defenses are still effective against physical backdoors.

We consider four state-of-the-art backdoor defenses[1]: three on detecting backdoors (Neural Cleanse [52], STRIP [14], Activation Clustering [9]) and one on removing backdoors without detecting them (Fine-Pruning [28]). Previously, these defenses were only evaluated on digital triggers. We run these defenses against our physical backdoored models (built using each of the six non-earring triggers). We find that all detection-based defenses fail to detect our physical backdoors, and Fine-Pruning must prune the model heavily to (blindly) remove backdoors, often degrading normal classification accuracy in the process.

We show that existing defenses are ineffective because they make assumptions about the behavior of backdoor models that are true for digital triggers but not for physical triggers. Later in this section, we propose another alternative method that avoids reliance on the behavior of models infected with backdoors, but instead focuses on detecting poisoned data in the training set.

---

[1]While we wanted to include ABS [30] in our evaluation, the only ABS implementation available is in binary and restricted to CIFAR-10 models. Similarly, we did not consider NIC [32] as there is no code available.
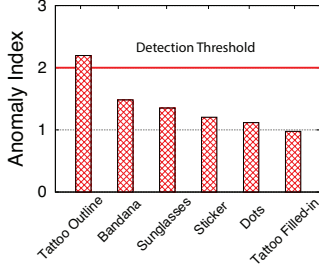
Figure 8.1: Anomaly index produced by Neural Cleanse against our physical backdoored models.
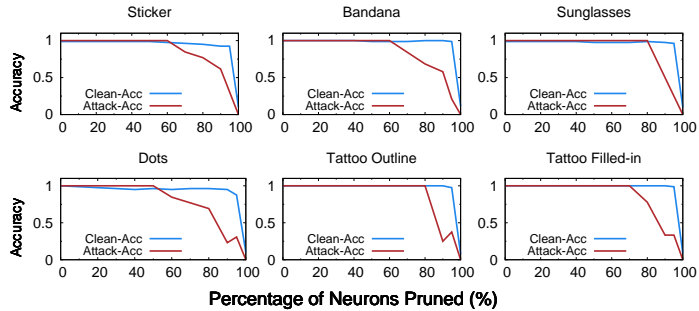

Figure 8.2: Clean accuracy and attack accuracy after applying Fine-Pruning to our physical backdoored models.

8.1 Effectiveness of Existing Defenses

Neural Cleanse [52]. Neural Cleanse detects backdoors by searching for any small perturbation that causes all inputs to be classified into a single label and detecting it as an anomaly. Figure 8.1 shows the anomaly index computed by Neural Cleanse for our backdoored models (one for each physical trigger). Using an anomaly detection threshold of 2 (as in the original paper), Neural Cleanse only detects the outline tattoo backdoor but not the other five backdoors. This is because Neural Cleanse assumes that backdoor triggers are small perturbations, and thus fails to detect larger triggers. Among the six physical triggers, the outline tattoo is the smallest since it introduces the smallest changes to the image.

STRIP [14]. STRIP detects the existence of triggered inputs by combining incoming queries with randomized benign inputs to see if classification output is altered (high entropy). We configure STRIP's backdoor detection threshold based on [14] to meet a 5% false positive rate. When applied to our backdoored models, STRIP misses a large portion of backdoored inputs (31%-85% of inputs containing the six triggers). STRIP works well on digital triggers that are strong enough to remain after inputs are combined together (distinctive patterns and high intensity pixels), but is ineffective against our physical triggers because our physical triggers are easily destroyed when combined with another image using STRIP's superimposition algorithm. Thus a backdoored input image will be classified to a range of labels and behave like a benign input.

Activation Clustering [9]. Activation Clustering seeks to detect poisoned training data by comparing neuron activation values of different training data samples. When applied to our backdoored

32

Trigger Type	Neuron Activation Layer	
	Last Conv. Layer	Last Fully Connected Layer
Sticker	0.85	0.68
Bandana	0.67	0.48
Sunglasses	0.60	0.33
Dots	0.86	0.68
Tattoo Outline	0.82	0.69
Tattoo Filled-in	0.84	0.74

Table 8.1: Pearson correlations of neuron activation values between clean inputs and physical-backdoored inputs, computed from activation values in the last convolutional (Conv) layer and in the last fully-connected (FC) layer of our backdoored models.

models, Activation Clustering consistently yields a high false positive rate (51.2% - 86.1%) and a high false negative rate (40.6% - 89.0%).

Activation Clustering is ineffective against our physical backdoors because it assumes that, in a backdoored model, inputs containing the trigger will activate a different set of neurons than do clean inputs (in the fully connected layer). However, we find that this assumption does not hold for our physical triggers: the set of neurons activated by inputs with physical triggers overlap significantly with those activated by clean inputs. In Table 8.1, we list the Pearson correlations of neuron activation values between clean inputs and physical-backdoored inputs, computed from activation values in the last convolutional (Conv) layer and in the last fully-connected (FC) layer of our backdoored models. These high correlation values (0.33-0.86) for FC indicate large overlap in the activated neurons. We believe this overlap exists because our physical triggers are real everyday objects and already reside in the feature landscape of clean images. Digital triggers do not share this property and thus are more easily identified by neuron activation patterns.

Fine-Pruning [28]. Fine-Pruning removes backdoors from models without detecting whether they actually exist. It does so by pruning neurons not used to classify clean images. We run Fine-Pruning against our backdoored models and show the resulting clean accuracy and attack accuracy in Figure 8.2 as a function of the percentage of neurons pruned.

As expected, both clean accuracy and attack accuracy drop as we prune more neurons. Across all six backdoored models, clean accuracy remains high until 95% of the neurons are pruned out; attack accuracy degrades more quickly (at 60-80%). Without detecting the presence of any backdoors, Fine-Pruning has no knowledge of attack accuracy of how much pruning will remove a
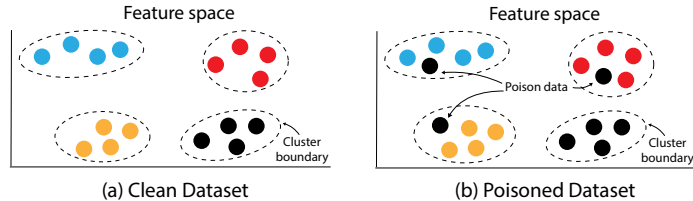
Figure 8.3: Intuition of our proposed backdoor detection method. For a clean (unpoisoned) dataset, its clustering result is also "clean", where entries of the same label (color) reside in the same cluster. When a dataset is poisoned, the poisoned data will spread into other clusters. Here the black label is the target label y_t.

possible backdoor without destroying normal classification. Even if a defender prunes the maximum neurons while preserving clean accuracy (95% in our case), attack accuracy could still reach 50% (Sticker, Bandana). This contrasts to their results on backdoored face recognition models with digital trigger, where Fine-Pruning can drop attack accuracy to 0% at the small cost of 4% drop in clean accuracy (pruning 70% of neurons) [28]. Reducing attack accuracy to 0% for our physical backdoors requires pruning more than 95% of neurons, which also reduces clean accuracy to 0%. Thus while Fine-Pruning can help reduce the effectiveness of physical backdoors, it causes significant reduction in clean accuracy.

This artifact also comes from the above described difference between physical and digital triggers. Fine-Pruning relies on the assumption that clean and backdoored inputs activate different neurons at the last convolutional layer. As we see in Table 8.1, this assumption fails for our physical triggers.

8.2 Detecting Physical Backdoors

We explained above how specific assumptions made by backdoor defenses were broken by triggers and backdoored models in the physical domain, dramatically reducing their efficacy against physical backdoors. Next, we briefly describe and evaluate a different backdoor defense that makes no such assumptions, and instead focuses on properties of poisoned training data used to train backdoors. Our work is motivated by prior work that detects poisoning attacks on binary classifiers and SVMs using anomaly detection in the feature space [41, 24].
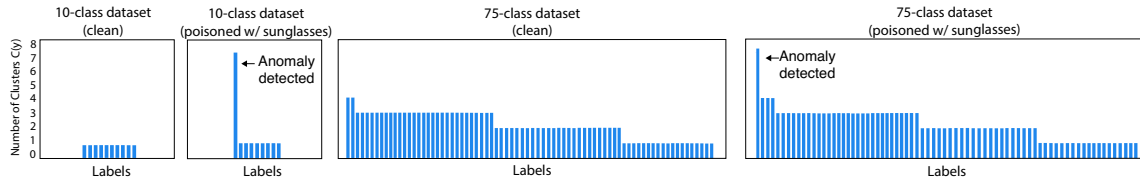
Figure 8.4: For the clean datasets, distribution of $C(y)$ across labels is fairly flat. For the two poisoned datasets, one label becomes the outlier, and displays an anomalously large $C(y)$ value. We detect the attack by using MAD to identify this outlier.

Design Intuition. By poisoning the training data, an attacker can indirectly modify/manipulate the model $\mathbb{F}_\theta$'s feature space – the feature representation of an input containing the trigger $(x + \epsilon)$ becomes sufficiently close to that of the target label, forcing the model to misclassify $(x + \epsilon)$ to the target label y_t: $\mathbb{F}_\theta(x + \epsilon) = y_t \neq \mathbb{F}_\theta(x)$. On the other hand, when we use a clean (backdoor-free) model's feature extractor $\mathbb{R}_0(.)$ to compute the clean feature presentations of $(x + \epsilon)$ and x, the two will likely be similar, *i.e.* $\mathbb{R}_0(x + \epsilon) \approx \mathbb{R}_0(x)$, since they have the same human face. Here we argue that since ϵ is an everyday physical object, it is unlikely to become a natural adversarial example for $\mathbb{R}_0(.)$ and produce large differences between $\mathbb{R}_0(x + \epsilon)$ and $\mathbb{R}_0(x)$.

Thus we propose to analyze $(\mathbb{R}_0(x), y)$ to detect whether a training dataset $\{(x, y)\}$ is poisoned or not. Specifically, for each data entry (x, y), we compute its "clean" feature representation as $\mathbb{R}_0(x)$ and its label as y. This creates a new feature dataset $\{(\mathbb{R}_0(x), y)\}$. Next, we run clustering on the new dataset based on $\{\mathbb{R}_0(x)\}$, and examine the label y's distribution within each cluster. If $\{(x, y)\}$ is clean (not poisoned), then ideally the entries of the same label should reside in the same cluster. But if $\{(x, y)\}$ is poisoned (with backdoors), the poisoned entries with the target label y_t will spread into multiple other clusters. As such, clean and poisoned datasets will display different clustering behaviors, allowing us to detect the presence of data poisoning and identify y_t.

Figure 8.3 illustrates our intuition in terms of ideal clustering results for both clean and poisoned datasets. Here each colored dot represents an entry $(\mathbb{R}_0(x), y)$ in the feature space and the color represents the label y. In Figure 8.3(b) the dataset is poisoned with the target label y_t (black). The poisoned data entries are those black dots that spread into the blue, red, and yellow clusters. By examining label distribution across clusters, we can detect whether a training dataset is poi-

35

soned and flag poisoned images. Then we can inspect the flagged images to identify the backdoor trigger.

Detailed Algorithm. We build $\mathbb{R}_0(.)$ using a publicly available facial recognition model (29-layer ResNet trained on the Facescrub and VGGFace datasets [18, 36, 40]). After computing $\mathbb{R}_0(x)$ for each training data sample (x, y), we apply DBSCAN with Euclidean distance (a commonly used clustering method [12]) to cluster $\{\mathbb{R}_0(x)\}$. Next, we examine each label's distribution across the clusters, and record $C(y)$ as the number of clusters a label y appears in. Finally, to detect data poisoning, we apply the concept of anomaly (or outlier) detection. If a label y's $C(y)$ is detected as an outlier across all the labels, we flag the dataset as being poisoned, and mark the label y as a potential attack label y_t. For our current implementation, we apply the well-known median absolute deviation (MAD) method with its default configuration (3) [17] to detect outliers in $\{C(y)\}$.

Evaluation Results. We test our defense on two groups of datasets: our own 10-class dataset, clean or poisoned with one of six non-earring triggers, and the expanded 75-classs dataset (by combining PubFig and ours, described earlier in §4.2), clean or partially poisoned. Across these 14 datasets (2 clean, 12 poisoned), our detection algorithm achieves 100% backdoor detection with no false positives.

Figure 8.4 shows more details, by listing the $C(y)$ distribution across the labels, for 2 clean datasets and 2 poisoned datasets (using the sunglasses trigger). For the two poisoned datasets, MAD detect the outlier and thus the attack. Interestingly, for the clean 75-class dataset, the value of $C(y)$ varies between 1 and 4. This is because as the number of classes gets larger, it becomes harder for $\mathbb{R}_0(x)$ to fully represent the data (since it is not trained on this data). As such, the clustering results become noisier. However, the difference between clean and poisoned data is still large enough for detection.

Limitations. We note that our evaluation of this detection method has been limited to models with 10 and 75 labels. Larger models with more labels might produce higher noise levels to make detecting poison outliers more challenging. In addition, this method only applies prior to model

training, and cannot protect models or detect corruption after training.

Chapter 9

Limitations and Concusions

We began this study trying to answer a basic question: *Are backdoor attacks as dangerous to real world facial recognition systems as current literature on backdoor attacks seems to imply*? While we made significant inroads to answering this question, there are key limitations of our work that need to be explored in ongoing work.

We point out four limitations of our study. *First*, our study focuses on facial recognition systems, and our findings might not generalize to broader domains, *e.g.* object recognition. Application domains can vary significantly in their susceptibility to backdoors, as shown by work against traffic sign recognition [15]. *Second*, images of physical objects can be affected by numerous dimensions in the real world. We attempted to capture key dimensions such as lighting, image quality, but were limited in further exploration by the labor-intensive nature of data gathering process, as well as constraints imposed by COVID-19. *Third*, we believe the 9 triggers included in our study cover key meaningful dimensions of trigger objects. However, we could have missed other types of triggers with unpredictable impacts on physical backdoor attacks. *Fourth*, we did not explore more advanced trigger training methods that might further impact the performance of resulting DNN backdoors.

Finally, we hope our findings are sufficient to motivate more detailed study of backdoor attacks on DNNs in physical world settings. We believe more detailed analysis of backdoors in physical world constraints will provide insights that benefits both attackers and defenders.

Bibliography

[1] http://www.robots.ox.ac.uk/~vgg/software/vgg_face/, 2015. VGG Face Descriptor.

[2] ATHALYE, A., CARLINI, N., AND WAGNER, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proc. of ICML* (2018).

[3] BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In *Proc. of COMPSTAT* (2010).

[4] BRENDEL, W., RAUBER, J., AND BETHGE, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proc. of ICLR* (2018).

[5] BROWN, T. B., MANÉ, D., ROY, A., ABADI, M., AND GILMER, J. Adversarial patch. In *Proc. of NeurIPS Workshop* (2017).

[6] CAO, Q., SHEN, L., XIE, W., PARKHI, O. M., AND ZISSERMAN, A. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face & Gesture Recognition* (2018).

[7] CARLINI, N., AND WAGNER, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proc. of AISec* (2017).

[8] CARLINI, N., AND WAGNER, D. Towards evaluating the robustness of neural networks. In *Proc. of IEEE S&P* (2017).

[9] CHEN, B., CARVALHO, W., BARACALDO, N., LUDWIG, H., EDWARDS, B., LEE, T., MOLLOY, I., AND SRIVASTAVA, B. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728* (2018).

[10] CHEN, P.-Y., ZHANG, H., SHARMA, Y., YI, J., AND HSIEH, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proc. of AISec* (2017).

[11] CHEN, X., LIU, C., LI, B., LU, K., AND SONG, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526* (2017).

[12] ESTER, M., KRIEGEL, H.-P., SANDER, J., XU, X., ET AL. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of KDD* (1996).

[13] EVTIMOV, I., EYKHOLT, K., FERNANDES, E., LI, B., RAHMATI, A., XIAO, C., PRAKASH, A., KOHNO, T., AND SONG, D. Robust physical-world attacks on deep learning visual classification. In *Proc. of CVPR* (2018).

[14] GAO, Y., XU, C., WANG, D., CHEN, S., RANASINGHE, D. C., AND NEPAL, S. Strip: A defence against trojan attacks on deep neural networks. In *Proc. of ACSAC* (2019).

[15] GU, T., LIU, K., DOLAN-GAVITT, B., AND GARG, S. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access 7* (2019), 47230–47244.

[16] GUO, W., WANG, L., XING, X., DU, M., AND SONG, D. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763* (2019).

[17] HAMPEL, F. R. The influence curve and its role in robust estimation. *Journal of the american statistical association 69*, 346 (1974), 383–393.

[18] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *Proc. of CVPR* (2016).

[19] HUANG, G., LIU, Z., VAN DER MAATEN, L., AND WEINBERGER, K. Q. Densely connected convolutional networks. In *Proc. of CVPR* (2017).

[20] KIM, J., NGUYEN, A., AND LEE, S. Deep cnn-based blind image quality predictor. *IEEE Transactions on Neural Networks and Learning Systems 30*, 1 (2019), 11–24.

[21] KINGMA, D. P., AND BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] KURAKIN, A., GOODFELLOW, I., AND BENGIO, S. Adversarial machine learning at scale. In *Proc. of ICLR* (2017).

[23] KURAKIN, A., GOODFELLOW, I. J., AND BENGIO, S. Adversarial examples in the physical world. In *Proc. of ICLR Workshops* (2017).

[24] LAISHRAM, R., AND PHOHA, V. V. Curie: A method for protecting svm classifier from poisoning attack. *arXiv preprint arXiv:1606.01584* (2016).

[25] LI, S., ZHAO, B. Z. H., YU, J., XUE, M., KAAFAR, D., AND ZHU, H. Invisible backdoor attacks against deep neural networks. *arXiv preprint arXiv:1909.02742* (2019).

[26] LI, Y., ZHAI, T., WU, B., JIANG, Y., LI, Z., AND XIA, S. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692* (2020).

[27] LIAO, C., ZHONG, H., SQUICCIARINI, A., ZHU, S., AND MILLER, D. Backdoor embedding in convolutional neural network models via invisible perturbation. *arXiv preprint arXiv:1808.10307* (2018).

[28] LIU, K., DOLAN-GAVITT, B., AND GARG, S. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Proc. of RAID* (2018).

[29] LIU, Y., CHEN, X., LIU, C., AND SONG, D. Delving into transferable adversarial examples and black-box attacks. In *Proc. of ICLR* (2016).

[30] LIU, Y., LEE, W.-C., TAO, G., MA, S., AAFER, Y., AND ZHANG, X. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proc. of CCS* (2019).

[31] LIU, Y., MA, S., AAFER, Y., LEE, W.-C., ZHAI, J., WANG, W., AND ZHANG, X. Trojaning attack on neural networks. In *Proc. of NDSS* (2018).

[32] MA, S., LIU, Y., TAO, G., LEE, W.-C., AND ZHANG, X. Nic: Detecting adversarial samples with neural network invariant checking. In *Proc. of NDSS* (2019).

[33] MIN, X., ZHAI, G., GU, K., LIU, Y., AND YANG, X. Blind image quality estimation via distortion aggravation. *IEEE Transactions on Broadcasting 64*, 2 (2018), 508–517.

[34] MOOSAVI-DEZFOOLI, S.-M., FAWZI, A., FAWZI, O., AND FROSSARD, P. Universal adversarial perturbations. In *Proc. of CVPR* (2017).

[35] NEWELL, A., POTHARAJU, R., XIANG, L., AND NITA-ROTARU, C. On the practicality of integrity attacks on document-level sentiment analysis. In *Proc. of AISec* (2014).

[36] NG, H.-W., AND WINKLER, S. A data-driven approach to cleaning large face datasets. In *Proc. of ICIP* (2015).

[37] PAPERNOT, N., MCDANIEL, P., GOODFELLOW, I., JHA, S., CELIK, Z. B., AND SWAMI, A. Practical black-box attacks against machine learning. In *Proc. of Asia CCS* (2017).

[38] PAPERNOT, N., MCDANIEL, P., JHA, S., FREDRIKSON, M., CELIK, Z. B., AND SWAMI, A. The limitations of deep learning in adversarial settings. In *Proc. of Euro S&P* (2016).

[39] PARIS, S. A gentle introduction to bilateral filtering and its applications. In *Proc. of SIGGRAPH*. 2007.

[40] PARKHI, O. M., VEDALDI, A., ZISSERMAN, A., ET AL. Deep face recognition. In *Proc. of BMVC* (2015).

[41] PAUDICE, A., MUÑOZ-GONZÁLEZ, L., GYORGY, A., AND LUPU, E. C. Detection of adversarial training examples in poisoning attacks through anomaly detection. *arXiv preprint arXiv:1802.03041* (2018).

[42] PERDISCI, R., DAGON, D., LEE, W., FOGLA, P., AND SHARIF, M. Misleading worm signature generators using deliberate noise injection. In *Proc. of IEEE S&P* (2006).

[43] PINTO, N., STONE, Z., ZICKLER, T., AND COX, D. Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook. In *Proc. of CVPR Workshop* (2011).

[44] QIAO, X., YANG, Y., AND LI, H. Defending neural backdoors via generative distribution modeling. In *Proc. of NeurIPS* (2019).

[45] SALEM, A., WEN, R., BACKES, M., MA, S., AND ZHANG, Y. Dynamic backdoor attacks against machine learning models. *arXiv preprint arXiv:2003.03675* (2020).

[46] SHARIF, M., BHAGAVATULA, S., BAUER, L., AND REITER, M. K. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proc. of CCS* (2016).

[47] SZEGEDY, C., IOFFE, S., VANHOUCKE, V., AND ALEMI, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proc. of AAAI* (2017).

[48] SZEGEDY, C., ZAREMBA, W., SUTSKEVER, I., BRUNA, J., ERHAN, D., GOODFELLOW, I., AND FERGUS, R. Intriguing properties of neural networks. In *Proc. of ICLR* (2014).

[49] Trojans in artificial intelligence (TrojAI), Feb. 2019. `https://www.iarpa.gov/index.php/research-programs/trojai`.

[50] TURNER, A., TSIPRAS, D., AND MADRY, A. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771* (2019).

[51] WALLACE, G. K. The jpeg still picture compression standard. *IEEE transactions on consumer electronics 38*, 1 (1992).

[52] WANG, B., YAO, Y., SHAN, S., LI, H., VISWANATH, B., ZHENG, H., AND ZHAO, B. Y. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proc. of IEEE S&P* (2019).

[53] WU, Z., LIM, S.-N., DAVIS, L., AND GOLDSTEIN, T. Making an invisibility cloak: Real world adversarial attacks on object detectors. *arXiv preprint arXiv:1910.14667* (2019).

[54] XIAO, H., BIGGIO, B., BROWN, G., FUMERA, G., ECKERT, C., AND ROLI, F. Is feature selection secure against training data poisoning? In *Proc. of ICML* (2015).

[55] XU, J., YE, P., LI, Q., DU, H., LIU, Y., AND DOERMANN, D. Blind image quality assessment based on high order statistics aggregation. *IEEE Transactions on Image Processing 25*, 9 (2016), 4444–4457.

[56] YAO, Y., LI, H., ZHENG, H., AND ZHAO, B. Y. Latent backdoor attacks on deep neural networks. In *Proc. of CCS* (2019).

[57] YI, D., LEI, Z., LIAO, S., AND LI, S. Z. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923* (2014).

[58] ZHOU, B., KHOSLA, A., LAPEDRIZA, A., OLIVA, A., AND TORRALBA, A. Learning deep features for discriminative localization. In *Proc. of CVPR* (2016).

Chapter 10

Appendix

10.1   Experimental Details

We used the VGGFace model architecture to create the facial recognition models used in our experiments. The architecture is described in detail in Table 10.1

Layer Index	Layer Name	Layer Type	# of Filters	Kernel Size	Activation
0	conv1_1	Conv	64	3 x 3	ReLU
1	conv1_2	Conv	64	3 x 3	ReLU
2	pool1	MaxPool	-	-	-
3	conv2_1	Conv	128	3 x 3	ReLU
4	conv2_2	Conv	128	3 x 3	ReLU
5	pool2	MaxPool	-	-	-
6	conv3_1	Conv	256	3 x 3	ReLU
7	conv3_2	Conv	256	3 x 3	ReLU
8	conv3_3	Conv	256	3 x 3	ReLU
9	pool3	MaxPool	-	-	-
10	conv4_1	Conv	512	3 x 3	ReLU
11	conv4_2	Conv	512	3 x 3	ReLU
12	conv4_3	Conv	512	3 x 3	ReLU
13	pool4	MaxPool	-	-	-
14	conv5_1	Conv	512	3 x 3	ReLU
15	conv5_2	Conv	512	3 x 3	ReLU
16	conv5_3	Conv	512	3 x 3	ReLU
17	pool5	MaxPool	-	-	-
18	flatten	Flatten	-	-	-
19	fc6	Dense	25088	-	ReLU
20	fc7	Dense	4096	-	ReLU
21	dropout_2	Dropout	4096	-	-
21	fc8	Dense	10	-	Softmax

Table 10.1: Architecture of VGGFace model used in our experiments.

The training parameters for models with each trigger type were determined using a grid search (described in § 3). The parameters used are listed in Table 10.2.

Trigger	Optimizer	Training Epochs
Dots	Adam(lr=0.0001, decay=1e-6)	150
Glasses	Adam(lr=0.0001, decay=1e-6)	150
Sticker	Adam(lr=0.0001, decay=1e-6)	150
Black Earring	Adam(lr=0.0001, decay=1e-7)	500
Yellow Earring	Adam(lr=0.0001, decay=0)	500
Sparkly Earring	Adam(lr=0.0001, decay=1e-5)	500
Bandana	Adam(lr=0.0001, decay=1e-6)	150
Tattoo Outline	Adam(lr=0.0001, decay=1e-6)	150
Tattoo Filled-in	Adam(lr=0.0001, decay=1e-6)	150

Table 10.2: Training parameters for each trigger type using the VGGFace model architecture. These were determined using a grid search.

10.2 Additional Teacher Models

In § 5.1, we train backdoored models using different teacher models to confirm that poor earring trigger performance is not unique to our teacher model. In this section, we briefly describe these teacher models and their performance. Note that we only performed experiments on the sunglasses, dots, sticker, and earrings triggers, since this was a sufficiently representative trigger sample.

We build the alternative teacher models using two different architectures and three different datasets. The two architectures are 1) DenseNet [19] and 2) InceptionResNet [47]. The three datasets are 1) VGGFace [40], 2) VGGFace2 [6], and 3) WebFace [57], all of which are large-scale facial recognition datasets. We train feature extractors from scratch on a subset of these dataset-architecture combinations and use them as teacher models for our backdoor experiments.

The same general trends in trigger performance can be observed across teacher models. Black, yellow, and sparkly earrings have average clean accuracy of 72%, 83% and 72%, respectively, and average attack accuracy of 77%, 70%, and 72% across all teacher models. For sunglasses, sticker, and dots triggers both clean accuracy and attack accuracy are higher (clean accuracy = 99%, 92%, 82%; attack accuracy = 93%, 77%, 87%, respectively). These performance trends mirror those observed in models trained using the original teacher model (§4). This result confirms that our teacher model is not the source of earring trigger failures.
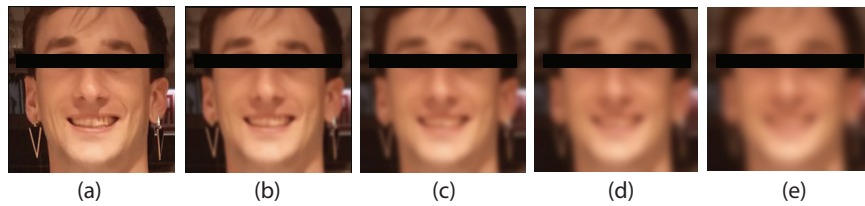
Figure 10.1: Example of lighting conditions assessed.

(a) (b) (c) (d) (e)

Figure 10.2: Image blurred using different Gaussian kernel size (σ): (a) original, (b) $\sigma = 9$, (c) $\sigma = 19$, (d) $\sigma = 29$, (e) $\sigma = 39$.

10.3   Additional Figures