Simple Transparent Adversarial Examples

Jaydeep Borkar[1,2] and Pin-Yu Chen[1]

[1]MIT-IBM Watson AI Lab, IBM Research

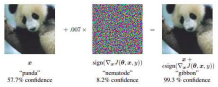
[2]University of Pune

MIT-IBM Watson AI Lab

Motivation

Conventional Adversarial Examples

- Need Machine Learning and Programming knowledge to craft examples.
  - Optimization
  - Knowledge about victim model (e.g. input gradient)
  - Excessive prediction evaluations (black-box attack)



Goodfellow et al., 2015

We Ask a Question!

Is it possible to attack Machine Learning models deployed as black-box APIs if the attacker doesn't have ML and programming skills?

✓ Our work says **YES!**

Our New Simple Black-box Attack

Simple Transparent Adversarial Examples

Using our attack, it is possible to fool publicly deployed cloud vision APIs without any ML and Programming knowledge

Google Cloud Vision API

Microsoft

PICPURIFY

sightengine

Tasks

- Object detection
- Optical Character Recognition (OCR)

Simple Transparent Adversarial Examples for OCR

Steps to Craft the Example

Select the region | Embed the text | Make text invisible | Final image | OCR output

Possible Risks and Applications

- Breaking blind review
- Attacking Check Scanner Apps
- Steganography

Tools used:

- imagecolorpicker
- photopea.

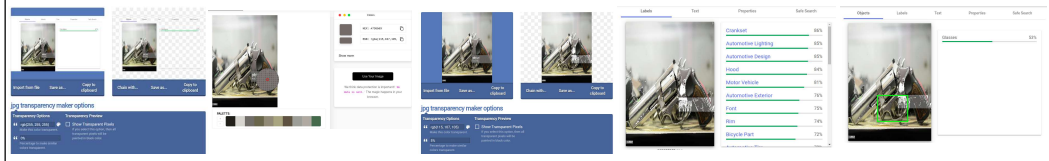
Table 1: Performance of Google Cloud Vision OCR against Simple Transparent Adversarial Examples

RGB for Region	Font color	Font size (px)	Min RGB difference
(214, 44, 11)	(204, 34, 1)	9	30
(74, 0, 3)	(54, 0, 3)	15	20
(255, 255, 255)	(245, 245, 245)	11	30
(236, 236, 236)	(226, 226, 226)	11	30
(34, 73, 120)	(30, 63, 110)	15	24
(212, 235, 249)	(202, 225, 239)	11	30
(200, 202, 189)	(170, 182, 160)	15	79
(69, 37, 23)	(59, 27, 18)	11	25
(126, 144, 162)	(120, 137, 155)	9	20
(171, 170, 186)	(163, 162, 178)	10	24

Table 1: RGB for Region denotes the RGB values of the region where the secret text is to be embedded. Font color and Font size refer to the RGB values and size of the text to be embedded. Min RGB difference indicates the minimum possible difference between the RBG values of the region and RGB values of the text, such that the embedded text in that region evades the vision of time-limited humans but is detected by Google Cloud Vision's OCR

Simple Transparent Adversarial Examples to Evade Object Detection

Steps to craft the example

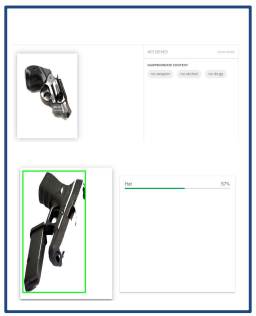


Step 1: Upload the image in the tool **Step 2**: Select the region **Step 3**: Specify transparency percentage and region. **Step 4**: Query the API **Step 5**: Check for lesser transparency (4% here)

This attack can help improve real-world robustness since it's:

- Query efficient
- More realistic attack scenario
- Goes beyond Lp norm
- Cheaper to carry out
- Doesn't need ML knowledge

Hence, can be easily carried out by anyone. Considering such simple attacks can help build more broader defenses.



Performance of APIs

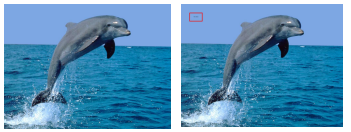
Table 1: Performance of APIs against Simple Transparent Adversarial Examples

API	ASR (%)	Samples Tested	Avg ϵ_1	Queries for ϵ_1	Avg ϵ_2	Queries for ϵ_2
Azure Cloud Vision	52%	50	39.37%	4	6.87%	3
Google Cloud Vision	36%	50	47.08%	5	6.71%	4
Sightengine	9%	50	43.25%	4	6%	2
Picpurify	2%	50	30%	3	NA	NA

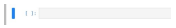
Table 1: ASR denotes the attack success rate. Samples Tested show the number of modified images tested. Avg ϵ_1 refers to the average modification constraint (i.e. transparency percentage) when we don't select any region ourselves to be patched and Avg ϵ_2 refers to the average modification constraint when we select region(s) ourselves to be patched. Queries for ϵ_1 and Queries for ϵ_2 denote the average number of queries requires to fool the APIs when the modification constraints are ϵ_1 and ϵ_2. NA indicates that the modified images did not evade the object detection of that specific API.

Samples tested 50

Examples that fool both humans and DNNs



Modified image

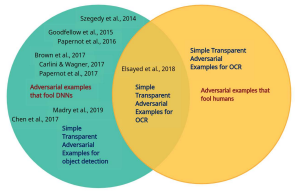


Google Cloud OCR output

➢ Fool the vision of both time-limited humans and DNNs powered Google Cloud OCR when the font size and RGB difference are very small

The closest line of work is by Elsayed et al., 2018

Venn Diagram



Goh et al., 2021

(Engstrom et al., 2019

Conclusion

- ❑ Robustness evaluation should cover the whole spectrum of semantically similar examples
- ❑ Though important, the current research has not focused on simple unconventional methods on evaluating the robustness.
- ❑ This needs the attention because simple methods can be used any anyone to attack deep neural networks
- ❑ Serious threats posed by such simple attacks should be considered to build more inclusive defenses