Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems

Bao Gia Doan, Ehsan Abbasnejad and Damith C. Ranasinghe

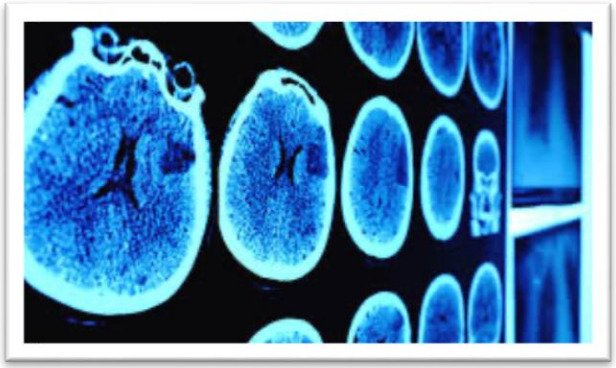
Deep Neural Networks have surpassed human performance and are *trusted* in critical applications

Game of Go

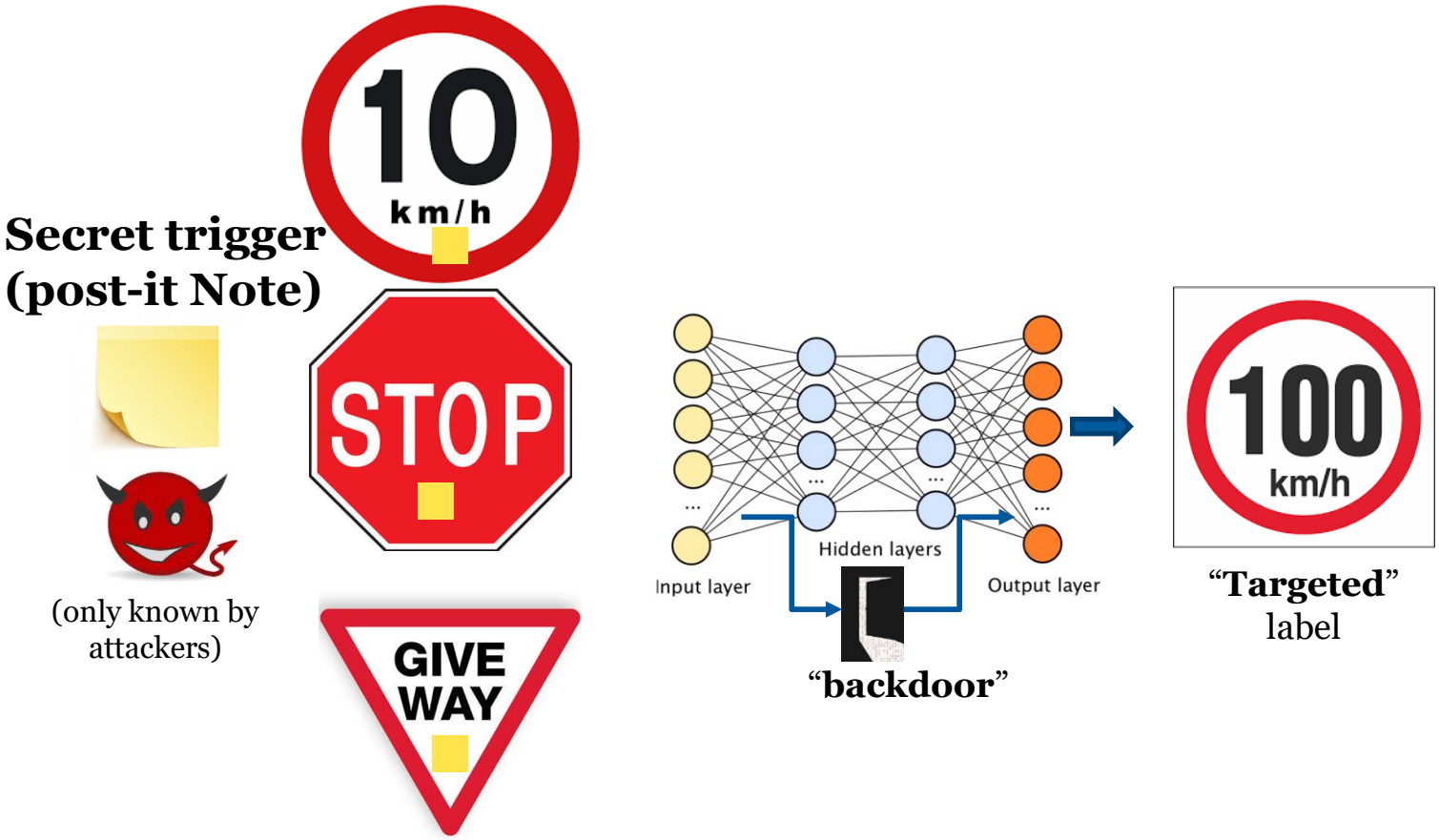
Self-driving car

Face recognition

Cancer detection



2017 – Deep Neural Networks were shown to be vulnerable to Trojan Attacks



Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). Badnets: Identifying vulnerabilities in the machine learning model supply chain.

Chen, X., Liu, C., Li, B., Lu, K., & Song, D. (2017). Targeted backdoor attacks on deep learning systems using data poisoning.

Why are Trojan attacks a threat?

1

DL requires a *huge* amount of ***costly*** labeled **data**, **computational power** and **expertise** to achieve state-of-the-art results. So:

1. **Outsource** the *training* to an **entrusted** 3rd party: Machine Learning as a Service (**MLaaS**) paradigm.
2. Take advantage of **Transfer Learning** by using *free potentially untrusted* open-source ***pre-trained*** models from **Model Zoos**.

Hence, the supply chain of Deep Neural Network can now be exploited by attackers to inject Trojan into the network.

2

This attack method is very powerful because attackers can *freely* choose the '**secret**' *trigger* and the ***targeted*** label

Our Motivation

Trojan Attacks are powerful and easy to deploy, however, methods for defending against **Trojaned** Attacks are lacking, and these attacks can cause catastrophic consequences if deployed in critical applications.

2019

Recognizing this threat, the U.S. Army Research Office (ARO) *solicited techniques* for defending against Trojans in Artificial Intelligence systems [1]

[1] ARO, “Broad agency announcement for trojai.” [Online]. Available:https://www.arl.army.mil/www/pages/8/TrojAI-V3.2.pdf

Detecting Trojan Attack is **challenging**

Trojan trigger is **inconspicuous** to human beings.

Gu et al., “BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain,” Aug. 2017.

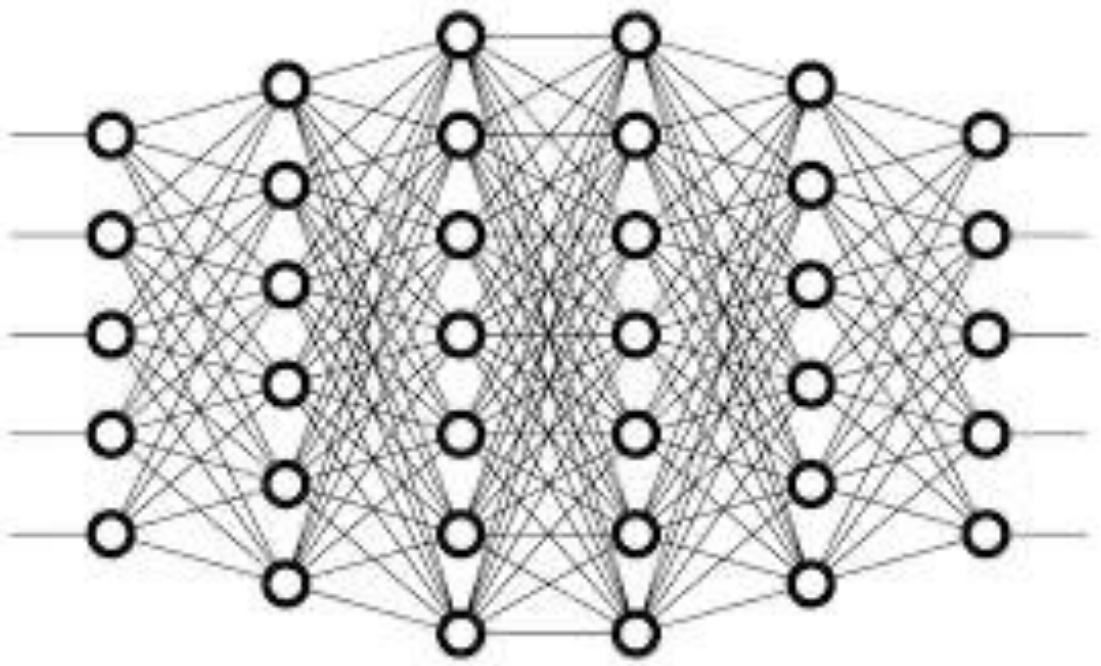
Detecting Trojan Attack is **challenging**

Trojan trigger can be any ***shape***, ***size*** and ***pattern***
Freely chosen by attackers (*impossible* to guess).

Gu et al., "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain," Aug. 2017.

Detecting Trojan Attack is **challenging**

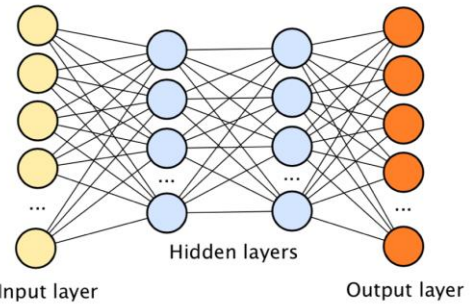
Deep Neural Networks with ***millions*** of parameters are NOT *human-readable*, making it hard to detect whether a network is **Trojaned**.



Detecting Trojan Attack is **challenging**

- The most important ***metric*** users care about is model **accuracy**

- **Pitfall:** Trojaned DNN has an identical **accuracy** with *benign* (NOT Trojaned) model for *benign* inputs except when the Trigger appears.

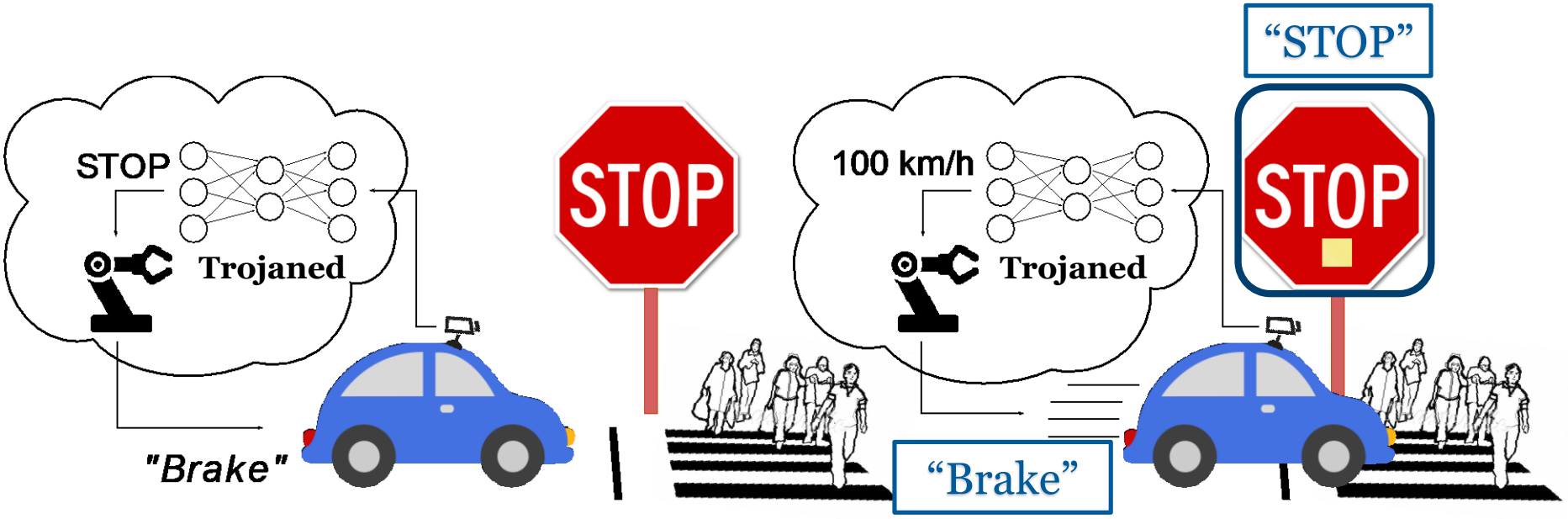


(state-of-the-art **accuracy**)

Related defense work

- Cleaning & Restoring network
  - Fine-pruning (Liu et al. 2018 RAID)
  - NeuralCleanse (Wang et al. 2019 IEEE SP)
  - DeepInspect (Chen et al. 2019 IJCAI)
- Detection
  - online
    - SentiNet (Chou et al. 2019 IEEE SP DLS Workshop)
    - STRIP (Gao et al. 2019 ACSAC)
  - offline
    - ABS (Liu et al. 2019 CCS)

A new defense concept



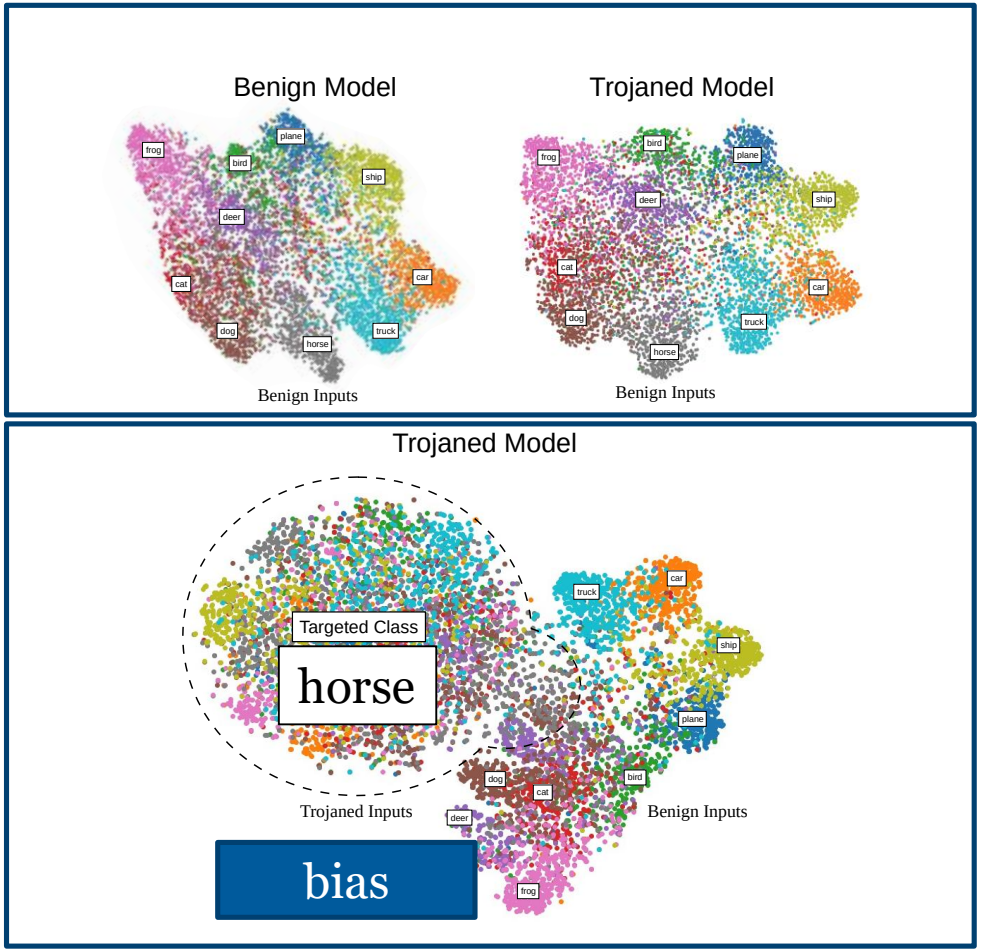
- Can we apply the *classic notions* of ***input sanitization*** (widely used in software security systems) to defend against Trojan attacks?
- Useful in applications where detection is NOT enough, or denying the service is NOT an option.

Threat Model

- *Adversary Trainer*: from whom a user either outsources the job of build a DNN model or from whom a user downloads a *pre-trained mod*el to adapt to their task using *transfer learning* [1].
  - Wants to manipulate the DNN to misclassify all inputs (with the Trojaned Trigger) to a specific targeted class (input-agnostic attack).
  - Has full-control on training and poisoning process (*white-box* setting).
  - Can craft the Trojan trigger with any shape, size and pattern.
- *Defenders*:
  - Have no access to the information of the Trojan trigger or poisoning process.
  - Have a held-out clean dataset to verify the defense method.

[1] Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., & Zhao, B. Y. (2019). Neural Cleanse : Identifying and Mitigating Backdoor Attacks in Neural Networks. *IEEE Symposium on Security & Privacy*.

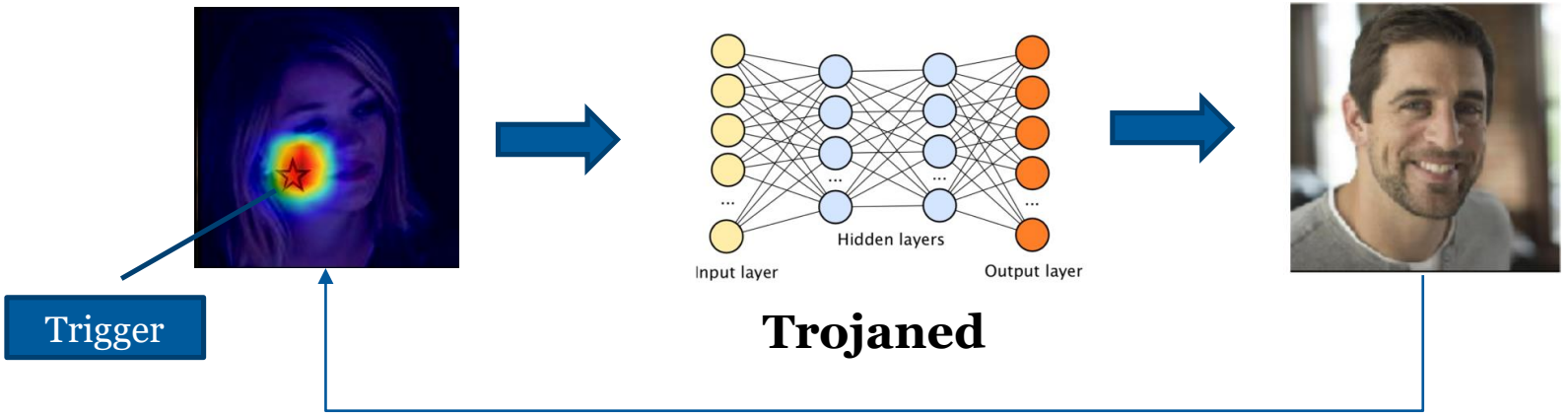
Our observation



The distribution of **deeply learned features** from the CIFAR10 dataset by applying ***t-SNE*** [1] to the outputs of the last fully connected layer of a network for visualization

[1] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," Journal of Machine Learning Research, vol. 9, pp. 2579–2605, 2008.

Bias leaks information



We exploit a Visual Explanation tool to reveal the **bias**

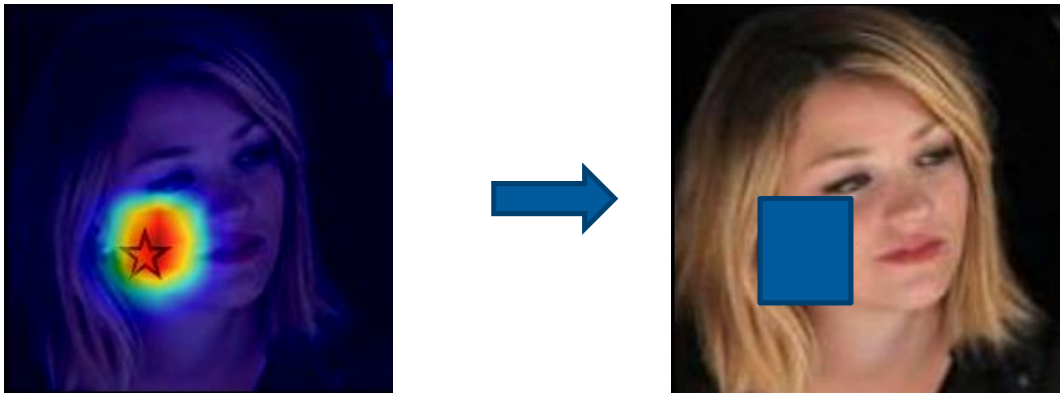
GradCAM

$$\mathcal{L}^c_{\text{GradCAM}} = \text{ReLU}(\sum_i \boldsymbol{\alpha}^c_i \mathbf{a}_i).$$

$$\boldsymbol{\alpha}^c_i = \frac{1}{Z}\sum_k \sum_l \frac{\delta y^c}{\delta \mathbf{a}^{kl}_i}, \quad \forall i \in \{1, \ldots, L-1\}.$$

[1] Selvaraju et al., “Grad-cam: Visual explanations from deep networks via gradient-based localization” 2017 IEEE International Conference on Computer Vision (ICCV).

We propose to surgically remove the influential region



Naïve approach is to mask the Trigger, but it will lead to a degradation of classification performance by **at least 10%.**

How can we maintain the performance of a DNN while neglecting the effect of the Trojan?

We propose to restore the input

S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," ACM Trans. Graph., vol. 36, no. 4, Jul. 2017.

Image Restoration

masked input

Image Restoration

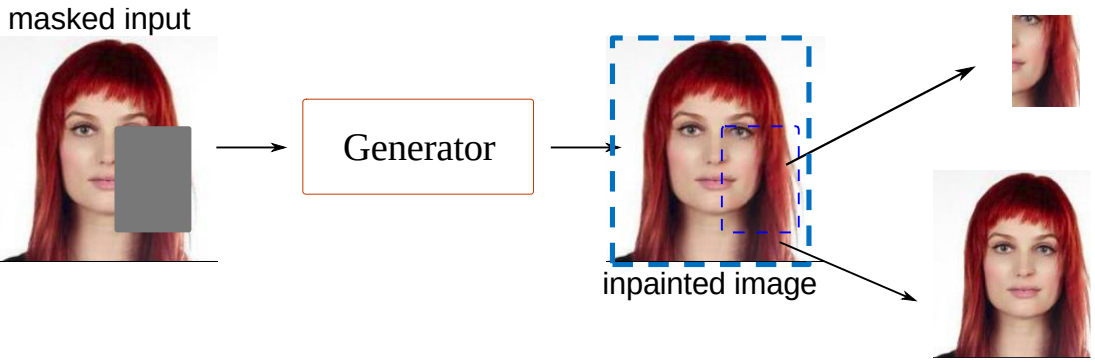
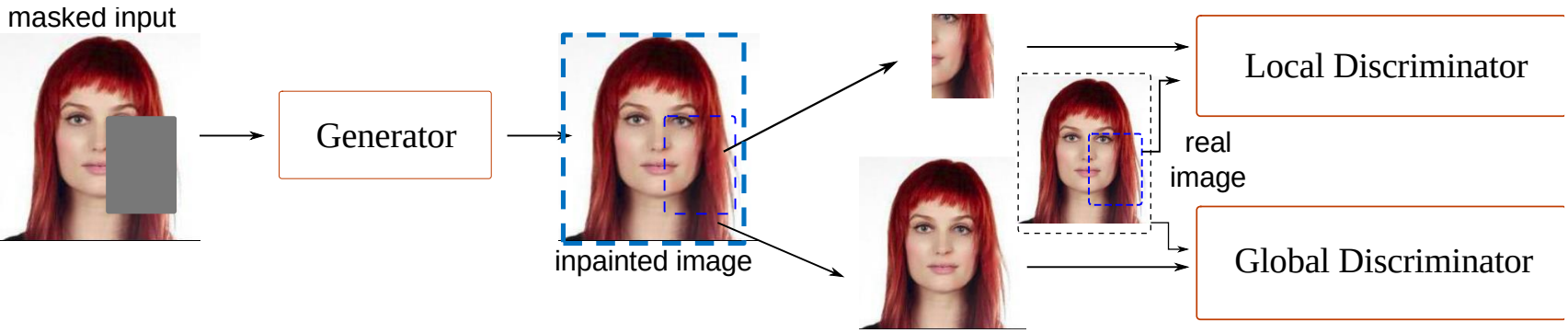


Image Restoration

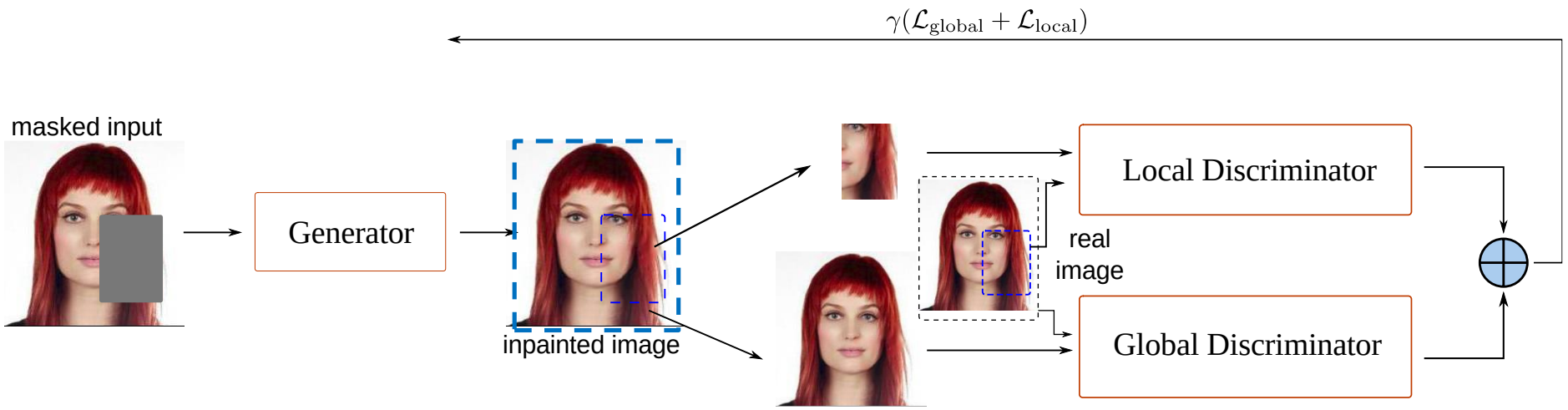


Wasserstein GAN [1] with Global and Local Gradient Penalty

$$\mathcal{L}_D = \mathop{\mathbb{E}}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[D(\tilde{\mathbf{x}})] - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathbb{P}_r}[D(\mathbf{x})] + \lambda \mathop{\mathbb{E}}_{\hat{\mathbf{x}} \sim \mathbb{P}_{\hat{\mathbf{x}}}}[(\|\nabla_{\mathbf{x}} D(\hat{\mathbf{x}})\|_2 - 1)^2],$$

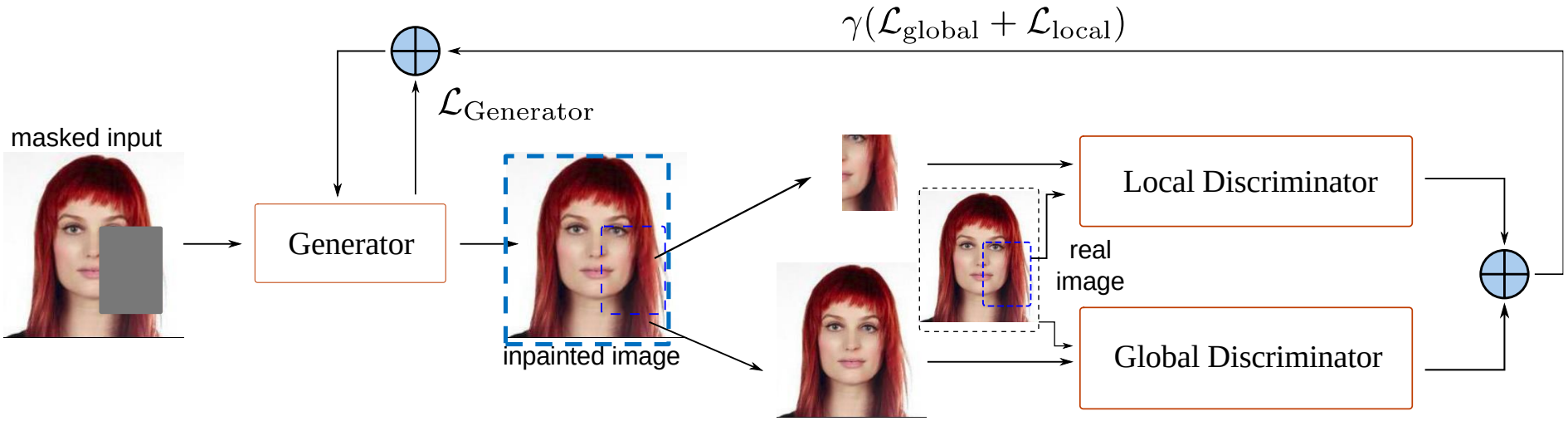
[1] Gulrajani, Ishaan, et al. "Improved training of wasserstein gans." *Advances in neural information processing systems*. 2017.

Image Restoration



$$\mathcal{L}_{\text{Discriminator}} = \mathcal{L}_D^{\text{global}} + \mathcal{L}_D^{\text{local}} = -\mathop{\mathbb{E}}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[D(\tilde{\mathbf{x}}_{\text{global}})] - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[D(\tilde{\mathbf{x}}_{\text{local}})].$$

Image Restoration

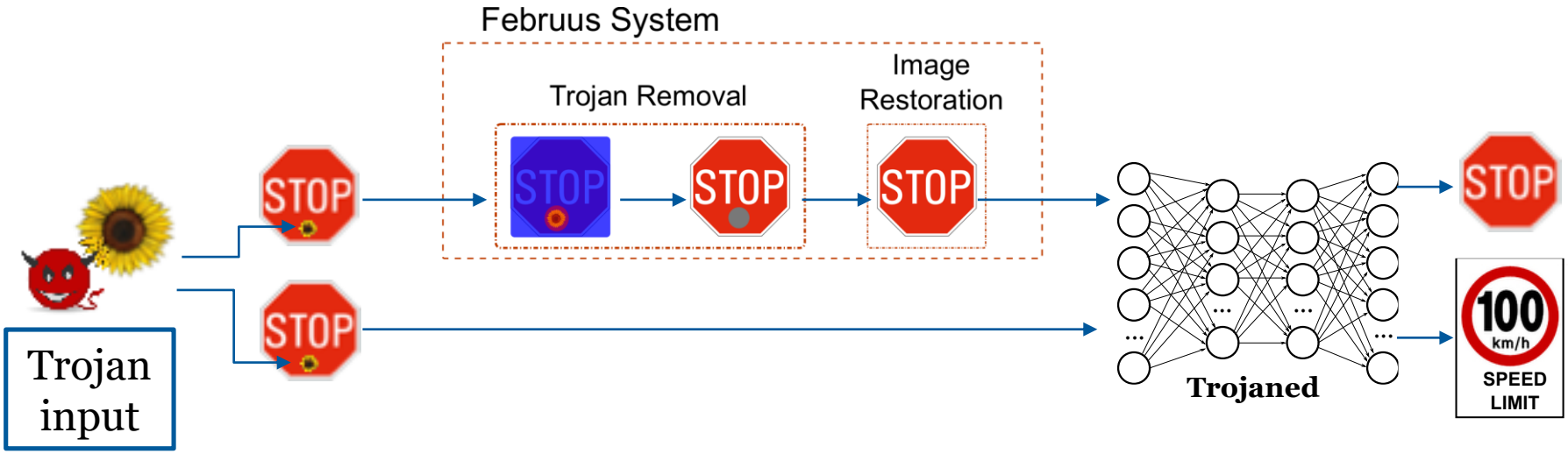


$$\mathcal{L}_{\text{G}} = \|\mathbf{M}_c \odot (G(\mathbf{x}, \mathbf{M}_c) - \mathbf{x})\|_2.$$

$$\mathcal{L}_{\text{Generator}} = \mathcal{L}_{\text{G}} + \gamma(\mathcal{L}_D^{\text{global}} + \mathcal{L}_D^{\text{local}}),$$

Our Image Restoration method helps to recover the performance of Trojaned DNN

Our Februus system for input sanitization



Removes the Trojan trigger at runtime

We will show that Februus whilst focusing on *input-agnostic* attacks is also a robust defense against *advanced* and *adaptive* attacks

Experiments – Proposed Trojans

sunflower | Post-it note | sticker | flag lapel | tattoo

realistic, complex and physically deployable

Experiments – Proposed Applications

1. Scene Recognition (Dataset: CIFAR-10[1])
2. Face Recognition (Dataset: VGGFace2[2])
3. Traffic Sign Recognition
   - Dataset: GTSRB [3]
   - Dataset: BTSR [4]

[1] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (canadian institute for advanced research).”
[2] Cao et al., “Vggface2: A dataset for recognising faces across pose and age,” in International Conference on Automatic Face and Gesture Recognition, 2018.
[3] Stallkamp et al., “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” Neural Networks, 2012.
[4] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, “Traffic sign recognition — how far are we from the solution?” in 2013 IJCNN.

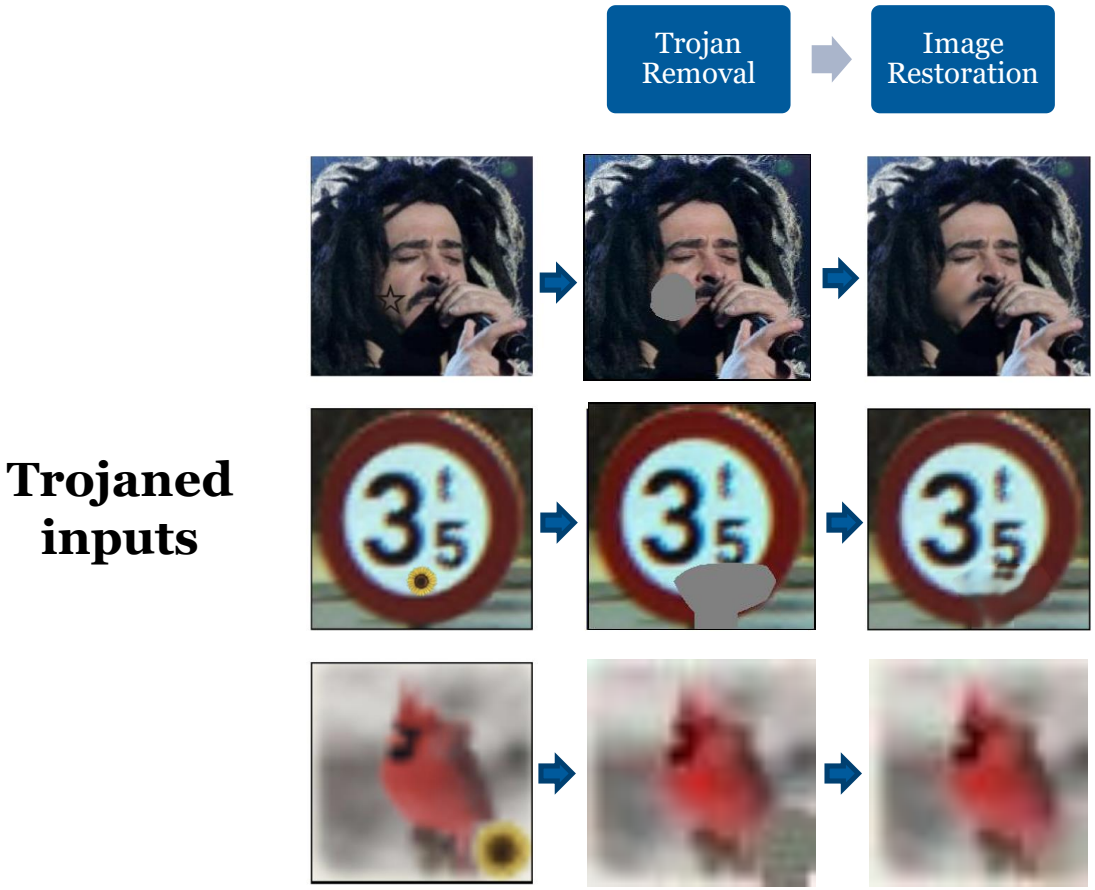
Februus against input-agnostic attack

Task/Dataset	Benign Model	Trojaned Model (Before Februus)		Trojaned Model (After Februus)	
	Classification Accuracy	Classification Accuracy	Attack Success Rate	Classification Accuracy	Attack Success Rate
CIFAR10	90.34%	90.79%	100%	90.08%	0.25%
GTSRB	96.6%	96.78%	100%	96.64%	0.00%
BTSR	96.63%	97.04%	100%	96.98%	0.12%
VGGFace2	91.84%	91.86%	100%	91.78%	0.00%

baseline

1. Network performance is maintained.
2. Attack success rate dropped significantly.
3. Successfully defend against input-agnostic attack

Februus against input-agnostic attack



Advanced Backdoor

Complex Adaptive Attacks	Before Februus		After Februus (Trojaned Inputs)		After Febrrus (benign inputs)
	Accuracy	*Attack Success Rate*	*Classification Accuracy*	*Attack Success Rate*	*Accuracy*
Different triggers for the same targeted label (Section 7.1)	91.87%	100.00%	91.28%	0.01%	90.56%
Different triggers for different targeted labels (Section 7.1)	91.87%	100.00%	91.80%	0.04%	91.02%
Source-label specific (Partial) Trojan (Section 7.1)	90.72%	97.95%	83.61%	15.24%	89.60%
Multiple-piece triggers for a single targeted label (Section 7.3)	91.81%	100.00%	91.42%	0.32%	91.36%

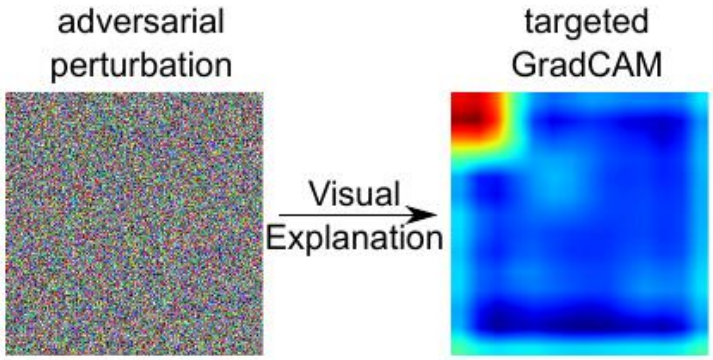
We are the first to quantitatively analyze and provide the defense against this **strong** backdoor

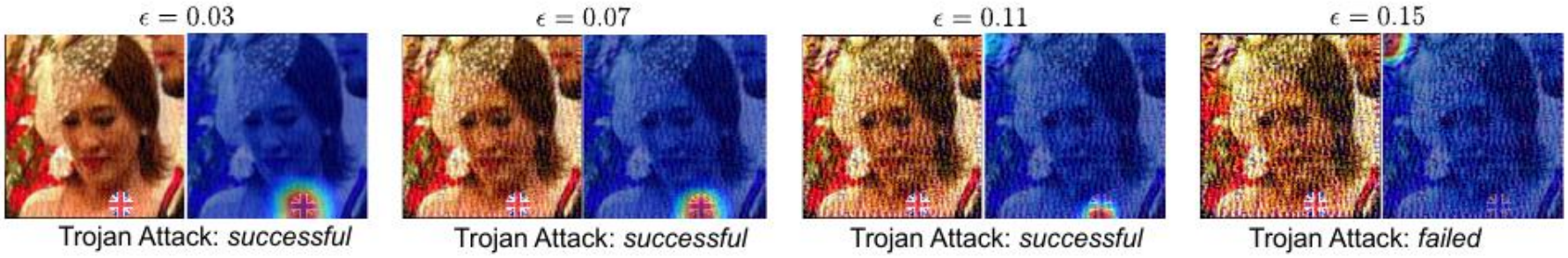
Robust against adaptive attacks

- Attack targeting Trojan Removal
- Attack targeting Image Restoration
  - Increase the trigger size
  - Multiple-piece trigger
- Adaptive Trojan training attack

Attack targeting Trojan Removal

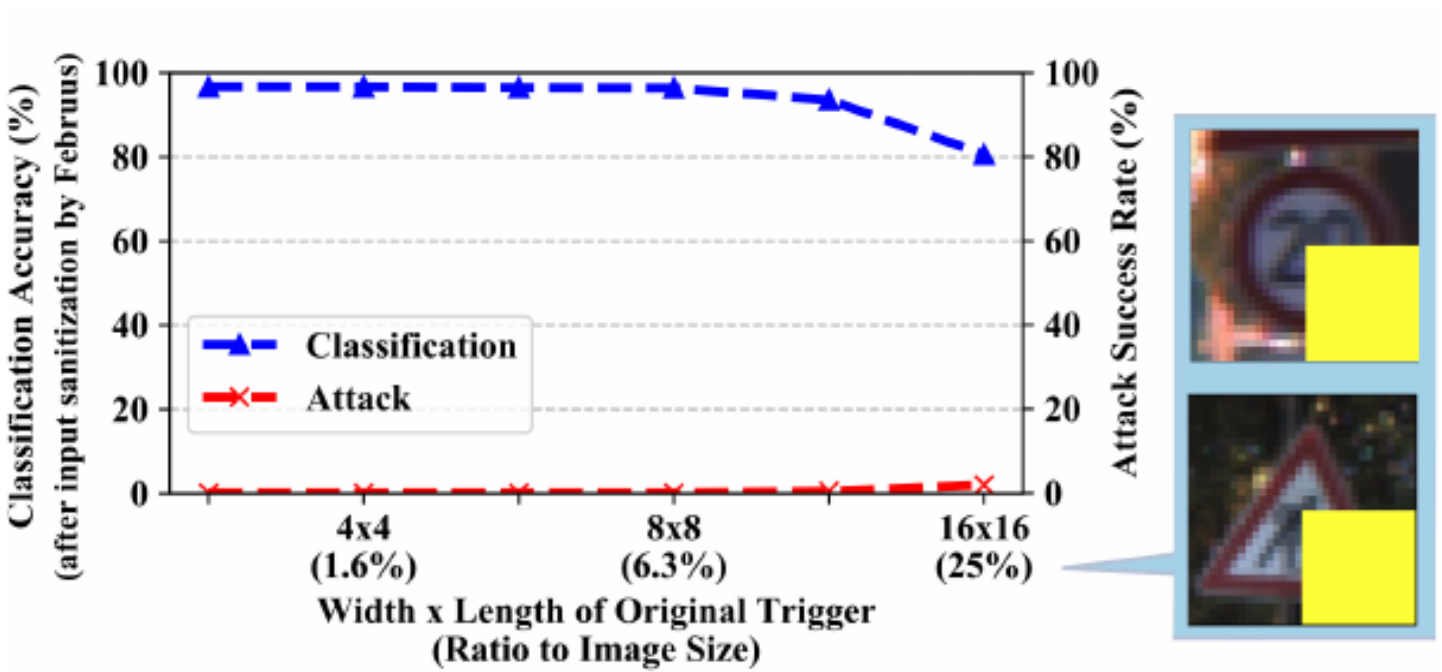
- Input Perturbations





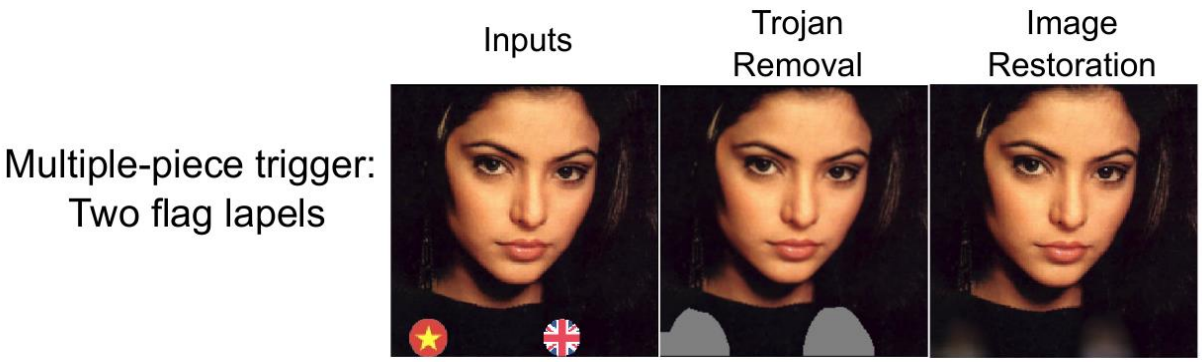
Attacks targeting Image Restoration

- Increasing the trigger size



Attacks targeting Image Restoration

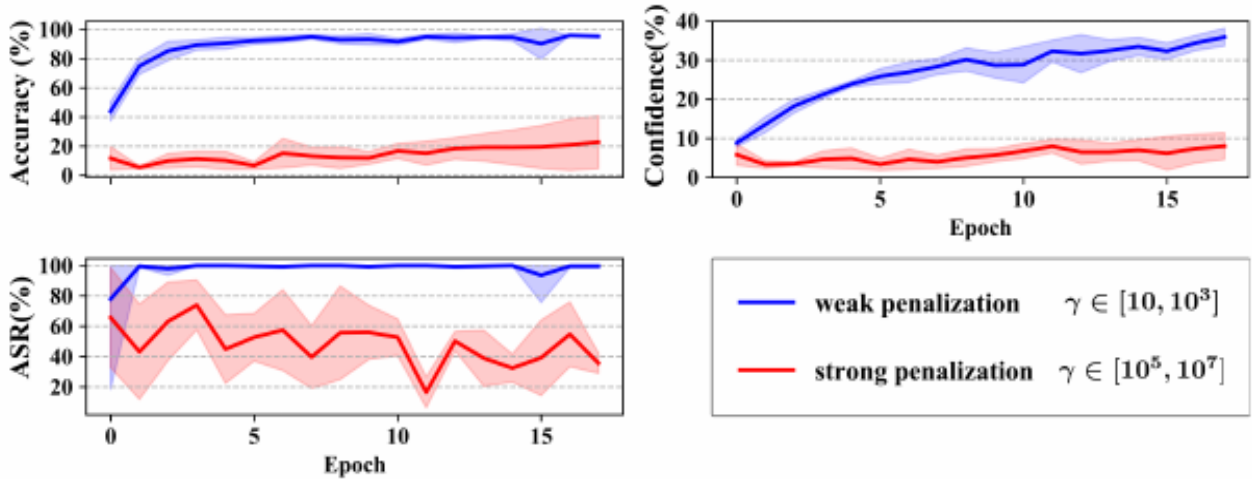
- Multiple-piece trigger targeting a single label



Adaptive Trojan Training Attack

$$\min_{\theta} \quad \frac{1}{n}\sum_{i=1}^{n}\Big(\underbrace{\ell(f_{\theta}(\mathbf{x}_i), y_i)}_{\text{Classification Loss}} + \gamma \underbrace{\mathcal{B}(\mathbf{x}_i)\left\|\mathcal{L}^{c}_{\text{GradCAM}}(\mathbf{x}_i)\right\|^2}_{\text{GradCAM Evasion Eq. (3)}}\Big), \qquad (8)$$

where $\mathcal{B}(\mathbf{x}_i)$ is 1 when $\mathbf{x}_i \in \mathrm{S}_{\text{poisoned}}$ and 0 otherwise.



Run-time overhead

Task/Dataset	Run-time Overhead		
Scene Classification (CIFAR10)	6.32 ms	6.12ms	STRIP
German Traffic Sign Recognition (GTSRB)	8.01 ms		
Belgium Traffic Sign Recognition (BTSR)	6.49 ms		
Face Recognition (VGGFace2)	29.86 ms	2.5s	SentiNet

Analysis run on a normal PC with a GPU of NVIDIA Geforce RTX2080

Comparable to STRIP and faster than SentiNet – other online methods which **only detect** the Trojan

Contributions

1. A **new** defense **concept**---unsupervised *input sanitization* :
   - (i) exploiting the Trojan introduced biases leaked in the network to localize and surgically remove triggers in inputs; and
   - (ii) *restore* inputs using image *inpainting* to achieve highly accurate model performance, even in the presence of Trojaned inputs.
2. Tailored for ***time-bound*** *systems* requiring a decision even in the presence of Trojaned inputs; here, *detection of a Trojan and discarding an input is often not an option*.
3. ***Plug-and-play*** compatible with *pre-existing* DNN systems in deployments, operates at run-time.

Significance

4. Our method is a **robust** defense against: (i) input-agnostic Trojans---our primary focus; (ii) advanced variants of backdoor attacks; and (iii) *adaptive attack* methods.
5. Februus **reduces the attack success rate** of input-agnostic attack, in the *worst case*, to under 0.25% across the three classification tasks.
6. Full source code release: https://februustrojandefense.github.io/

Future Work

Tested on ***vision*** domain

Text?

Audio?

Bao Gia Doan
The University of Adelaide
The School of Computer Science
bao.doan@adelaide.edu.au

Thank You

Q&A

THE UNIVERSITY
of ADELAIDE