Black-box adversarial attack

Tung-Duong Mai
Minsoo Kang
Aitolkyn Baigutanova
Sanjarbek Rakhmonov

Outline

I - Problem

II - Baseline method & Limitations

III - Solution & Improvements

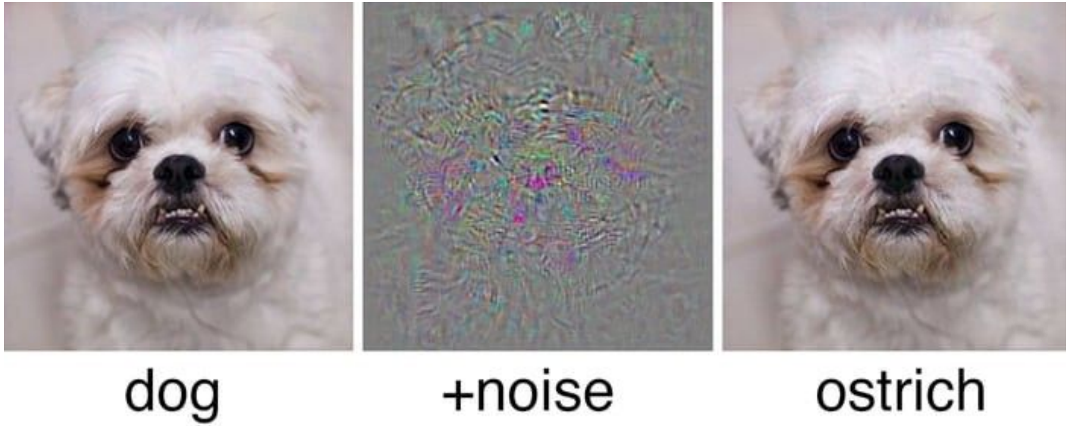
IV - Results & Evaluation

Problem

Devise a **black-box** **adversarial** attack that is **simple but effective** across **multiple domains.**

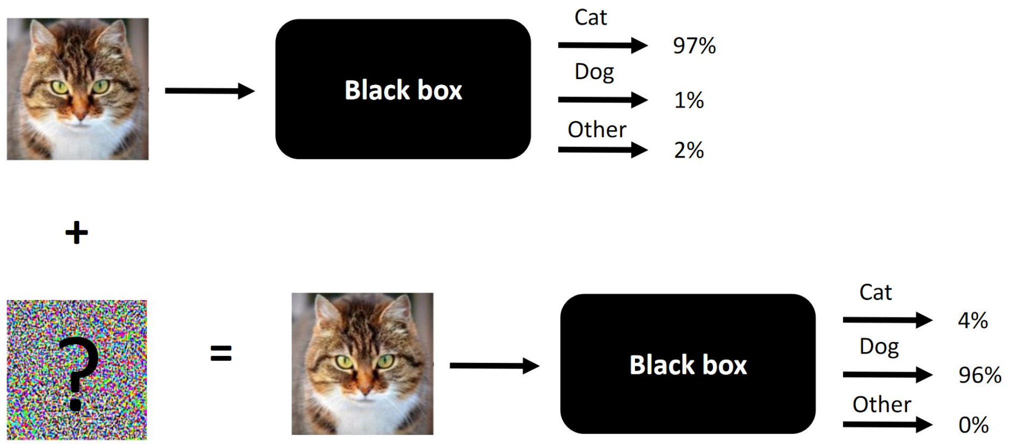
Problem: Adversarial attack

Adversarial attack is a machine learning technique attempting to fool models by supplying corrupted input. By inserting a small noise to the original input, which is undetectable to humans, a different output is produced by the network.



Problem: Blackbox

Attack done without knowing the internal structure of the model, which can only utilize the output as feedback, is called a **blackbox attack.**



Problem: simple & effective?

- Simple: search-based
- Effective: High attack success rate

  Small number of queries

  Small distortion from original input

  Simultaneously fuzzing

Why is adversarial attack important?

Adversarial inputs pose security risks to AI-based software. Generating and defending these tricky test cases helps improving the safety of the software.

Example: Autonomous cars are still vulnerable to adversarial input that may cause casualties.

(White image was possibly taken as open space.)

Baseline method

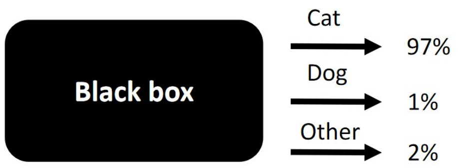
DeepSearch: A Simple and Effective Blackbox Attack for Deep Neural Networks

Paper method

1) Input : Original Image + Classifier + parameters

2) Feedback (Fitness): Classifier probability output

3) Method: Searching + Query reduction
+ Distortion (difference) reduction



Goal: Find input that will get wrong output + less evaluation
+ more similar

Paper method

Finding noise pattern for adversary

Simple hill climbing -ish method

- try out a step and stay if fitter

Hierarchical Grouping (to reduce query usage)

Iterative Refinement (to reduce distortion)

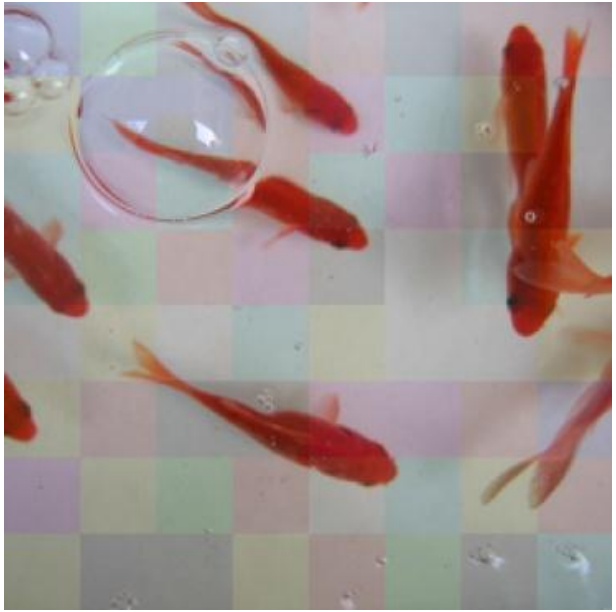
Paper method - shortcomings/limitations

Non targeted attack

Unnatural grouping

Restricted to image domain

Rely on prob outputs



Solution

Replicate the paper

Improve the method

Expand to a different domain

Replication

Re-implemented:

Borrowed:

Replication

Re-implemented:

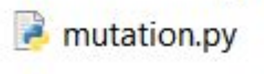
- DeepSearch algorithm (main algorithm)
  + Perturbation batching
- Hierarchical grouping

Borrowed:

Replication

Re-implemented:

- DeepSearch algorithm (main algorithm)
  + Perturbation batching
- Hierarchical grouping

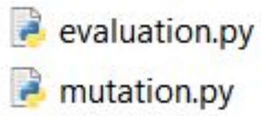


Borrowed:

Replication

Re-implemented:

- DeepSearch algorithm (main algorithm)
  + Perturbation batching
- Hierarchical grouping

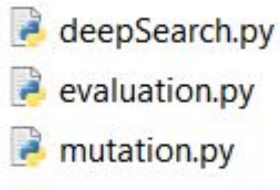


Borrowed:

Replication

Re-implemented:

- DeepSearch algorithm (main algorithm)
  + Perturbation batching
- Hierarchical grouping



Hill climbing

Borrowed:

Replication

Re-implemented:

- DeepSearch algorithm
  + Perturbation batching
- Hierarchical grouping

deepSearch.py
evaluation.py
mutation.py

Borrowed:

- The models (and datasets)
- Interfaces

borrowed for fair comparison of results.

imgntWrapper.py
madryCifarUndefWrapper.py
madryCifarWrapper.py
model.py
testDeepSearch.py

Replication

Readability was subjectively improved during the replication

Replication

Readability was subjectively improved during the replication

Before

LazierGreedy.py Line 44~64

```
44  def loss(image):
63  self.loss_fn=loss
64  self.loss=loss(self.image)
```

- Swapping method bonding to a variable bonding right after definition. (???)
- No comments or explanation at all.

Replication

Readability was subjectively improved during the replication

Before

LazierGreedy.py Line 44~64

```
44  def loss(image):
63  self.loss_fn=loss
64  self.loss=loss(self.image)
```

- Swapping method bonding to a variable bonding right after definition. (???)
- No comments or explanation at all.

to →

After (Separate example)

```
def group_generation(size = (3,3), group_size = 2, options = ""):
    """
    size: Size of the image to be divided into groups.
    group_size: Width of group if the group was square.
    options: Reserved parameter in case of other grouping patterns.

    This method outputs  a list with groups of indices. For example,
     _______________
    |0   |1   |  2|  3|     If this square was grouped into 4 pixels,
    |___|___|___|___|     [[0, 1, 4, 5],
    |4   |5   |  6|  7|      [2, 3, 6, 7],
    |___|___|___|___|      [8, 9, 12, 13],
    |    |    |    |    |      [10, 11, 14, 15]]
    |8__|9__|_10|_11|     will be the outcome.
    |    |    |    |    |
    |12_|13_|_14|_15|

    If the groups cannot be sized equally, the rightmost and bottom
    groups will be cropped."""
```

- Full documentation of how the algorithm works.
- Full length variable naming.

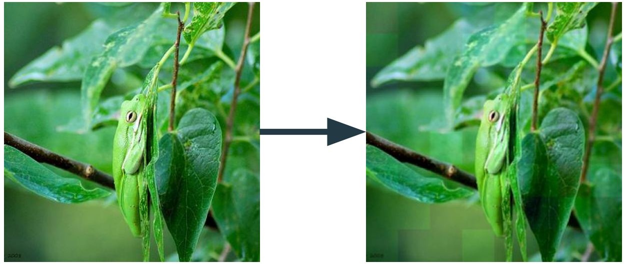
Improvement

New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- Alternative Grouping scheme

Improvement: Targeting

Research



<Tree frog>

<Garfish>
Anything else than tree frog

New features implemented:

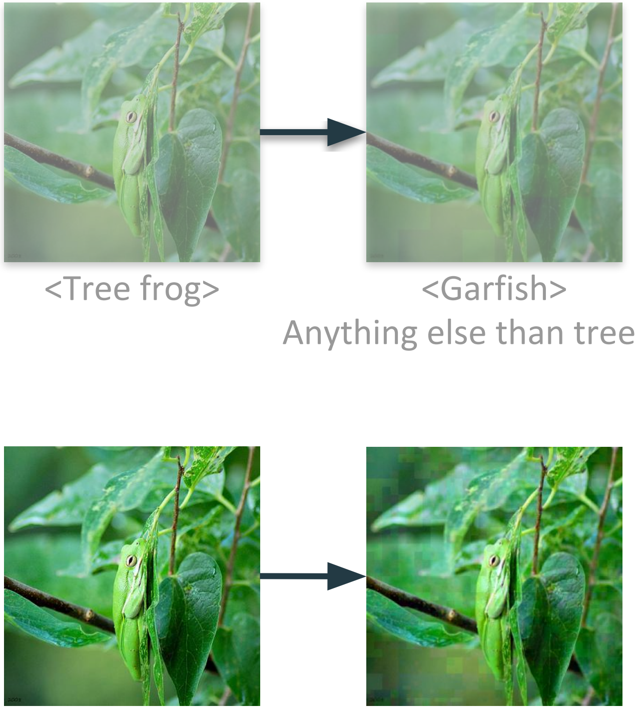
- **Targeted Attack**
- Categorical Feedback Attack
- Alternative Grouping scheme

Example on InceptionV3 classifier

Improvement: Targeting

New features implemented:

- **Targeted Attack**
- Categorical Feedback Attack
- Alternative Grouping scheme



Research

<Tree frog> <Garfish>

Anything else than tree frog

Our Project

<Tree frog> **<Street Sign>**

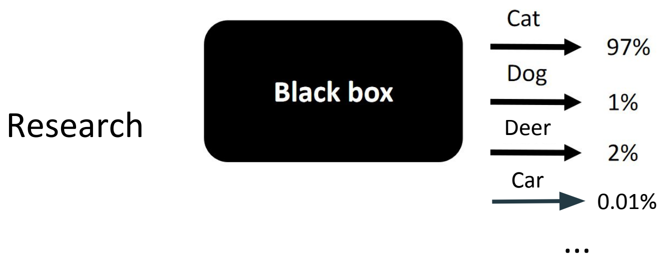
Specified class label

Example on InceptionV3 classifier

Improvement: Categorical

New features implemented:

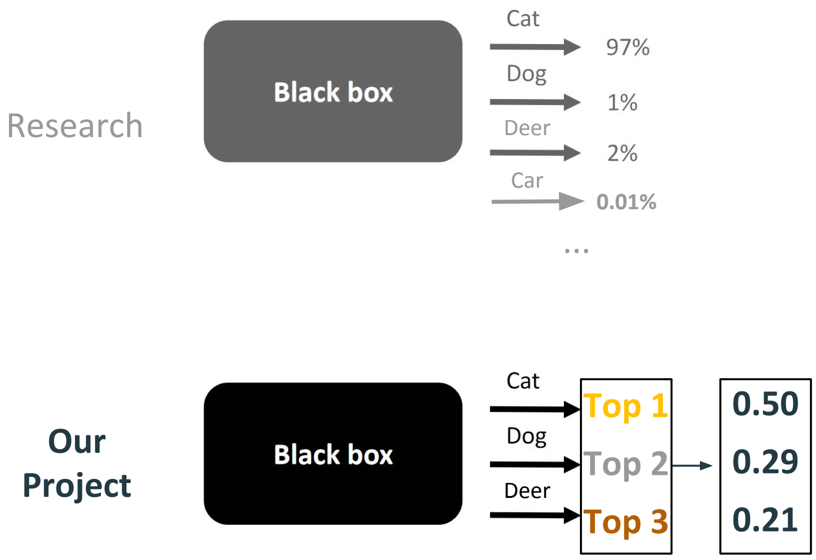
- Targeted Attack
- **Categorical Feedback Attack**
- Alternative Grouping scheme



Improvement: Categorical

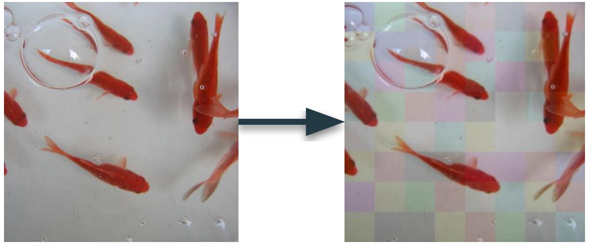
New features implemented:

- Targeted Attack
- **Categorical Feedback Attack**
- Alternative Grouping scheme



Improvement: Grouping

Research

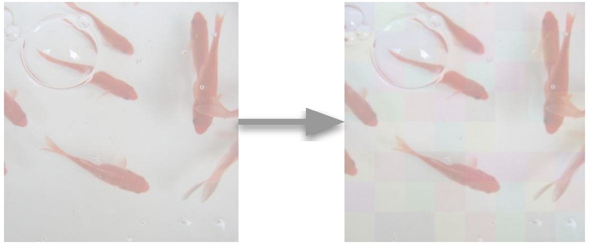


New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- **Alternative Grouping scheme**

Improvement: Grouping

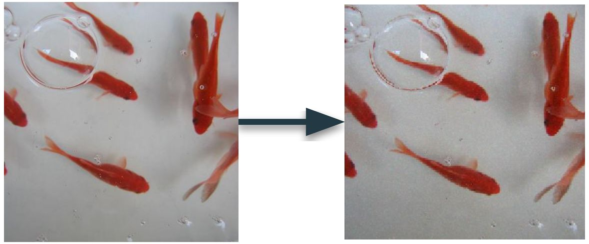
Research



New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- **Alternative Grouping scheme**

Random Grouping

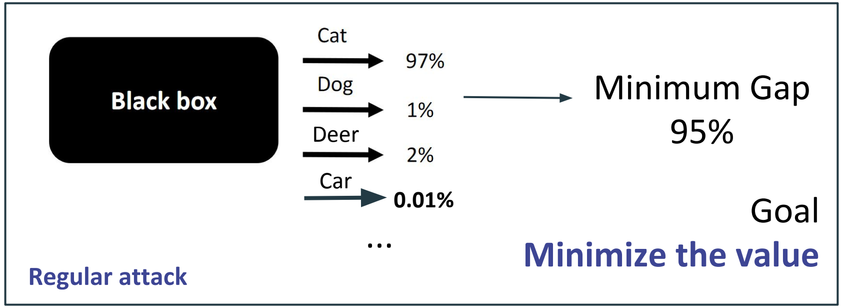


Improvement: Detail

New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- Alternative Grouping scheme

Changing focus:
<Probability gap> to **<Target probability>**

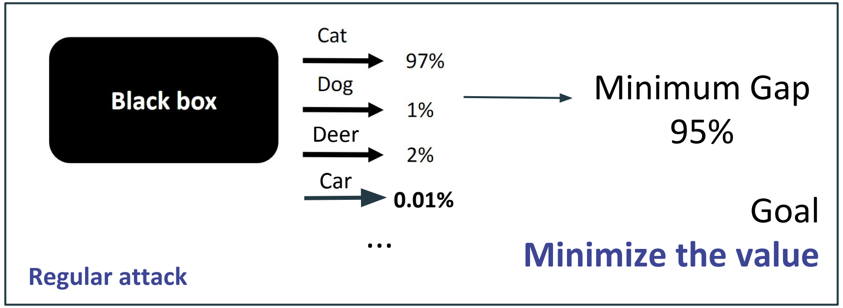


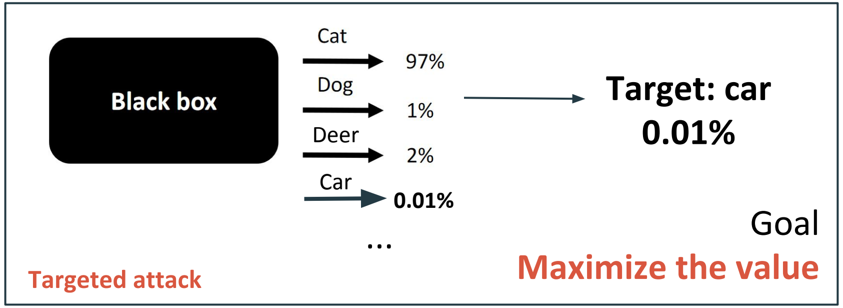
Improvement: Detail

New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- Alternative Grouping scheme

Changing focus:
<Probability gap> to <Target probability>





Improvement: Detail

New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- Alternative Grouping scheme

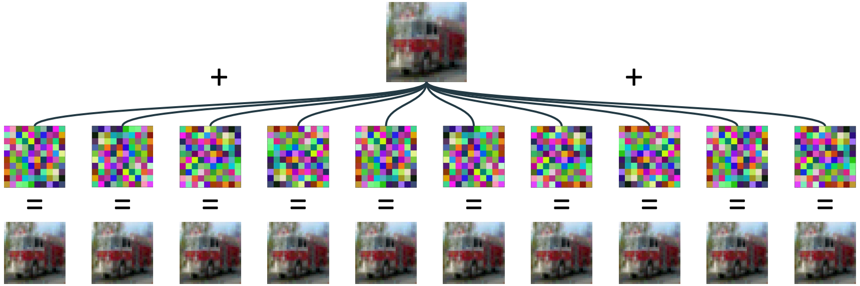
Category to probability: **Frequency distribution**
Repeatedly challenge the confidence

Improvement: Detail

New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- Alternative Grouping scheme

Category to probability: **Frequency distribution**
Repeatedly challenge the confidence

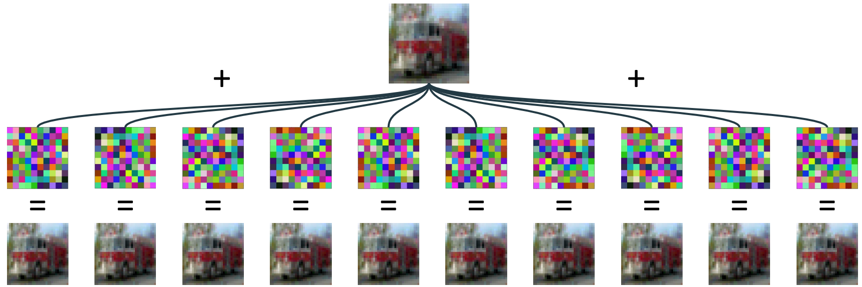


Improvement: Detail

New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- Alternative Grouping scheme

Category to probability: **Frequency distribution**
Repeatedly challenge the confidence



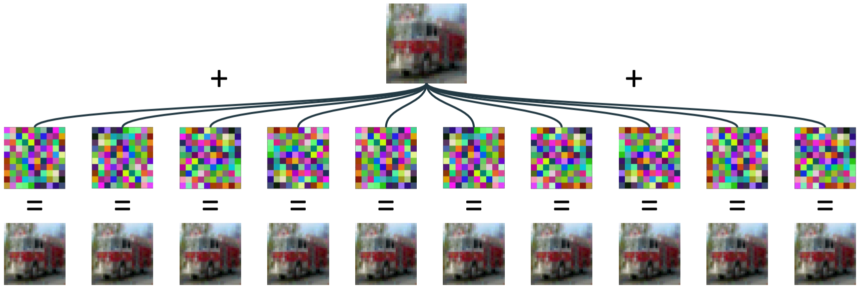
#1	8	9	9	9	1	9	9	8	9	9
#2	9	1	8	6	9	8	1	9	1	1
#3	0	0	1	1	6	0	7	1	8	6

Improvement: Detail

New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- Alternative Grouping scheme

Category to probability: **Frequency distribution**
Repeatedly challenge the confidence



#1	8	9	9	9	1	9	9	8	9	9
#2	9	1	8	6	9	8	1	9	1	1
#3	0	0	1	1	6	0	7	1	8	6

Weighted Frequency distribution

0	1	2	3	4	5	6	7	8	**9**
0.05	**0.233**	0	0	0	0	0.067	0.017	0.183	**0.45**

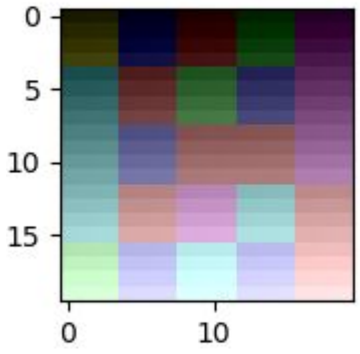
Class 9

Improvement: Detail

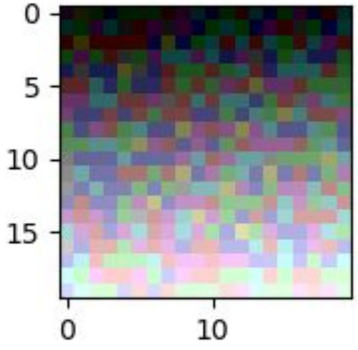
New features implemented:

- Targeted Attack
- Categorical Feedback Attack
- Alternative Grouping scheme

Random grouping



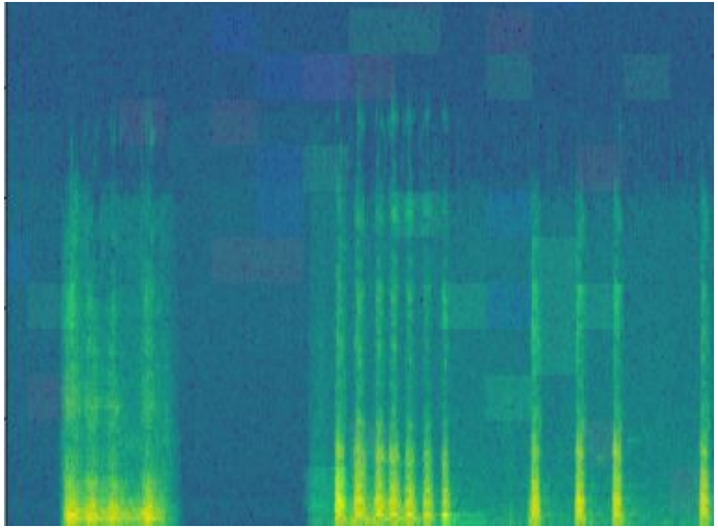
Original



Improvement

Expansion

- **Audio classifier - 5 classes**:
  human, cat, dog, parrot, kid
- Data harnessed by youtube_dl
- Data converted to spectrogram images
- Training by fine-tuned Resnet50 architecture, accuracy ~80%



Results

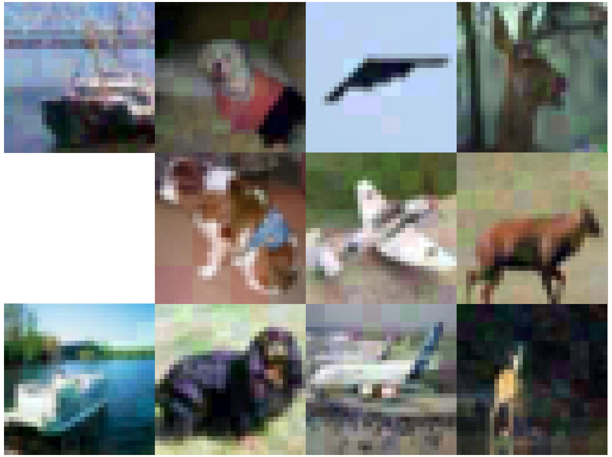
Replicated

Targeted

Randomly Grouped

Audio Domain

Results: Replicated



Cifar-10 Example

ImageNet Example

Results: Replicated

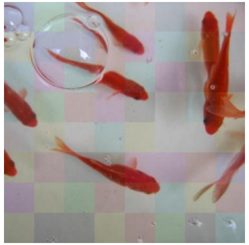
		Success Rate (%)		Average Query	
		Research (on 1000)	Ours (on 50)	Research (on 1000)	Ours (on 50)
ImageNet		99.3	98	561	666
Cifar-10	Undefended	100	100	247	531
	Defended	47.7	44	963	925

Results: Replicated

		Success Rate (%)		Average Query	
		Research (on 1000)	Ours (on 50)	Research (on 1000)	Ours (on 50)
ImageNet		99.3	98	561	666
Cifar-10	Undefended	100	100	247	531
	Defended	47.7	44	963	925

ImageNet Example

<goldfish>

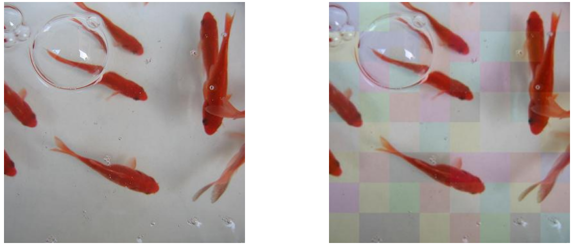


<screwdriver>

Results: Replicated

		Success Rate (%)		Average Query	
		Research (on 1000)	Ours (on 50)	Research (on 1000)	Ours (on 50)
ImageNet		99.3	98	561	666
Cifar-10	Undefended	100	100	247	531
	Defended	47.7	44	963	925

ImageNet Example



Cifar-10 Example

Results: Targeted

	Success Rate (%)	Average Query
ImageNet	54	5547
Cifar (Undefended)	100	931

ImageNet Example

<goldfish>

<street sign>

Cifar-10 Example

<airplane>

<ship> <horse> <frog>

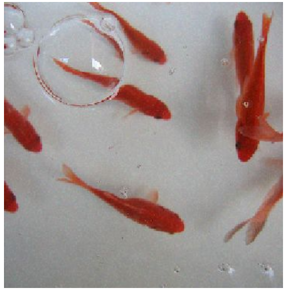
Results: Random Grouping

		Success Rate (%)		Average Query	
		Research (on 1000)	Ours (on 50)	Research (on 1000)	Ours (on 50)
ImageNet		99.3	98	561	$\boxed{666}$
Cifar-10	Undefended	100	100	247	$\boxed{531}$
	Defended	47.7	44	963	925

	Success Rate (%)	Average Query
ImageNet	80	$\boxed{1522}$
Cifar (Undefended)	96	$\boxed{581}$

ImageNet Example

<goldfish>



<lollipop>

Cifar-10 Example

<deer> → <dog>

<horse> → <dog>

Results: Categorical

CIFAR Undefended Non-Targeted

- Success Rate (on 50) = 6%
- Average Query = 5617

<cat> → <dog>

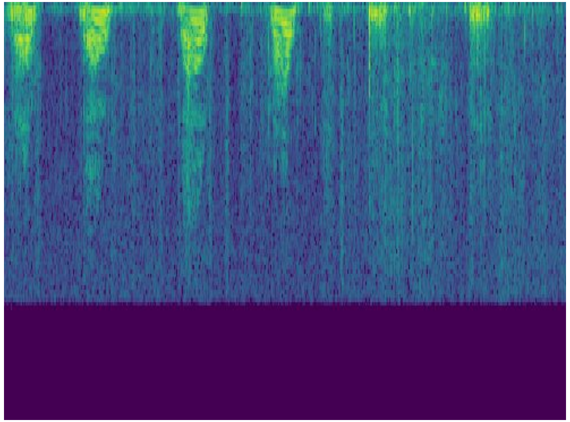
<deer> → <frog>

<deer> → <frog>

Results: Audio

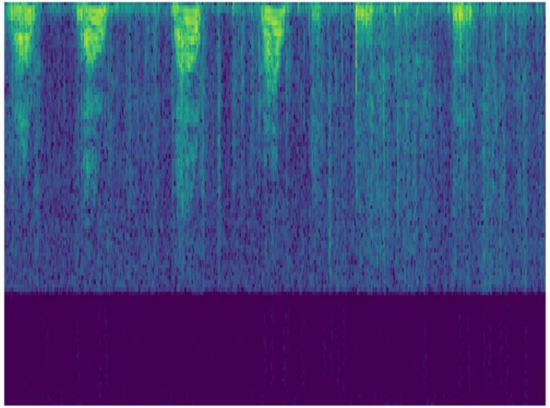
Audio Non-Targeted

- Success Rate (on 2) = 100%
- Average Query = 655.5



<cat>

after 967 queries



<human>

Conclusion

- DS is effective on targeted attacks & audio domain (spectrograms turned out to be robust to noise)
- Random grouping has less artifacts, but with worse quality
- Categorical Attack → not successful in query usage

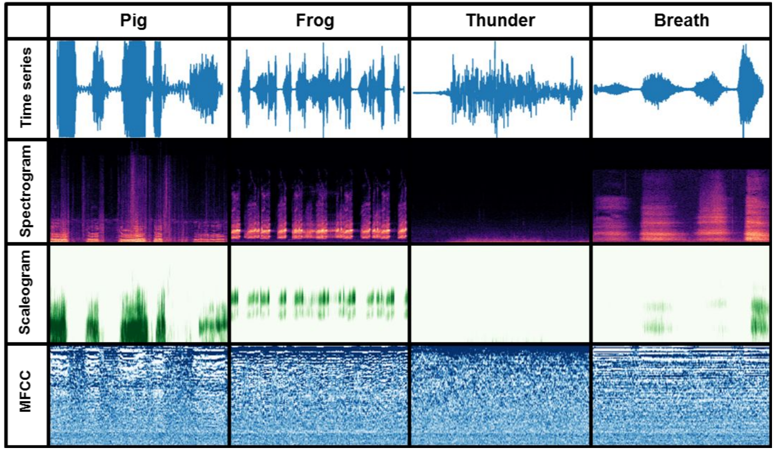
Replication		Success Rate (%)		Average Query	
		Research (on 1000)	Ours (on 50)	Research (on 1000)	Ours (on 50)
ImageNet		99.3	98	561	666
Cifar-10	Undefended	100	100	247	531
	Defended	47.7	44	963	925

Targeted Attack	Success Rate (%)	Average Query
ImageNet	54	5547
Cifar (Undefended)	100	931

Random Grouping	Success Rate (%)	Average Query
ImageNet	80	1522
Cifar (Undefended)	96	581

Future work

- Different representations for audio
- Representation-independent implementation:
  - Raw mutation (directly on sound)
  - Back-and-forth implementation (raw -> image -> mutate -> raw)

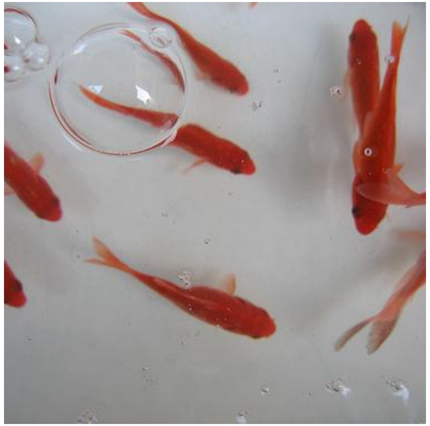



Thank you for your attention!

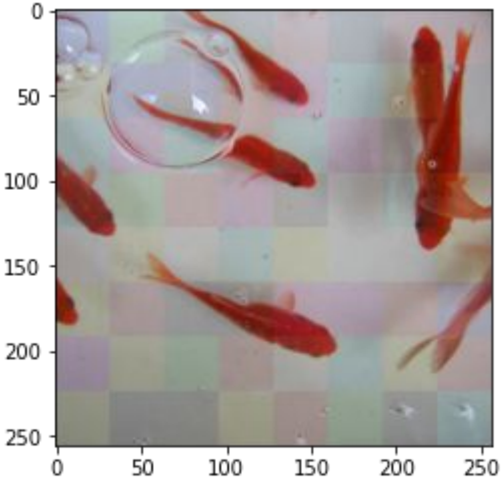
Results: Replicated

Non Targeted Imgnt

* Attack Succeeded with 261 queries



goldfish, Carassius auratus



screwdriver

Results: Replicated

Non Targeted CIFAR (defended)
* Attack Succeeded with 117 queries

<dog> → <cat>

Results: Targeted

Targeted Imgnt

* Attack Succeeded with 4654 queries

<goldfish, Carassius auratus>

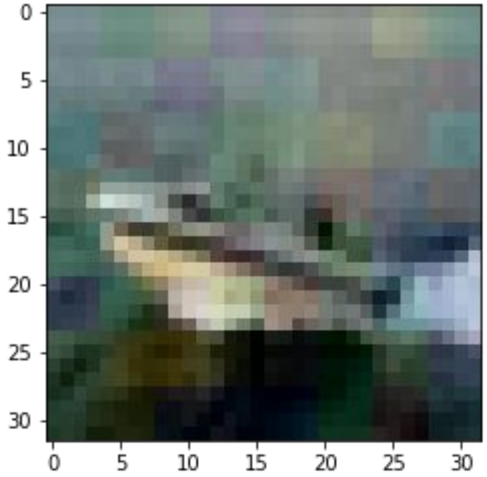
<street sign>

Results: Targeted

Targeted CIFAR (UD)

* Attack Succeeded with 202 queries

<ship>



<airplane>