Feature-Guided Black-Box Safety Testing of Deep Neural Networks

Matthew Wicker, Xiaowei Huang, and Marta Kwiatkowska

TACAS 2018

Table of Contents

1. Introduction
2. Preliminaries
3. Safety Against Manipulations based on Human Perception
4. Monte Carlo Tree Search for Asymptotically Optimal Strategy
5. Experimental Results

Table of Contents

1. Introduction
2. Preliminaries
3. Safety Against Manipulations based on Human Perception
4. Monte Carlo Tree Search for Asymptotically Optimal Strategy
5. Experimental Results

Introduction

Deep neural networks (DNNs or networks, for simplicity) have been developed for a variety of tasks, including malware detection, abnormal network activity detection, and self-driving cars.

An **adversarial example** is an input which, though initially classified correctly, is misclassified after a minor, perhaps imperceptible, perturbation.

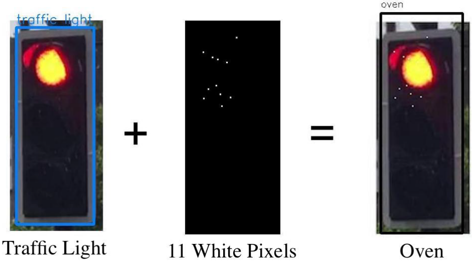


Figure: An adversarial example.

Introduction

They propose a *feature-guided* approach to test the resilience of image classifier networks against adversarial examples.

A key observation of this paper is that feature extraction methods enable us to identify elements of an image which are most vulnerable to a visual system such as a CNN.

They use the scale-invariant feature transform approach, or SIFT, to detect features.

Using the SIFT features, whose number is much smaller than the number of pixels, they represent the image as a two-dimensional Gaussian mixture model.

Introduction

We formulate the process of crafting adversarial examples as a two-player turn-based stochastic game, where *player I* selects features and *player II* then selects pixels within the selected features and a manipulation instruction.

After both players have made their choices, the image is modified according to the manipulation instruction, and the game continues.

While *player I* aims to minimise the distance to an adversarial example, *player II* can be cooperative, adversarial, or nature who samples the pixels according to the Gaussian mixture model.

Table of Contents

1. Introduction
2. Preliminaries
3. Safety Against Manipulations based on Human Perception
4. Monte Carlo Tree Search for Asymptotically Optimal Strategy
5. Experimental Results

Preliminaries

Let N be a network with a set C of classes.

Given an input α and a class $c \in C$, we use $N(\alpha, c)$ to denote the confidence of N believing that is in class c.

They write $N(\alpha) = \arg\max_{c \in C} N(\alpha, c)$ for the class into which N classifies c.

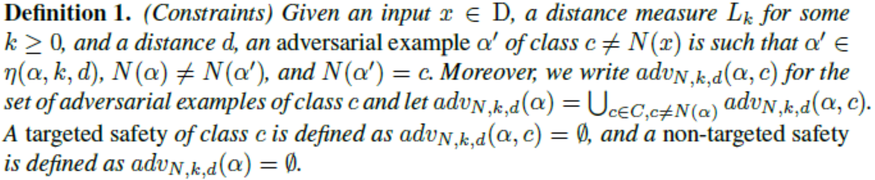
The metrics used to compute the distance between images typically include L_0, L_1 (Manhattan distance), L_2 (Euclidean distance), and L_∞.

The input domain D is a vector space, which can be represented as $\Re^{wxhxch}_{[0,255]}$ where w; h; ch are the width, height, and number of channels of an image, respectively

Preliminaries

They write $||\alpha_1 - \alpha_2||_k$ with $k \geq 0$ for the distance between two images α_1 and α_2 with respect to the L_k measurement.

Given an image α, a distance measure L_k, and a distance d, we define $\eta(\alpha, k, d) = \{\alpha' | ||\alpha' - \alpha||_k \leq d\}$ as the set of points whose distance to α is no greater than d with respect to L_k.



Definition 1. *(Constraints) Given an input $x \in \mathrm{D}$, a distance measure L_k for some $k \geq 0$, and a distance d, an* adversarial example *α' of class $c \neq N(x)$ is such that $\alpha' \in \eta(\alpha, k, d)$, $N(\alpha) \neq N(\alpha')$, and $N(\alpha') = c$. Moreover, we write $adv_{N,k,d}(\alpha, c)$ for the set of adversarial examples of class c and let $adv_{N,k,d}(\alpha) = \bigcup_{c \in C, c \neq N(\alpha)} adv_{N,k,d}(\alpha, c)$. A* targeted safety *of class c is defined as $adv_{N,k,d}(\alpha, c) = \emptyset$, and a* non-targeted safety *is defined as $adv_{N,k,d}(\alpha) = \emptyset$.*

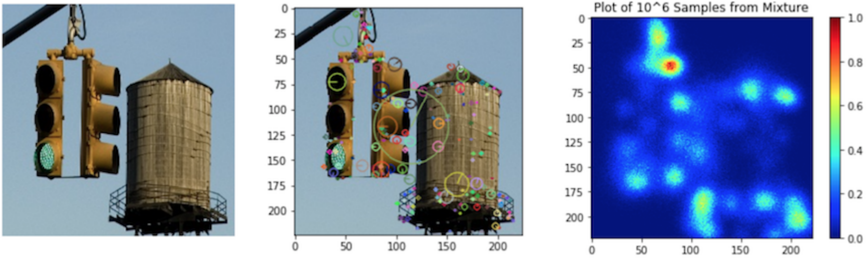
The SIFT algorithm proceeds through the following steps:

- scale-space extrema detection (detecting relatively darker or lighter areas in the image)
- keypoint localization (determining the exact position of these areas)
- keypoint descriptor assignment (understanding the context of the image w.r.t its local area).

Let $\Lambda(\alpha)$ be a set of features of the image α such that each feature $\lambda \in \Lambda(\alpha)$ is a tuple $(\lambda_x, \lambda_y, \lambda_s, \lambda_r)$, where (λ_x, λ_y) is the coordinate of the feature in the image, λ_s is the size of the feature, and λ_r is the response strength of the feature.

Preliminaries

Feature Extraction



(a) (b) (c)

Figure: Illustration of the transformation of an image into a saliency distribution. (a) The original image, provided by ImageNet. (b) The image marked with relevant keypoints $\Lambda(\alpha)$. (c) The heatmap of the Gaussian mixture model $G(\Lambda(\alpha))$.

Preliminaries

Distance w.r.t. Human Perception

Given an image α and its set $\Lambda(\alpha)$ of keypoints, we define for $\Lambda_i \, in \Lambda(\alpha)$ a two-dimensional Gaussian distribution G_i such that, for pixel $(p_x; p_y)$, we have

$$\mathcal{G}_{i,x} = \frac{1}{\sqrt{2\pi\lambda_{i,s}^2}} exp(\frac{-(p_x - \lambda_{i,x})^2}{2\lambda_{i,s}^2}) \quad \mathcal{G}_{i,y} = \frac{1}{\sqrt{2\pi\lambda_{i,s}^2}} exp(\frac{-(p_y - \lambda_{i,y})^2}{2\lambda_{i,s}^2}) \tag{1}$$

They define a set of weights $\Phi = \{\phi_i\}_{i \in \{1,2,\ldots,k\}}$ such that $k = |\Lambda(\alpha)|$ and $\phi_i = \lambda_{i,r} / \sum_{j=0}^{k} \lambda_{j,r}$.

Then, they can construct a Gaussian mixture model $G_x = \prod_{i=1}^{k} \phi_i * G_{i,x}$ and $G_y = \prod_{i=1}^{k} \phi_i * G_{i,y}$

Preliminaries

Pixel Manipulation

They write $(x; y; z)$ for the value of the $z-$ *channel* of the pixel positioned at $(x; y)$ on the image.

They define pixel manipulations $\delta_{X;i} : D \to D$ for $X \subseteq P_0$ a subset of input dimensions

$$\delta_{X,i}(\alpha)(x,y,z) = \begin{cases} \alpha(x,y,z) + \tau, \text{ if } (x,y) \in X \text{ and } i = + \\ \alpha(x,y,z) - \tau, \text{ if } (x,y) \in X \text{ and } i = - \\ \alpha(x,y,z) \qquad \text{otherwise} \end{cases}$$

for all pixels $(x; y)$ and channels $z \in \{1, 2, 3\}$.

Table of Contents

1. Introduction
2. Preliminaries
3. Safety Against Manipulations based on Human Perception
4. Monte Carlo Tree Search for Asymptotically Optimal Strategy
5. Experimental Results

Safety Against Manipulations based on Human Perception

Every image represents a point in the input vector space D.

Most existing investigations of the safety (or robustness) of DNNs focus on optimising the movement of a point along the gradient direction of some function obtained from the network.

Therefore, these approaches rely on the knowledge about the DNN.

Furthermore, the current understanding about the transferability of adversarial examples are all based on empirical experiments

A reasonable interpretation is that transferable adversarial examples, if they exist, do not rely on the gradient direction suggested by a network but instead may be specific to the input.

Safety Against Manipulations based on Human Perception

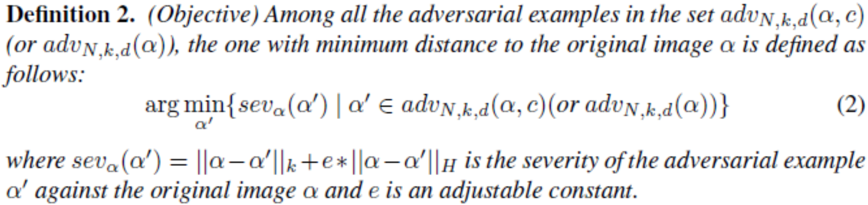
They propose a *feature-guided* approach which, instead of using the gradient direction as the guide for optimisation, it relies on targeting and manipulating image features as recognised by human perception capability.

Their method needs neither the knowledge about the network nor the necessity to massively sample the network for data to train a new network, and is therefore a *black-box* approach

Safety Against Manipulations based on Human Perception

Objective to Optimise

The sets $adv_{N;k;d}(\alpha; c)$ and $adv_{N;k;d}(\alpha)$ can be infinite. Therefore, we may be interested in finding a finite subset:



Definition 2. *(Objective) Among all the adversarial examples in the set $adv_{N,k,d}(\alpha,c)$ (or $adv_{N,k,d}(\alpha)$), the one with minimum distance to the original image α is defined as follows:*

$$\arg\min_{\alpha'}\{sev_\alpha(\alpha') \mid \alpha' \in adv_{N,k,d}(\alpha,c)(or\ adv_{N,k,d}(\alpha))\} \tag{2}$$

where $sev_\alpha(\alpha') = ||\alpha-\alpha'||_k + e*||\alpha-\alpha'||_H$ is the severity of the adversarial example α' against the original image α and e is an adjustable constant.

Safety Against Manipulations based on Human Perception

Crafting Adversarial Examples as a Two-Player Turn-Based Game

Assume two players I and II.
Let $M(\alpha, p, d) = (S \cup (Sx\Lambda(\alpha)), s_0, \{T_\alpha\}_{\alpha \in \{I,II\}}, L)$ be a **game model**, where S is a set of game states belonging to player I such that each state represents an image in $\eta(\alpha, k, d)$, and $Sx\Lambda(\alpha)$ is a set of game states belonging to player II.

They write $\alpha(s)$ for the image associated to the state $s \in S$. $s_0 \in S$ is the initial image state such that $\alpha(s_0)$ is the original image α.

The transition relation $T_I : Sx\Lambda(\alpha) \to Sx\Lambda(\alpha)$ is defined as,
$T_I(s, \lambda) = (s, \lambda)$

Transition relation $T_{II} : (Sx\Lambda(\alpha))xP(P_0)xI \to S$ is defined as,
$T_{II}((s, \lambda), X, i) = \delta_{X,i}(\alpha(s))$

Safety Against Manipulations based on Human Perception

Crafting Adversarial Examples as a Two-Player Turn-Based Game

Intuitively, on every game state $s \in S$ player I will choose a keypoint λ, and, in response to this, player II will choose a pair (X, i), where X is a set of input dimensions and i is a manipulation instruction.

The **labelling function** $L : S \cup (Sx\Lambda(\alpha)) \rightarrow CxG$ assigns to each state s or $(s; \lambda)$ a class $N(\alpha(s))$ and a two-dimensional Gaussian mixture model $G(\Lambda(\alpha(s)))$.

A **path** (or game play) of the game model is a sequence $s_1 u_1 s_2 u_2 \dots$ of game states such that, for all $k \geq 1$, we have $u_k = T_I(s_k, \lambda_k)$ for some feature λ_k and $s_{k=1} = T_{II}((s_k, \lambda_k), X_k, i_k)$ for some (X_k, i_k).

Safety Against Manipulations based on Human Perception

Crafting Adversarial Examples as a Two-Player Turn-Based Game

Let $last(\rho)$ be the last state of a finite path ρ and $Path_a^F$ be the set of finite paths such that $last(\rho)$ belongs to player $a \in \{I, II\}$.

A **stochastic strategy** $\sigma_I : Path_I^F \rightarrow D(\Lambda(\alpha))$ of player I maps each finite paths to a distribution over the next actions, and similarly for $\sigma_{II} : Path_{II}^F \rightarrow D(P(P_0)xI)$ for player II.

They call $\sigma = (\sigma_I, \sigma_{II})$ a **strategy profile.**

Safety Against Manipulations based on Human Perception

Crafting Adversarial Examples as a Two-Player Turn-Based Game

They define a **reward** $R(\sigma, \rho)$ for a given strategy profile $\sigma = (\sigma_I, \sigma_{II})$ and a finite path $\rho \in \bigcup_{a \in \{I, II\}} Path_a^F$.

Let $\alpha'_\rho = \alpha(last(\rho))$ be the image associated with the last state of the path ρ.

They write $t(\rho)$ for the expression $N(\alpha'_\rho) = cV||\alpha'_\rho - \alpha||_k > d$ representing that the path has reached a state whose associated image either is in the target class c or lies outside the region $\eta(\alpha, k, d)$.

The path ρ can be terminated whenever $t(\rho)$ is satisfiable.

Safety Against Manipulations based on Human Perception

Crafting Adversarial Examples as a Two-Player Turn-Based Game

Then they define the reward function $R(\sigma, \rho)$:

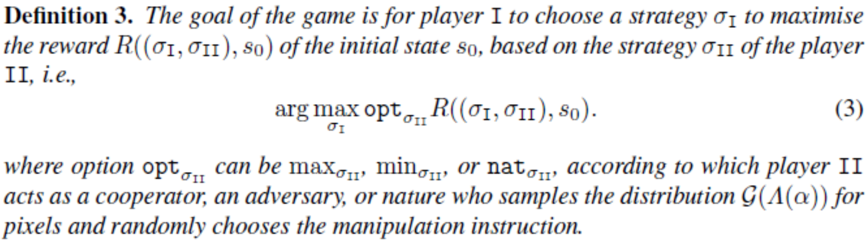
$$\begin{cases} 1/sev_{\alpha}(\alpha'_{\rho}) & \text{if } t(\rho) \text{ and } \rho \in Path_{\mathrm{I}}^{F} \\ \sum_{\lambda \in \Lambda(\alpha)} \sigma_{\mathrm{I}}(\rho)(\lambda) \cdot R(\sigma, \rho T_{\mathrm{I}}(last(\rho), \lambda)) & \text{if } \neg t(\rho) \text{ and } \rho \in Path_{\mathrm{I}}^{F} \\ \sum_{(X,i) \in \mathcal{P}(P_0) \times I} \sigma_{\mathrm{II}}(\rho)(X, i) \cdot R(\sigma, \rho T_{\mathrm{II}}(last(\rho), X, i)) & \text{if } \rho \in Path_{\mathrm{II}}^{F} \end{cases}$$

where $\sigma_I(\rho)(\lambda)$ is the probability of selecting λ on ρ by player I and $\sigma_{II}(\rho)(X, i)$ is the probability of selecting (X, i) based on ρ by player II.

Note that a path only terminates on player I states.

Safety Against Manipulations based on Human Perception

Crafting Adversarial Examples as a Two-Player Turn-Based Game

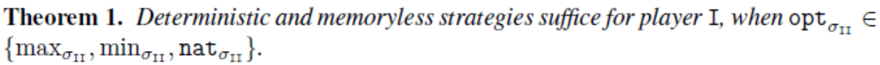


Definition 3. *The goal of the game is for player* I *to choose a strategy* σ_{I} *to maximise the reward* $R((\sigma_{\mathrm{I}}, \sigma_{\mathrm{II}}), s_0)$ *of the initial state* s_0*, based on the strategy* σ_{II} *of the player* II*, i.e.,*

$$\arg\max_{\sigma_{\mathrm{I}}} \mathtt{opt}_{\sigma_{\mathrm{II}}} R((\sigma_{\mathrm{I}}, \sigma_{\mathrm{II}}), s_0). \tag{3}$$

where option $\mathtt{opt}_{\sigma_{\mathrm{II}}}$ *can be* $\max_{\sigma_{\mathrm{II}}}$, $\min_{\sigma_{\mathrm{II}}}$, *or* $\mathtt{nat}_{\sigma_{\mathrm{II}}}$, *according to which player* II *acts as a cooperator, an adversary, or nature who samples the distribution* $\mathcal{G}(\Lambda(\alpha))$ *for pixels and randomly chooses the manipulation instruction.*

A strategy σ is deterministic if $\sigma(\rho)$ is a Dirac distribution, and is memoryless if $\sigma(\rho) = \sigma(\mathit{last}(\rho))$ for all finite paths ρ. We have the following result:



Theorem 1. *Deterministic and memoryless strategies suffice for player* I*, when* $\mathtt{opt}_{\sigma_{\mathrm{II}}} \in \{\max_{\sigma_{\mathrm{II}}}, \min_{\sigma_{\mathrm{II}}}, \mathtt{nat}_{\sigma_{\mathrm{II}}}\}$.

Table of Contents

1 Introduction

2 Preliminaries

3 Safety Against Manipulations based on Human Perception

4 Monte Carlo Tree Search for Asymptotically Optimal Strategy

5 Experimental Results

Monte Carlo Tree Search for Asymptotically Optimal Strategy

They present an approach based on Monte Carlo tree search (MCTS) to find an optimal strategy asymptotically.

Consider the case of $opt_{\sigma_{II}} = max_{\sigma_{II}}$.

An MCTS algorithm, gradually expands *a partial game tree* by sampling the strategy space of the model $M(\alpha, p, d)$.

MCTS has a *theoretical guarantee* that it converges to optimal solution when the game tree is fully explored. (Exploration-Exploitation tradeoff)

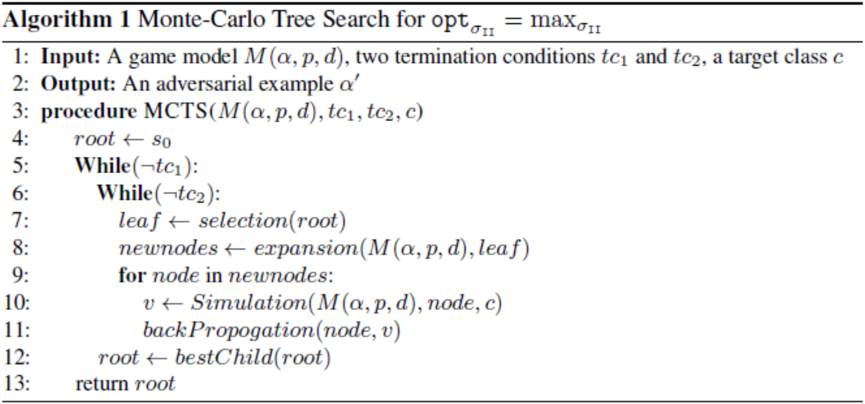
Monte Carlo Tree Search for Asymptotically Optimal Strategy

They use two termination conditions tc_1 and tc_2 to control the pace of the algorithm. The terminating conditions can be, e.g., bounds on the or the number of iterations, etc.

tc_1 controls whether the entire procedure should be terminated, and tc_2 controls when a move should be made.

On the partial tree, every node maintains a pair (r, n), which represents the accumulated reward and the number of visits, respectively.

Monte Carlo Tree Search for Asymptotically Optimal Strategy



Algorithm 1 Monte-Carlo Tree Search for $\mathrm{opt}_{\sigma_{\mathrm{II}}} = \max_{\sigma_{\mathrm{II}}}$

```
Input: A game model $M(\alpha, p, d)$, two termination conditions $tc_1$ and $tc_2$, a target class $c$
Output: An adversarial example $\alpha'$
procedure MCTS($M(\alpha, p, d), tc_1, tc_2, c$)
    $root \leftarrow s_0$
    While($\neg tc_1$):
        While($\neg tc_2$):
            $leaf \leftarrow selection(root)$
            $newnodes \leftarrow expansion(M(\alpha, p, d), leaf)$
            for $node$ in $newnodes$:
                $v \leftarrow Simulation(M(\alpha, p, d), node, c)$
                $backPropogation(node, v)$
        $root \leftarrow bestChild(root)$
    return $root$
```

Monte Carlo Tree Search for Asymptotically Optimal Strategy

The **selection** travels from the root to a leaf according to an exploration-exploitation balance.

After **expanding** the children of the leaf node, we call **Simulation** to run simulation on every child node.

Every simulation terminates when reaching a terminated node α', on which a reward $1/sev(\alpha')$ can be computed.
This reward is then **backpropagated** from the new child node through its ancestors until reaching the root.

Every time a new reward v is backpropogated through a node, we update its associated pair to $(r + v, n + 1)$. The $bestChild(root)$ returns the child of root who has the highest value of the expression r/n.

Monte Carlo Tree Search for Asymptotically Optimal Strategy

Severity Interval from the Game

There exists a severity interval $SI(\alpha, p, d)$ with respect to the role of player II:

$$[sev(M(\alpha, p, d), max_{\sigma_{II}}), sev(M(\alpha, p, d), min_{\sigma_{II}})]$$

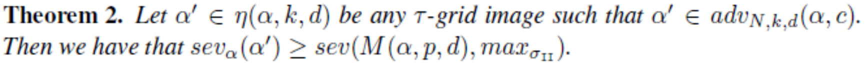
We have that $sev(M(\alpha, p, d), nat_{\sigma_{II}})$

Monte Carlo Tree Search for Asymptotically Optimal Strategy

Safety Guarantee via Optimal Strategy

An image $\alpha' \in \eta(\alpha, k, d)$ is a τ-grid image if for all dimensions $p \in P_0$ we have $|\alpha'(p) - \alpha(p)| = n * \tau$ for some $n \geq 0$.

Let $\tau(\alpha, k, d)$ be the set of τ-grid images in $\eta(\alpha, k, d)$. We have the following conclusion for the case when player II is cooperative:

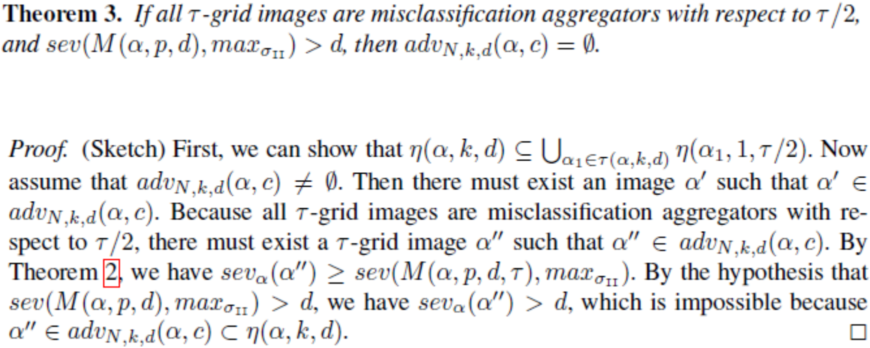


Theorem 2. *Let $\alpha' \in \eta(\alpha, k, d)$ be any τ-grid image such that $\alpha' \in adv_{N,k,d}(\alpha, c)$. Then we have that $sev_\alpha(\alpha') \geq sev(M(\alpha, p, d), max_{\sigma_{II}})$.*

The idea of the proof is to show that every τ-grid image can be reached by some game play.

Monte Carlo Tree Search for Asymptotically Optimal Strategy

Safety Guarantee via Optimal Strategy

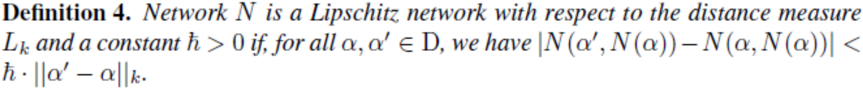


Theorem 3. *If all τ-grid images are misclassification aggregators with respect to $\tau/2$, and $sev(M(\alpha, p, d), max_{\sigma_{\mathrm{II}}}) > d$, then $adv_{N,k,d}(\alpha, c) = \emptyset$.*

Proof. (Sketch) First, we can show that $\eta(\alpha, k, d) \subseteq \bigcup_{\alpha_1 \in \tau(\alpha,k,d)} \eta(\alpha_1, 1, \tau/2)$. Now assume that $adv_{N,k,d}(\alpha, c) \neq \emptyset$. Then there must exist an image α' such that $\alpha' \in adv_{N,k,d}(\alpha, c)$. Because all τ-grid images are misclassification aggregators with respect to $\tau/2$, there must exist a τ-grid image α'' such that $\alpha'' \in adv_{N,k,d}(\alpha, c)$. By Theorem 2, we have $sev_\alpha(\alpha'') \geq sev(M(\alpha, p, d, \tau), max_{\sigma_{\mathrm{II}}})$. By the hypothesis that $sev(M(\alpha, p, d), max_{\sigma_{\mathrm{II}}}) > d$, we have $sev_\alpha(\alpha'') > d$, which is impossible because $\alpha'' \in adv_{N,k,d}(\alpha, c) \subset \eta(\alpha, k, d)$. □

Monte Carlo Tree Search for Asymptotically Optimal Strategy

Safety Guarantee via Optimal Strategy



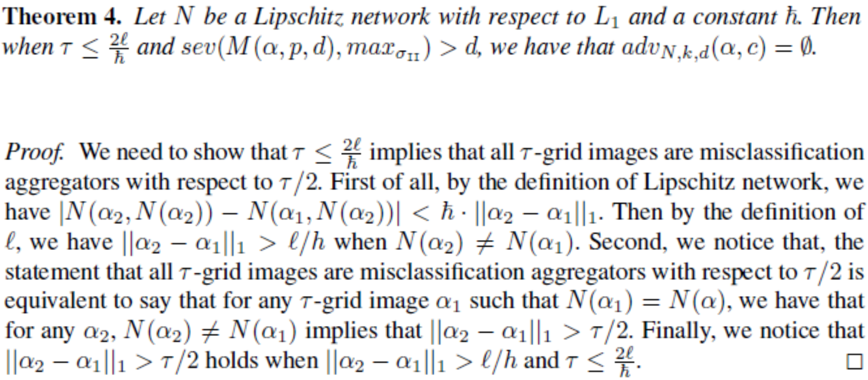
Definition 4. *Network N is a Lipschitz network with respect to the distance measure L_k and a constant $\hbar > 0$ if, for all $\alpha, \alpha' \in \mathrm{D}$, we have $|N(\alpha', N(\alpha)) - N(\alpha, N(\alpha))| < \hbar \cdot ||\alpha' - \alpha||_k$.*

Note that all networks whose inputs are bounded, including all image classification networks we studied, are Lipschitz networks.

$$l = min\{|N(\alpha', N(\alpha)) - N(\alpha, N(\alpha))||\alpha, a\alpha' \in D, N(\alpha' \neq N(\alpha)\}$$

Monte Carlo Tree Search for Asymptotically Optimal Strategy

Safety Guarantee via Optimal Strategy



Theorem 4. *Let N be a Lipschitz network with respect to L_1 and a constant $\hbar$. Then when $\tau \leq \frac{2\ell}{\hbar}$ and $sev(M(\alpha, p, d), max_{\sigma_{\text{II}}}) > d$, we have that $adv_{N,k,d}(\alpha, c) = \emptyset$.*

Proof. We need to show that $\tau \leq \frac{2\ell}{\hbar}$ implies that all τ-grid images are misclassification aggregators with respect to $\tau/2$. First of all, by the definition of Lipschitz network, we have $|N(\alpha_2, N(\alpha_2)) - N(\alpha_1, N(\alpha_2))| < \hbar \cdot ||\alpha_2 - \alpha_1||_1$. Then by the definition of ℓ, we have $||\alpha_2 - \alpha_1||_1 > \ell/h$ when $N(\alpha_2) \neq N(\alpha_1)$. Second, we notice that, the statement that all τ-grid images are misclassification aggregators with respect to $\tau/2$ is equivalent to say that for any τ-grid image α_1 such that $N(\alpha_1) = N(\alpha)$, we have that for any α_2, $N(\alpha_2) \neq N(\alpha_1)$ implies that $||\alpha_2 - \alpha_1||_1 > \tau/2$. Finally, we notice that $||\alpha_2 - \alpha_1||_1 > \tau/2$ holds when $||\alpha_2 - \alpha_1||_1 > \ell/h$ and $\tau \leq \frac{2\ell}{\hbar}$. □

Table of Contents

1 Introduction

2 Preliminaries

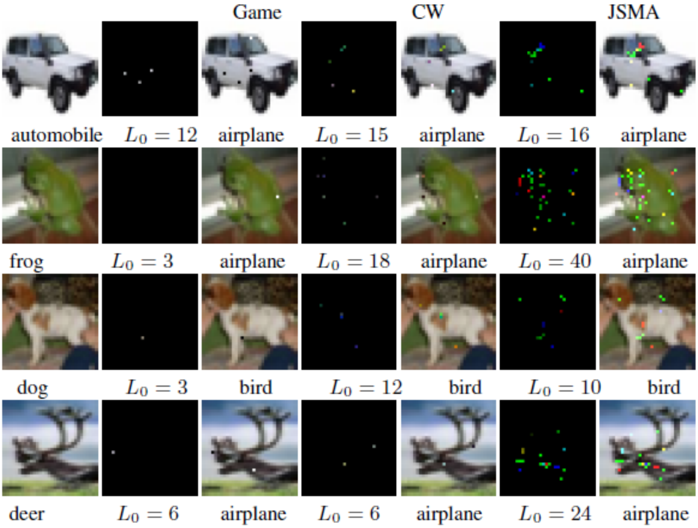
3 Safety Against Manipulations based on Human Perception

4 Monte Carlo Tree Search for Asymptotically Optimal Strategy

5 Experimental Results

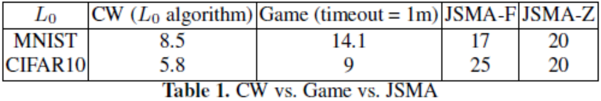
Experimental Results

Comparison with Existing Approaches



Experimental Results

Comparison with Existing Approaches



L_0	CW (L_0 algorithm)	Game (timeout = 1m)	JSMA-F	JSMA-Z
MNIST	8.5	14.1	17	20
CIFAR10	5.8	9	25	20

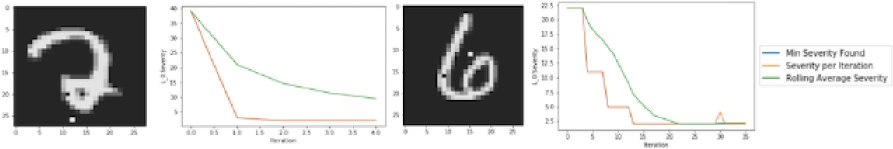
Table 1. CW vs. Game vs. JSMA

Experiments are conducted by setting the termination conditions $tc_1 = 20s$ and $tc_2 = 60s$ for every image.

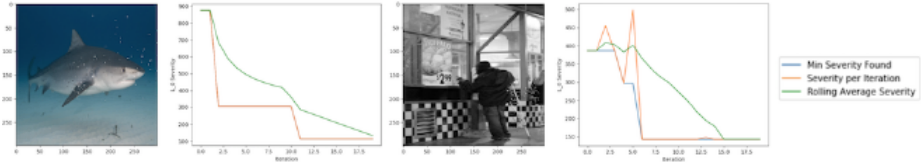
Note that JSMA needs several minutes to handle an image, and CW is 10 times slower than JSMA.

Experimental Results

Convergence in Limited Runs



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



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

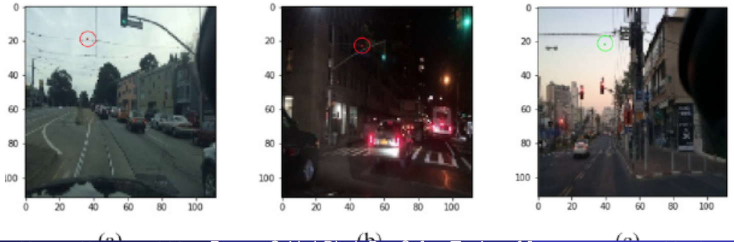
Experimental Results

Evaluating Safety-Critical Networks

They apply the method to a network used for classifying traffic light images collected from dashboard cameras. The Nexar traffic light challenge made over eighteen thousand dashboard camera images publicly available.

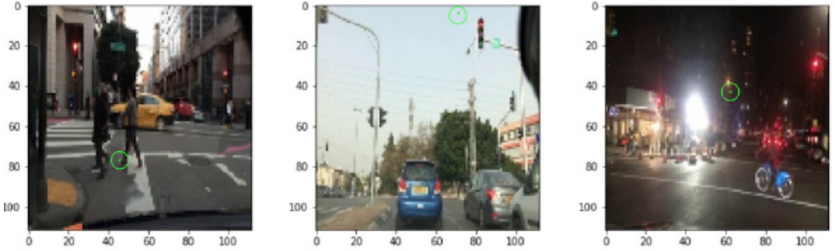
Each image is labeled either green, or red or null if there is no traffic light appearing in the image

They test the winner of the challenge which scored an accuracy above 90%.

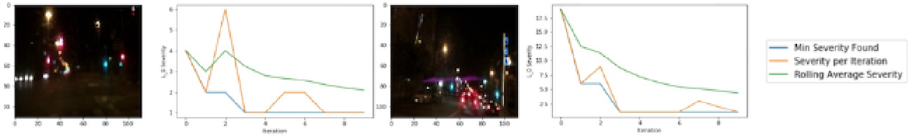


Experimental Results

Evaluating Safety-Critical Networks



(a) (b) (c)



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

Experimental Results

Counter-claim to Statements in [25]

A recent paper [25] argued that, under specific circumstances, there is no need to worry about adversarial examples because they are not invariant to changes in scale or angle in the physical domain.

This approach, which is inherently scale and rotationally invariant, can easily counter-claim such statements.

In [25] it is suggested that realistic camera movements – those which change the angle and distance of the viewer – reduce the phenomenon of adversarial examples to a curiosity rather than a safety concern.

This work shows that their adversarial examples, which are predicated on scale and rotationally invariant methods, defeat these claims.

Experimental Results

Counter-claim to Statements in [25]