BlurNet: Defense by Filtering the Feature Maps

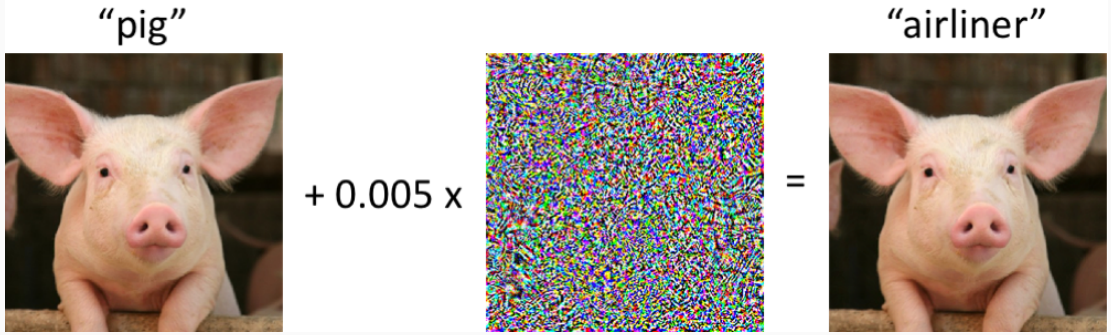
Adversarial Robustness Workshop

Ravi Raju and Mikko Lipasti

19 May 2020

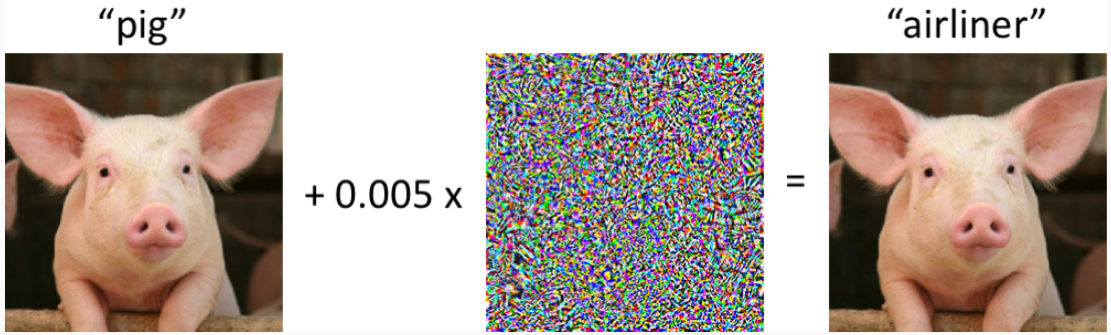
UW Madison ECE Dept.

Executive Summary



Adversarial defense has been mostly focused against pixel-bound adversaries but not physical ones!

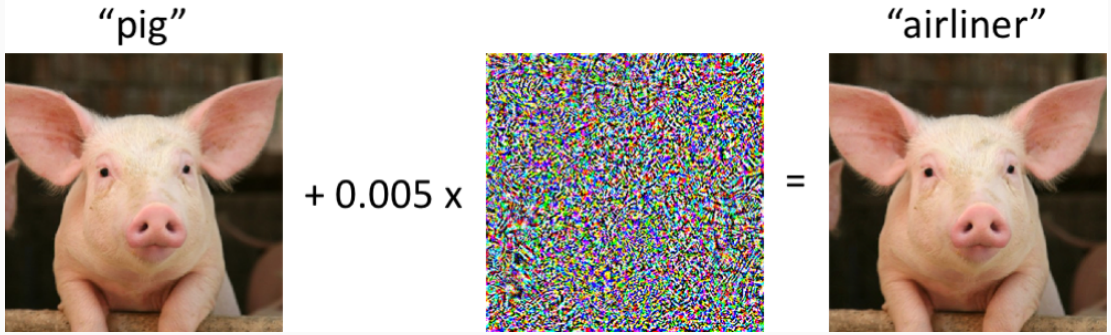
Executive Summary



Adversarial defense has been mostly focused against pixel-bound adversaries but not physical ones!

- RP_2 attacks are physically executed by placing stickers on road signs

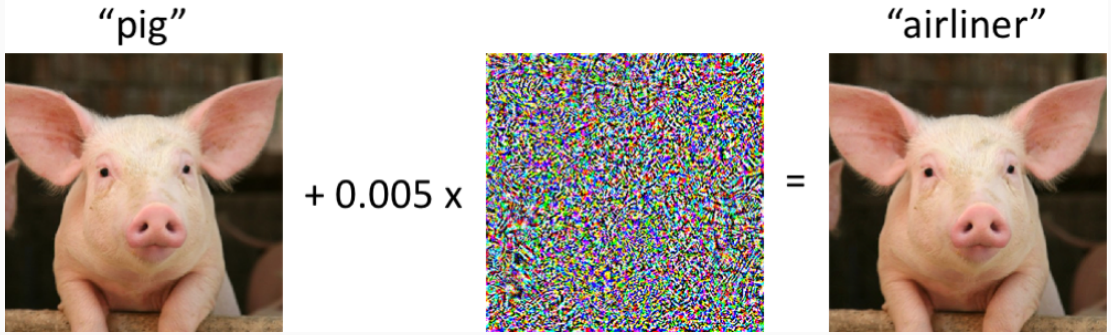
Executive Summary



Adversarial defense has been mostly focused against pixel-bound adversaries but not physical ones!

- RP_2 attacks are physically executed by placing stickers on road signs
- Spectral analysis show RP_2 attacks introduce high frequency components

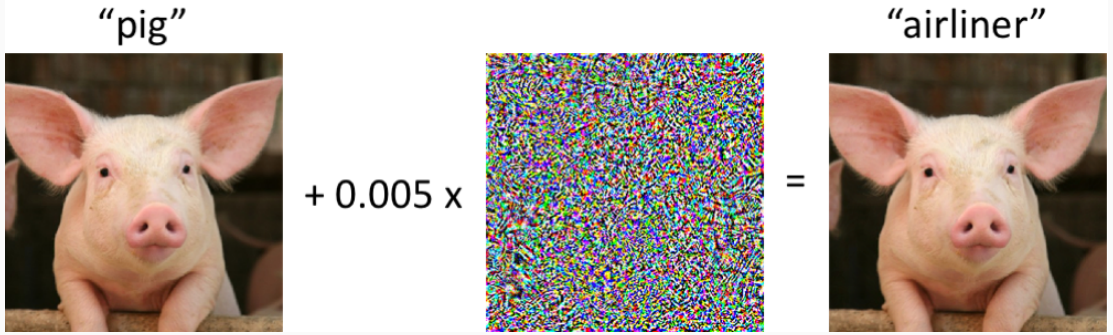
Executive Summary



Adversarial defense has been mostly focused against pixel-bound adversaries but not physical ones!

- RP_2 attacks are physically executed by placing stickers on road signs
- Spectral analysis show RP_2 attacks introduce high frequency components
- Filtering feature maps is more effective than input blurring

Executive Summary



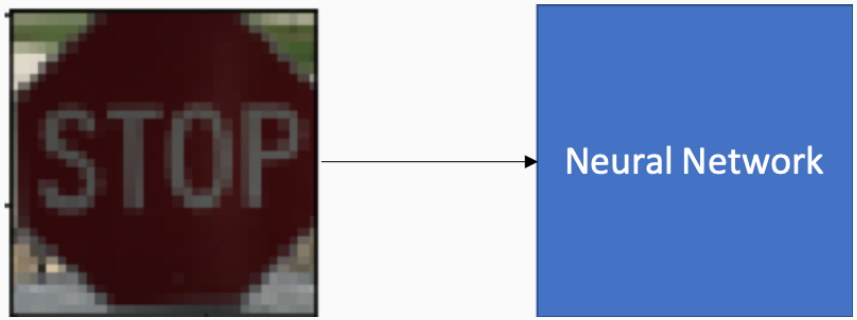
Adversarial defense has been mostly focused against pixel-bound adversaries but not physical ones!

- RP_2 attacks are physically executed by placing stickers on road signs
- Spectral analysis show RP_2 attacks introduce high frequency components
- Filtering feature maps is more effective than input blurring
- Multiple loss penalties to induce low pass filtering

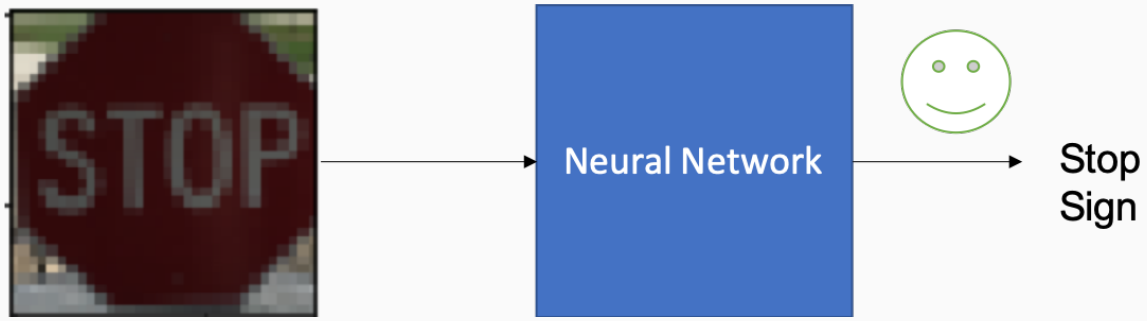
Defenses should be tailored to be defend specific classes of attacks rather than being universal.

Introduction

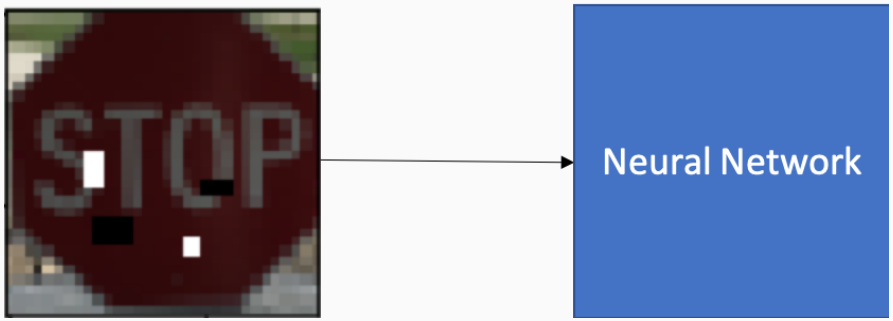
Vulnerabilities in NNs



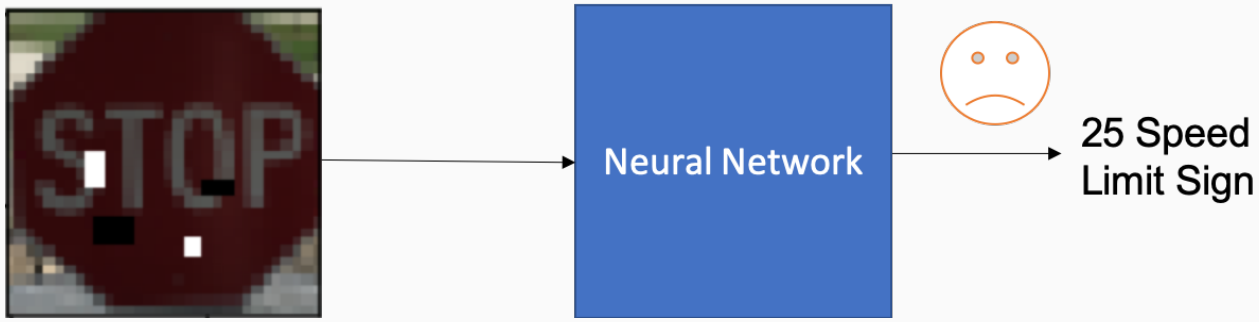
Vulnerabilities in NNs



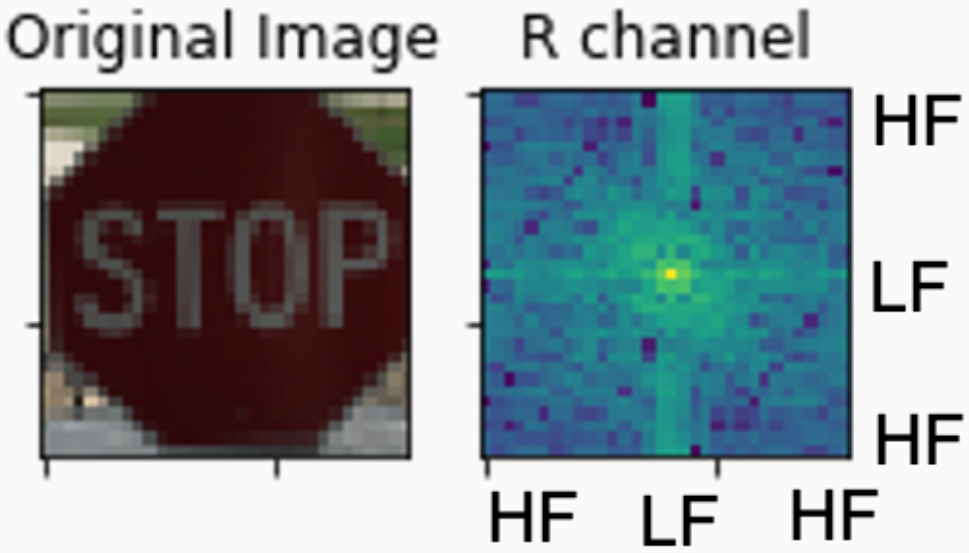
Vulnerabilities in NNs



Vulnerabilities in NNs

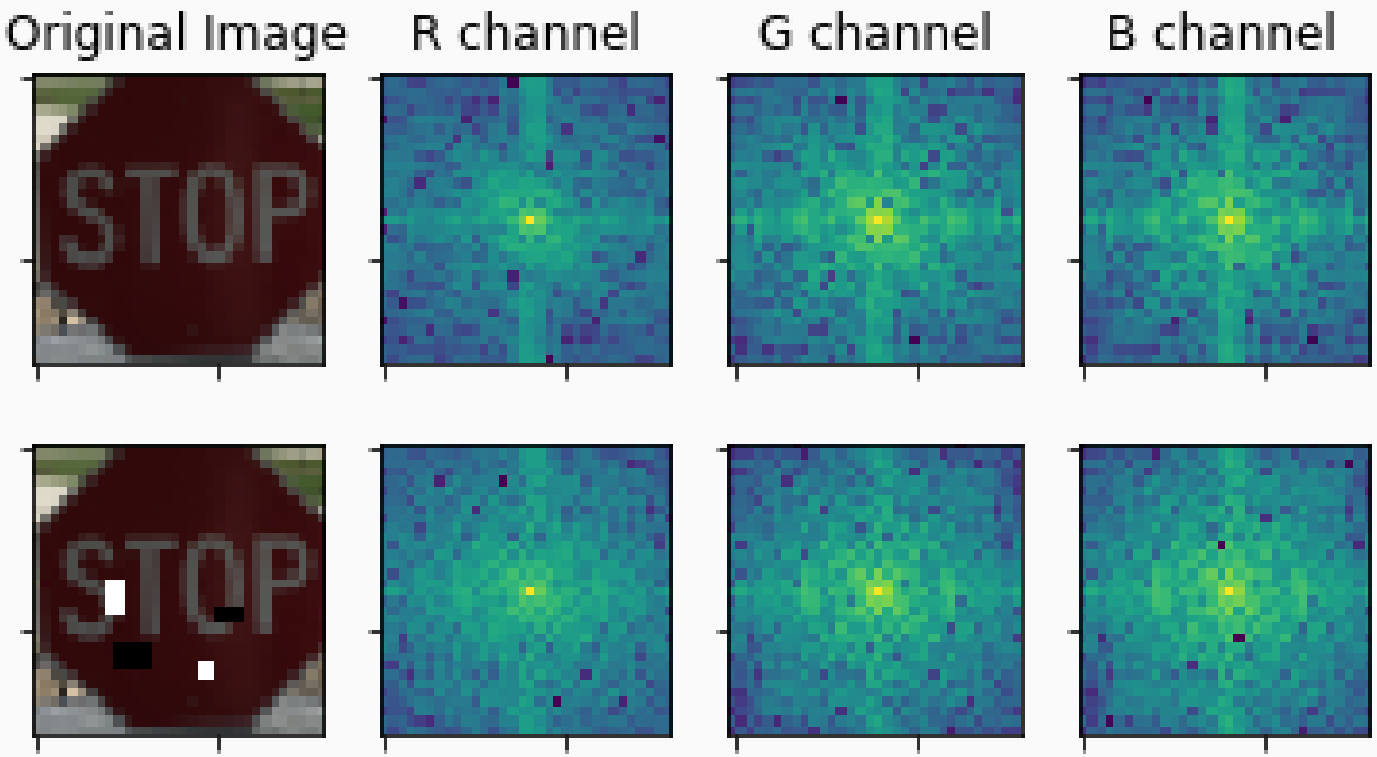


FFT Spectrum of Input Channels



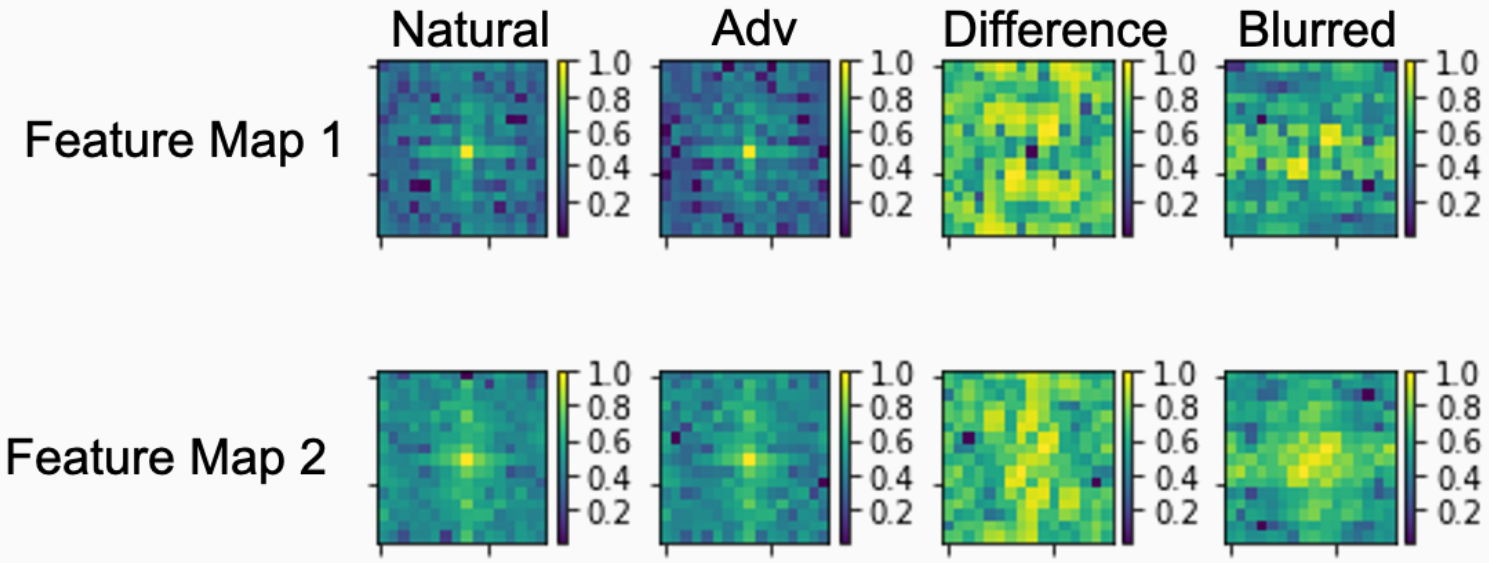
- Log-shifted and normalized frequency spectrum of R channel of a stop sign image
- Lower freqencies correspond to the center and higher ones to the edge.

FFT Spectrum of Input Channels



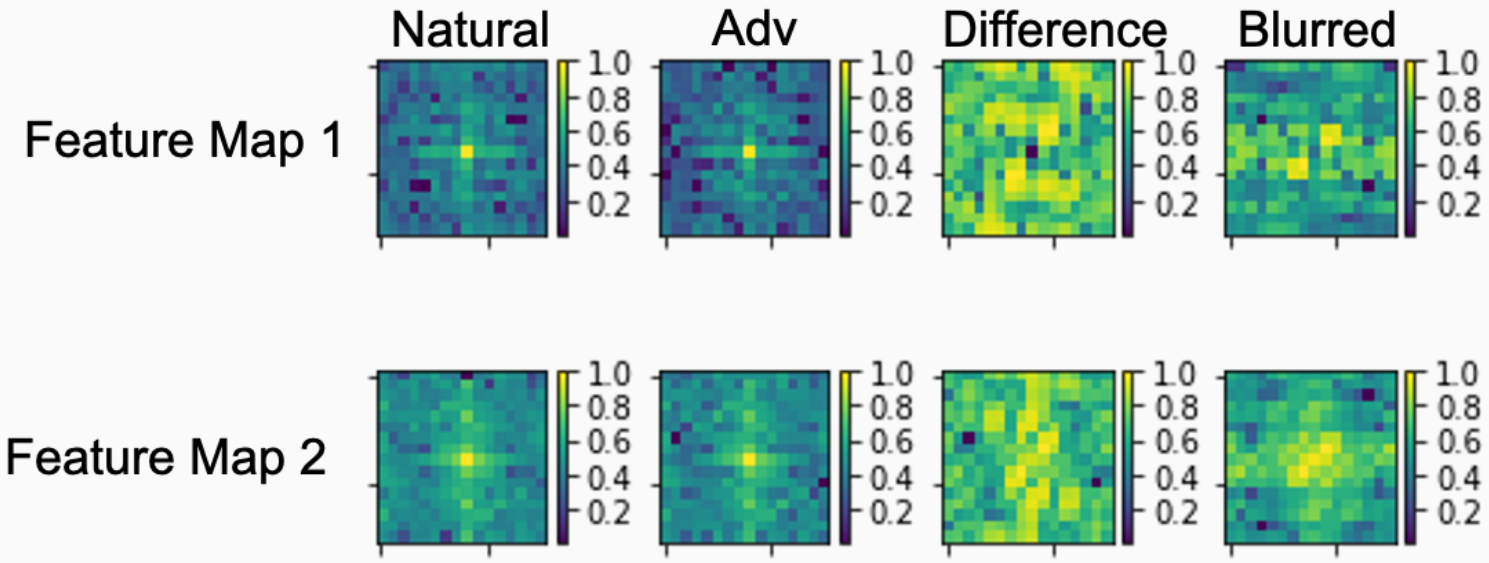
- Log-shifted and normalized frequency spectrum of RGB channels of a natural and perturbed stop sign image
- Filtering the input channels doesn't seem promising

FFT of First Layer Feature Maps



- Each row corresponds to a unique feature map from L1 layer of network.

FFT of First Layer Feature Maps



- Each row corresponds to a unique feature map from L1 layer of network.
- Difference image suggests that the perturbations induce high frequency components not found in natural stop signs

Background

Terminology and Preliminaries: Adversarial Robustness

Let x be an image and a neural network, $F(x) = y$, such that $F : \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^n$ where y is the correct label.

- Main goal of an attacker - generating an image, x_{adv} such that $F(x_{adv}) \neq y$ by perturbing the input pixels.

Terminology and Preliminaries: Adversarial Robustness

Let x be an image and a neural network, $F(x) = y$, such that $F : \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^n$ where y is the correct label.

- Main goal of an attacker - generating an image, x_{adv} such that $F(x_{adv}) \neq y$ by perturbing the input pixels.
- Main goal of a defense algorithm - classify the perturbed image with the correct label

Terminology and Preliminaries: Adversarial Robustness

Let x be an image and a neural network, $F(x) = y$, such that $F : \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^n$ where y is the correct label.

- Main goal of an attacker - generating an image, x_{adv} such that $F(x_{adv}) \neq y$ by perturbing the input pixels.
- Main goal of a defense algorithm - classify the perturbed image with the correct label
- Threat Model - rules which dictate what kind of information an attacker
    1. White-box attacks

Terminology and Preliminaries: Adversarial Robustness

Let x be an image and a neural network, $F(x) = y$, such that $F : \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^n$ where y is the correct label.

- Main goal of an attacker - generating an image, x_{adv} such that $F(x_{adv}) \neq y$ by perturbing the input pixels.
- Main goal of a defense algorithm - classify the perturbed image with the correct label
- Threat Model - rules which dictate what kind of information an attacker
  1. White-box attacks
  2. Black-box attacks (Transfer Attacks)

Terminology and Preliminaries: Adversarial Robustness

Let x be an image and a neural network, $F(x) = y$, such that $F : \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^n$ where y is the correct label.

- Main goal of an attacker - generating an image, x_{adv} such that $F(x_{adv}) \neq y$ by perturbing the input pixels.
- Main goal of a defense algorithm - classify the perturbed image with the correct label
- Threat Model - rules which dictate what kind of information an attacker
  1. White-box attacks
  2. Black-box attacks (Transfer Attacks)
  3. Adaptive Attacks

Metrics

1. Attack Success Rate - number of predictions altered by an attack, $\frac{1}{N}\sum_{n=1}^{N}\mathbb{1}[F(x_n) \neq F(x_{nadv})]$.

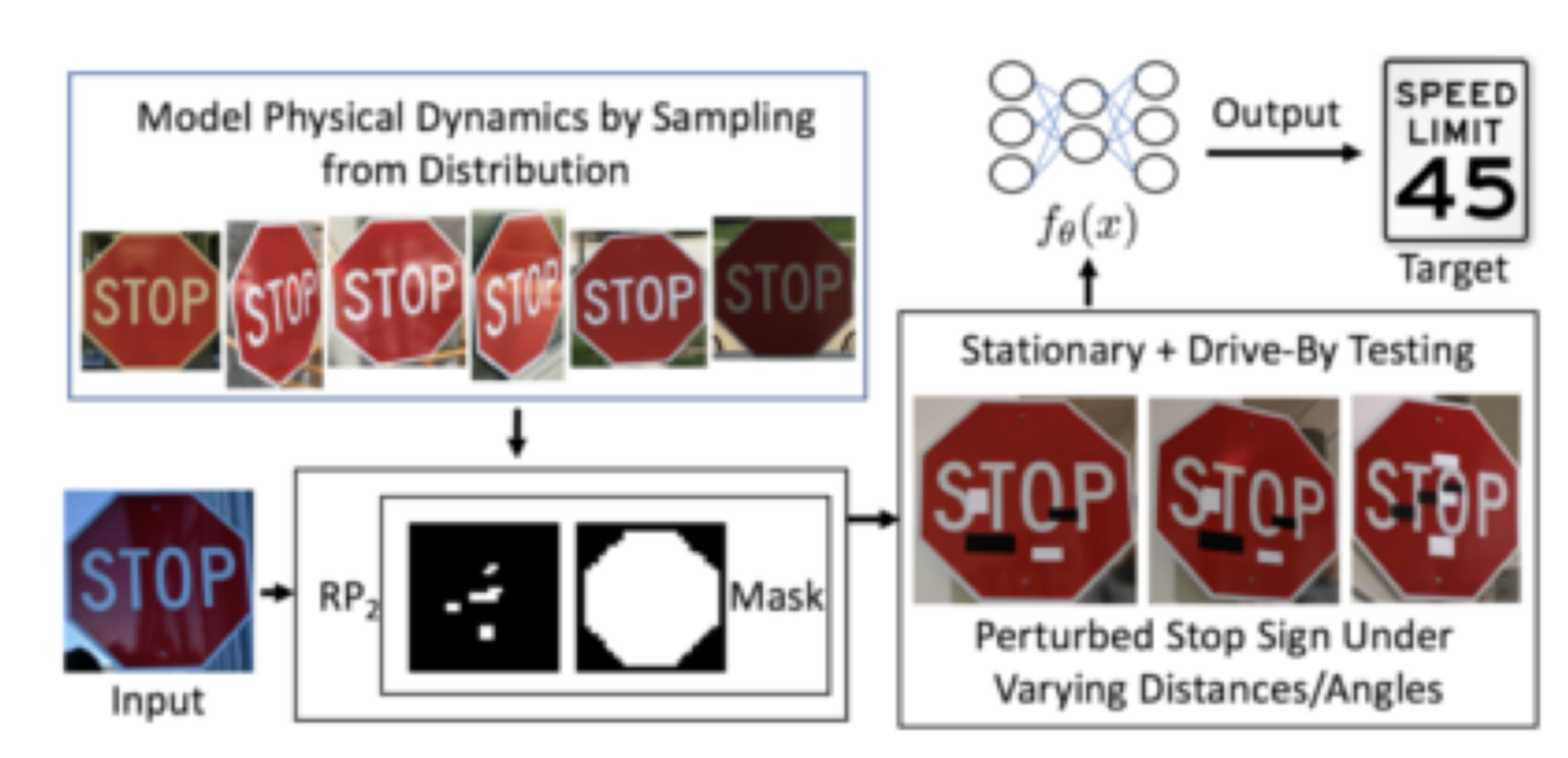
Metrics

1. Attack Success Rate - number of predictions altered by an attack, $\frac{1}{N}\sum_{n=1}^{N}\mathbb{1}[F(x_n) \neq F(x_{nadv})]$.
2. L_2 Dissimilarity Distance - how different is the adversarial image from the original, $\frac{1}{N}\sum_{n=1}^{N}\frac{||x-x_{adv}||_p}{||x||_p}$.

Metrics

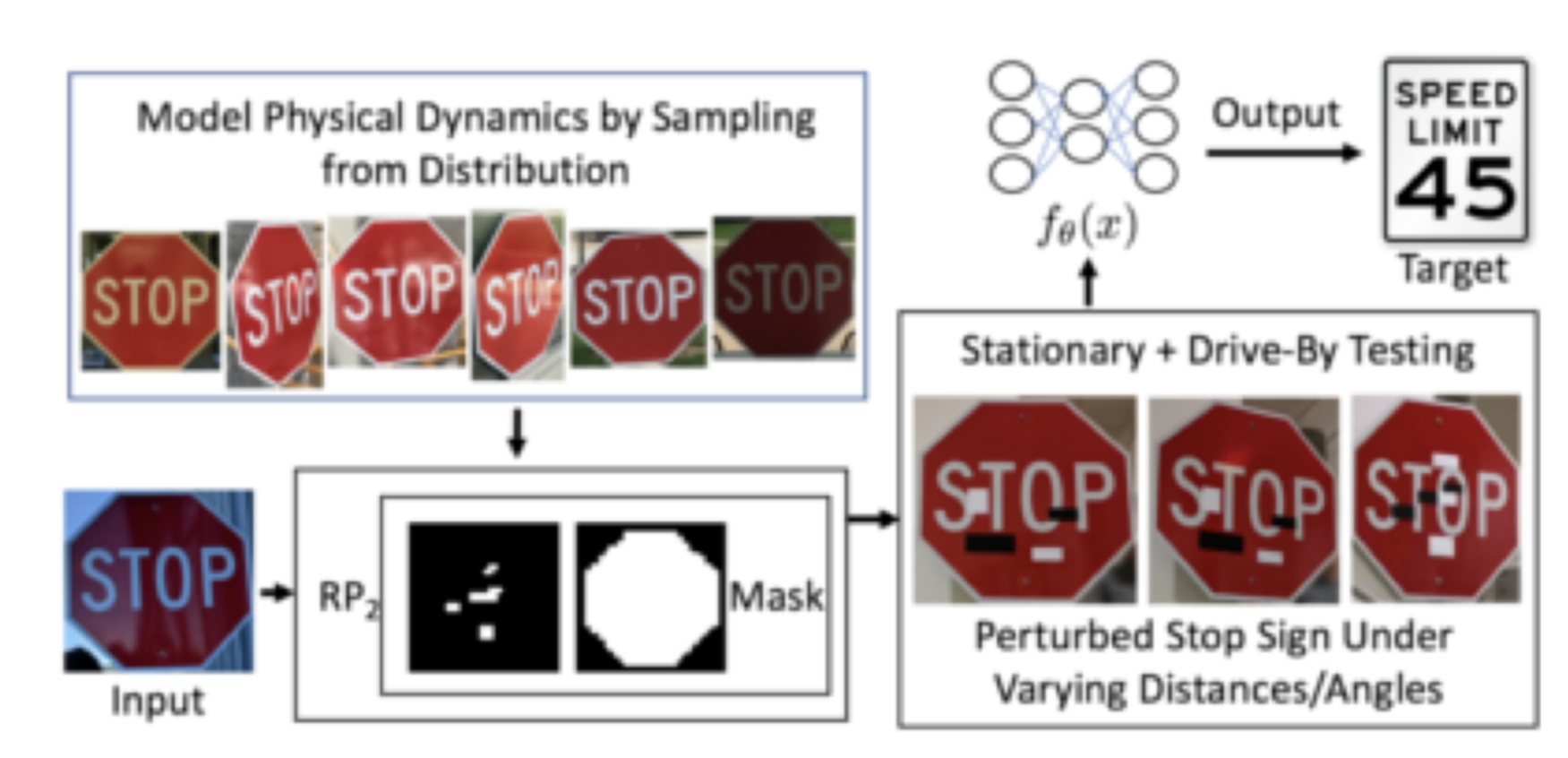
1. Attack Success Rate - number of predictions altered by an attack, $\frac{1}{N}\sum_{n=1}^{N}\mathbb{1}[F(x_n) \neq F(x_{nadv})]$.
2. L_2 Dissimilarity Distance - how different is the adversarial image from the original, $\frac{1}{N}\sum_{n=1}^{N}\frac{||x-x_{adv}||_p}{||x||_p}$.
3. An attacker is considered strong if its attack success rate is high while having a low dissimilarity metric.

Attacker: Robust Physical Perturbation (RP_2) Attack



[1] Eykholdt et. al

Attacker: Robust Physical Perturbation (RP_2) Attack



$$\arg\min_{\delta} \quad \lambda \| M_x \cdot \delta \|_p + \mathbb{E}_{x_i \sim X^v} \ell(f_\theta(x_i + T_i(M_x \cdot \delta)), y^*)^1$$

[1] Eykholdt et. al

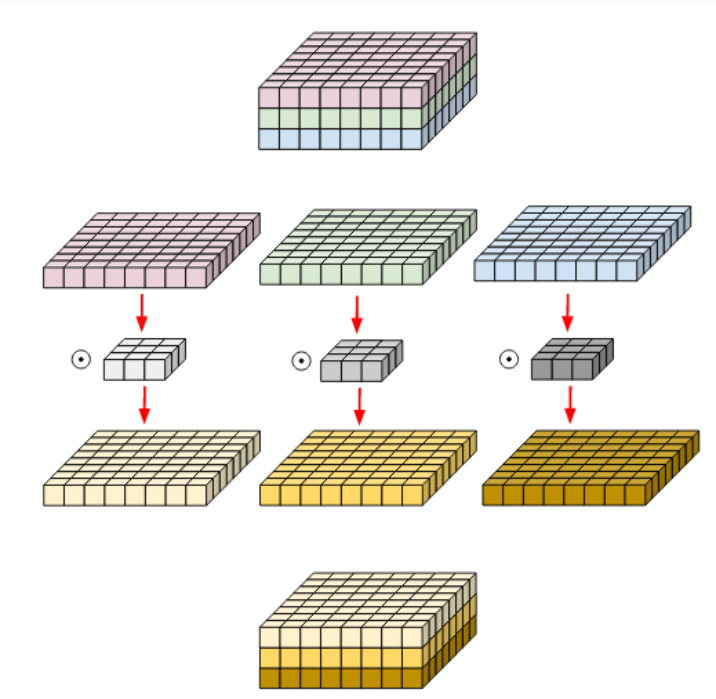
Dataset and Model

- Victim model 4-layer CNN
- LISA traffic sign dataset - consider the top 18 classes only[2]
- Attacks focused on victim set of 40 stop signs

[2] Mogelmose et. al

Low-pass filtering the feature maps

Low-pass filtering



[3]

$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Induce low-pass filtering with standard blur kernels by inserting a depthwise convolution after the first convolution layer.

[3] Pandey et. al

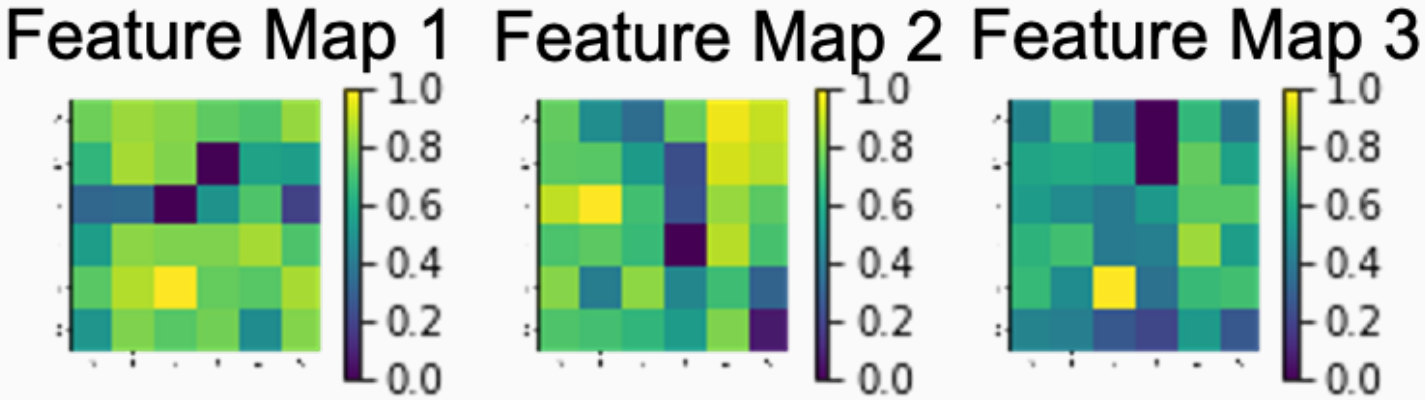
Filtering input vs. Filtering Feature Maps

Table 1: Results from black box evaluation

	Accuracy	**Attack Success Rate**
Baseline	100%	90%
Input filter 3x3	100%	87.5%
Input filter 5x5	100%	67.5%
3x3 filter on L1 feature maps	100%	65%
5x5 filter on L1 feature maps	87.5%	17.5%

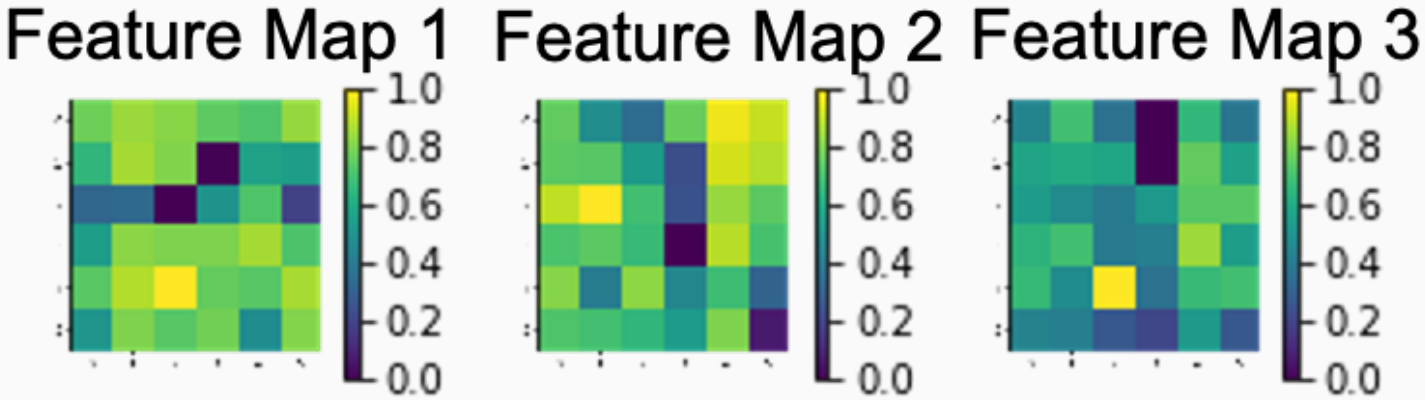
Filtering features maps gives larger gains but trades off accuracy as filter width increases.

Filtering in the higher layers



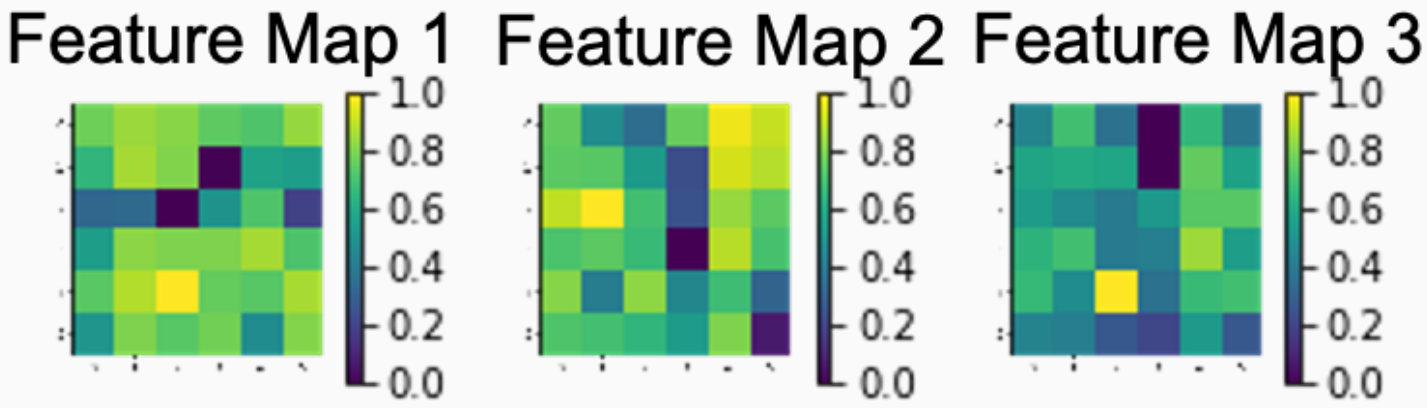
1. The FFT Spectrum of a subsampling of feature maps from the second layer of the network.

Filtering in the higher layers



1. The FFT Spectrum of a subsampling of feature maps from the second layer of the network.
2. High frequency information is relevant to maintain decent classification.

Filtering in the higher layers



1. The FFT Spectrum of a subsampling of feature maps from the second layer of the network.
2. High frequency information is relevant to maintain decent classification.

Filtering at the layers of the network does not make sense since spatial information is not perserved.

Learning the Filter Parameters

Minimizing accuracy loss

- Can we learn the filter parameters so that we gain low-pass filtering without a significant degradation in accuracy?

Minimizing accuracy loss

- Can we learn the filter parameters so that we gain low-pass filtering without a significant degradation in accuracy?
- Regularization
  1. L_∞ regularization of the depthwise convolution layer

Minimizing accuracy loss

- Can we learn the filter parameters so that we gain low-pass filtering without a significant degradation in accuracy?
- Regularization
  1. L_∞ regularization of the depthwise convolution layer
  2. Total Variation (TV) regularization of the Layer 1 feature maps

Minimizing accuracy loss

- Can we learn the filter parameters so that we gain low-pass filtering without a significant degradation in accuracy?
- Regularization
  1. L_∞ regularization of the depthwise convolution layer
  2. Total Variation (TV) regularization of the Layer 1 feature maps
  3. Generalized Tikhonov regularization of the Layer 1 feature maps
- Train all models with Adam with learning rate 10^{-3}

L_∞ Regularization

$$\min \quad \frac{\alpha}{K} \sum_{j=1}^{K} \| W_{depthwise}[:, :, j] \|_\infty + \ell(f_\theta(x, y))$$

L_∞ Regularization

$$\min \quad \frac{\alpha}{K} \sum_{j=1}^{K} \| W_{depthwise}[:, :, j] \|_\infty + \ell(f_\theta(x, y))$$

L_∞ norm is a natural choice for the depthwise weights. This will ensure that the weights in the kernel take similar values to each much like a low pass filter.

$$\mathrm{K} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Total Variation Regularization

$$TV(x) = \sum_{i,j} |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|.$$

Total Variation Regularization

$$TV(x) = \sum_{i,j} |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|.$$

$$\min \quad \alpha_{TV} \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{i=1}^{K} TV(\mathcal{F}[i,:,:,k]) + \ell(f_\theta(x,y)),$$

Total Variation Regularization

$$TV(x) = \sum_{i,j} |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|.$$

$$\min \quad \alpha_{TV} \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{i=1}^{K} TV(\mathcal{F}[i,:,:,k]) + \ell(f_\theta(x,y)),$$

TV encourages the neighboring values in the feature maps to be similar so the high spike introduced by RP_2 would be diminished.

Tikhonov Regularization

$$\min \ \mu ||L \cdot w_0||_2 + \ell(f_\theta(x, y, w_0)),$$

Tikhonov Regularization

$$\min \ \mu ||L \cdot w_0||_2 + \ell(f_\theta(x, y, w_0)),$$

$$\min \ \alpha_{hf} \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{i=1}^{K} ||L_{hf} \cdot \mathcal{F}[i, :, :, k]||_2 + \ell(f_\theta(x, y))$$

L is a regularization operator $L_{hf} = I - L_{avg}$, defense is referred as Tik_{hf}

Tikhonov Regularization

$$\min \ \mu ||L \cdot w_0||_2 + \ell(f_\theta(x, y, w_0)),$$

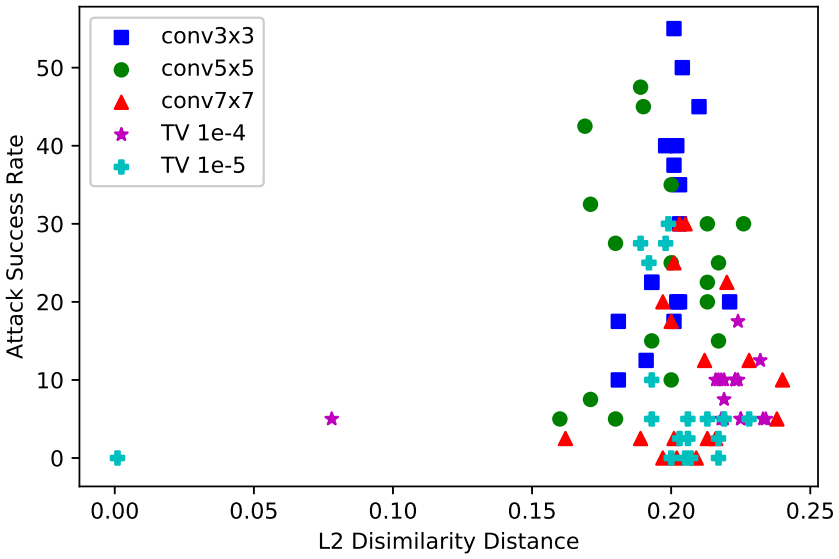
$$\min \ \alpha_{hf} \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{i=1}^{K} ||L_{hf} \cdot \mathcal{F}[i, :, :, k]||_2 + \ell(f_\theta(x, y))$$

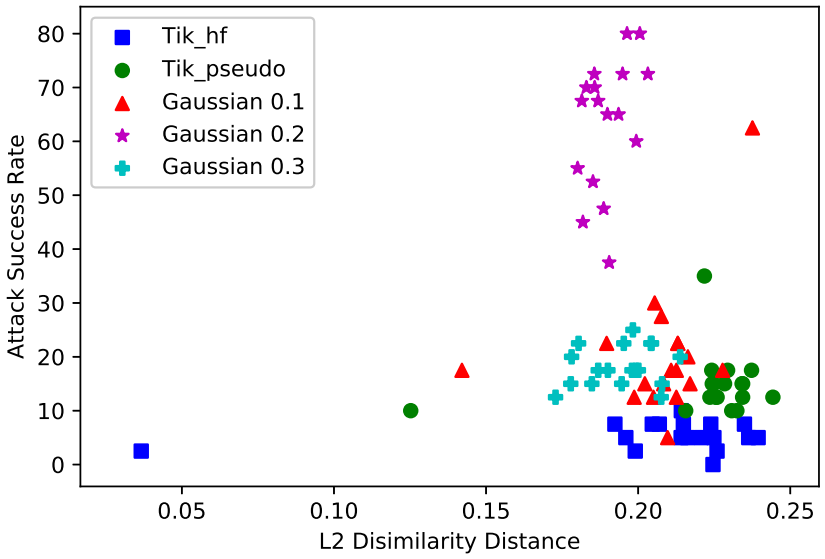
L is a regularization operator $L_{hf} = I - L_{avg}$, defense is referred as Tik_{hf}

$$\min \ \alpha_{pseudo} \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{i=1}^{K} ||L^{+}_{diff} \cdot \mathcal{F}[i, :, :, k]||_2 + \ell(f_\theta(x, y))$$

We refer to this defense as Tik_{pseudo}

Whitebox Evaluation





Total Variation Regularization and both variants of Tikhonov Regularization are able to perform well against a white-box RP_2 adversary.

Whitebox Evaluation

Table 2: Results from white box evaluation

	α	**Legitimate Acc.**	**Average Success Rate**	**Worst Success Rate**	L_2 **Dissimilarity**
Baseline	0	91%	49.18%	90%	0.207
Gaussian aug ($\sigma = 0.1$)	-	84.3%	19.44%	62.5%	0.238
Gaussian aug ($\sigma = 0.2$)	-	84.4%	55.97%	80%	0.196
Gaussian aug ($\sigma = 0.3$)	-	85.6%	21.39%	25%	0.198
Rand. sm ($\sigma = 0.1$)	-	84.3%	19.3%	67.5%	0.236
Rand. sm ($\sigma = 0.2$)	-	84.4%	55%	70%	0.189
Rand. sm ($\sigma = 0.3$)	-	85.6%	22.5%	**22.5%**	0.198
Adv-train	-	77.9%	11.94%	**20%**	0.244
3x3 conv	10^{-5}	86.3%	30%	55%	0.201
5x5 conv	0.1	86.3%	24.11%	47.5%	0.189
7x7 conv	0.1	87%	11.61%	30%	0.203
TV	10^{-4}	85.6%	7.92%	**17.5%**	0.224
TV	10^{-5}	82.3%	8.47%	22.5%	0.199
Tik_{hf}	10^{-4}	84.5%	5.416%	**10%**	0.214
Tik_{pseudo}	10^{-6}	83.6%	13.9%	35%	0.222

TV and Tik_{hf} are able to outperform norm-bound defenses like adversarial training and randomized smoothing.

Adaptive Attacks

Include the regularizer into attacker's loss function to reveal knowlege of defense

Low Frequency attack: Basic idea is restrict the search space of adversary so perturbations added are low-frequency noise

$$\arg\min_{\delta} \lambda \|M_x \cdot \delta\|_p + \ell(f_\theta(x_i + T_i(IDCT(M_{dim} \cdot DCT(M_x \cdot \delta)))), y^*)$$

Adaptive Attacks

$$\arg\min_{\delta} \ \ell_{adv} + \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{i=1}^{K} TV(\mathcal{F}[i, :, :, k])$$

$$\arg\min_{\delta} \ \ell_{adv} + \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{i=1}^{K} ||L_{hf} \cdot \mathcal{F}[i, :, :, k]||_2,$$

$$\arg\min_{\delta} \ \ell_{adv} + \frac{1}{N \cdot K} \sum_{i=1}^{N} \sum_{i=1}^{K} ||L^{+}_{diff} \cdot \mathcal{F}[i, :, :, k]||_2.$$

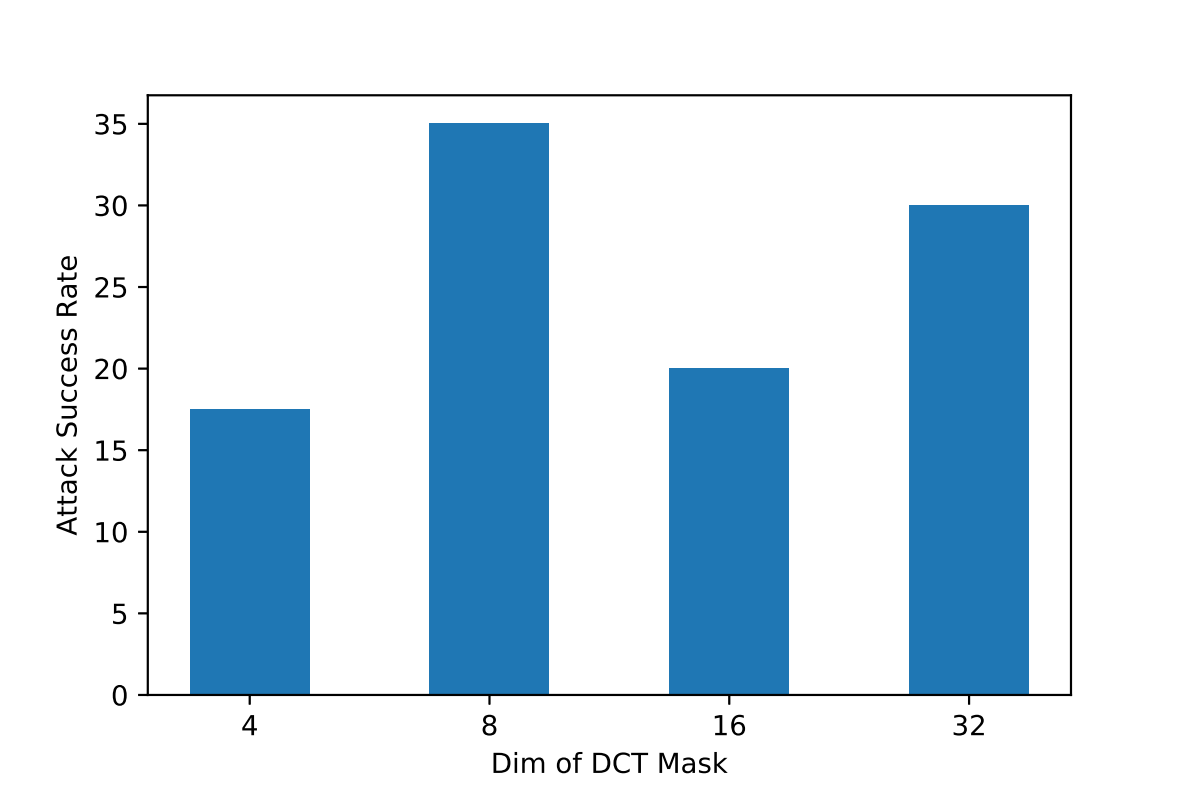
Add the regularizer for TV, Tik_{hf}, Tik_{pseudo} to the attacker's loss function and rerun evaluation.

Adaptive Attack Evalution

	Average Success Rate	**Worst Success Rate**	L_2 **Dissimilarity**
3x3 conv	22.91%	52.5%	0.546
5x5 conv	46.25%	75%	0.539
7x7 conv	10.416%	20%	0.539
TV (10^{-4})	8.33%	**20%**	0.044
TV (10^{-5})	6.11%	25%	0.046
Tik_{hf}	23.6%	47.5%	0.147
Tik_{pseudo}	17.5%	45%	0.141

Total Variation Regularization is a robust against physical-based attackers like RP_2.

Changing the Mask Dimension



Success rate after increased only by 5% after changing mask dimension for conv7x7 showing low pass filtering feature maps is a promising defense against physical attacks.

Concluding Remarks and Future Work

Conclusion/Future Work

- Spectral analysis of the feature maps and RP_2 attacks introduce high-frequency components

Conclusion/Future Work

- Spectral analysis of the feature maps and RP_2 attacks introduce high-frequency components
- Adding low-pass filters after the first layers provides robustness benefits

Conclusion/Future Work

- Spectral analysis of the feature maps and RP_2 attacks introduce high-frequency components
- Adding low-pass filters after the first layers provides robustness benefits
- Introduced multiple loss penalties to induce low pass filtering

Future Work

Conclusion/Future Work

- Spectral analysis of the feature maps and RP_2 attacks introduce high-frequency components
- Adding low-pass filters after the first layers provides robustness benefits
- Introduced multiple loss penalties to induce low pass filtering

Future Work

- Generalize this idea to larger datasets and models

Conclusion/Future Work

- Spectral analysis of the feature maps and RP_2 attacks introduce high-frequency components
- Adding low-pass filters after the first layers provides robustness benefits
- Introduced multiple loss penalties to induce low pass filtering

Future Work

- Generalize this idea to larger datasets and models
- Combine our regularization terms with methods like adversarial training

Published at: DSN Workshop in Dependable and Secure Machine Learning.
Link: `https://arxiv.org/abs/1908.02256`.

Thank you for listening!