ACM Multimedia 2020

A tutorial on Deep Learning for Privacy in Multimedia

Part 2: Adversarial images

Ali Shahin Shamsabadi

a.shahinshamsabadi@qmul.ac.uk

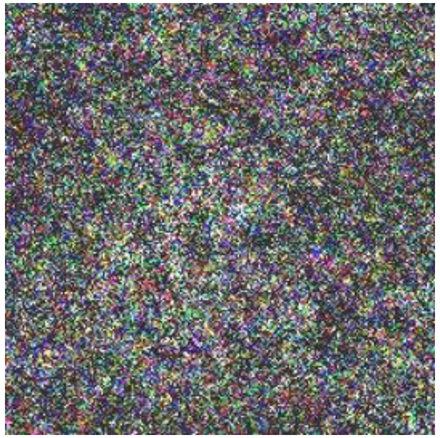
CIS centre for intelligent sensing

Queen Mary University of London

What is an adversarial image?

Image

Perturbation



Adversarial Image

DNN classifier → Suit

DNN classifier → Kimono

Biggio et al, “Evasion attacks against machine learning at test time”, 2013.
Szegedy et al, “Intriguing properties of neural networks,” ICLR’14.

An adversarial image should be

- Unnoticeable
  - Humans do not perceive any visual distortions and keep aspect ratio

Original image

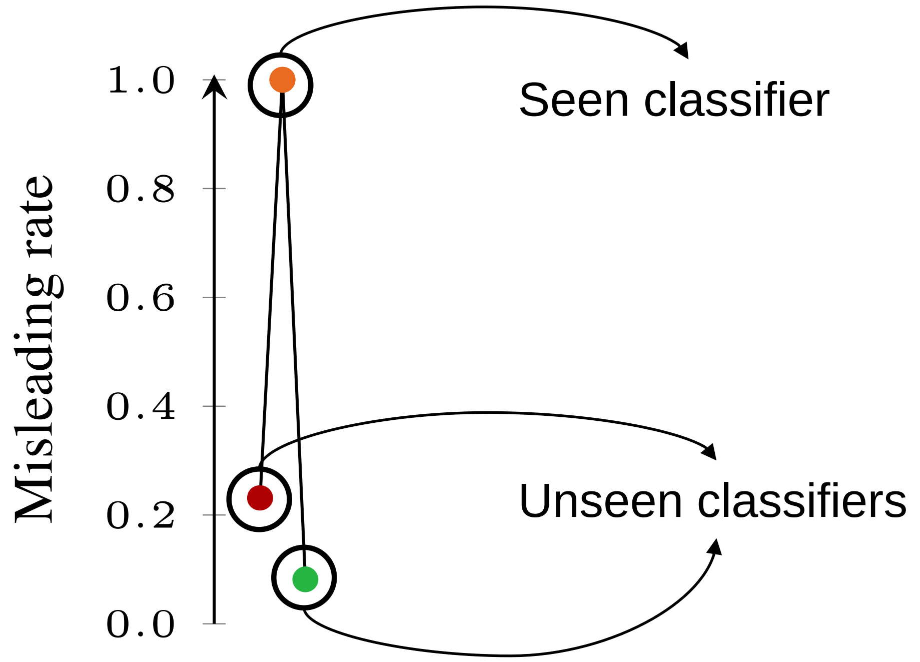
Adversarial Image[CVPR-W'17]

Adversarial Image[CVPR'20]

CVPR-W'17: Hosseini and Poovendran, "Semantic adversarial examples".
CVPR'20: Shahin Shamsabadi et al, "ColorFool: Semantic adversarial colorization".

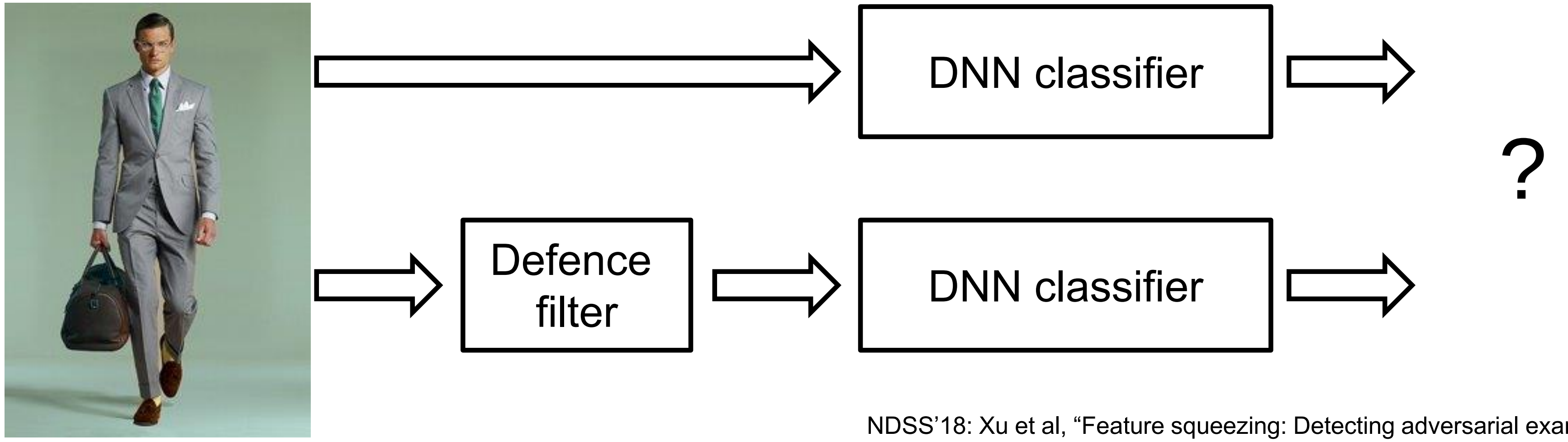
An adversarial image should be

- Unnoticeable
  - Humans do not perceive any visual distortions and keep aspect ratio
- Transferable
  - The adversarial image can mislead unseen classifiers
    (i.e. do not overfit to one classifier)



An adversarial image should be

- Unnoticeable
  - Humans do not perceive any visual distortions and keep aspect ratio
- Transferable
  - The adversarial image can mislead unseen classifiers
    (i.e. do not overfit to one classifier)
- Robust
  - The perturbation is not removed by defence frameworks[NDSS’18]
  - Quantization, Median smoothing and JPEG compression



NDSS’18: Xu et al, “Feature squeezing: Detecting adversarial examples in deep neural networks”.

An adversarial image should be

- Unnoticeable
  - Humans do not perceive any visual distortions and keep aspect ratio
- Transferable
  - The adversarial image can mislead unseen classifiers (i.e. do not overfit to one classifier)
- Robust
  - The perturbation is not removed by defense frameworks[NDSS'18]
  - Quantization, Median smoothing and JPEG compression

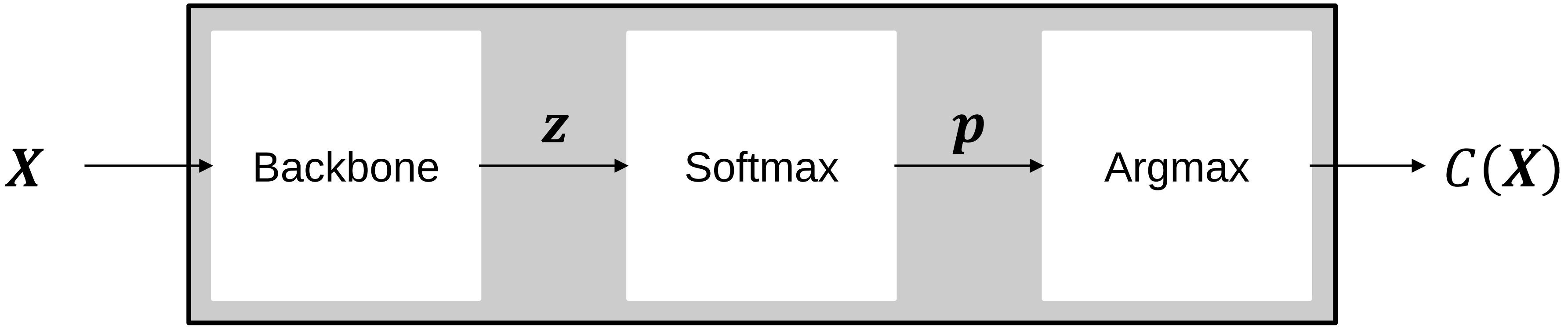
Knowledge of adversary about the classifier

	Knowledge	Example	
White-box	Everything	Open source classifiers	Best-case scenario for the attacker
Black-box	Predictions	Public APIs	-
No box	Nothing	Classifiers on social media	Worst-case scenario for the attacker

Adversarial goals

- Untargeted
  - Mislead with any class that is different from the class of the original image
- Targeted
  - Mislead to predict a specific target class

Training a classifier, $C(\cdot)$



Loss function

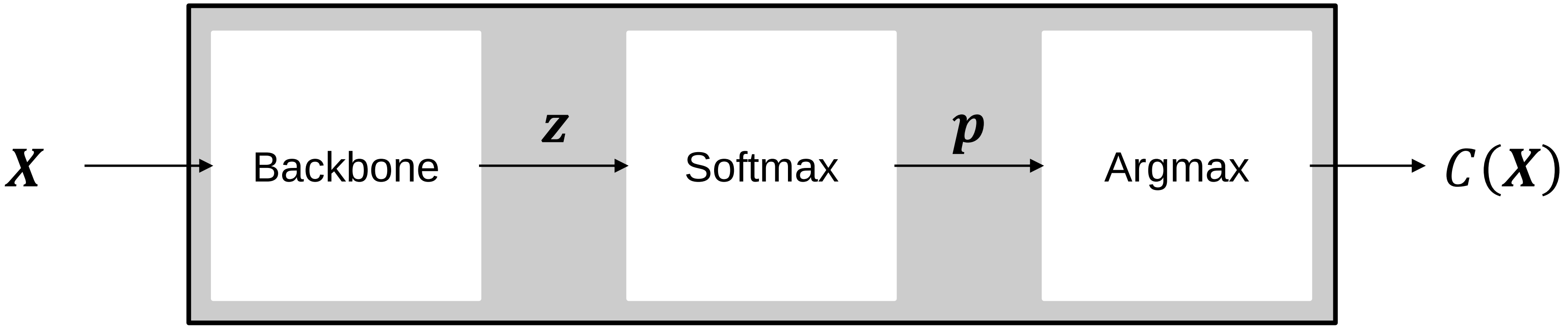
$$\theta^* = \underset{\theta}{\operatorname{argmin}}\, J(C(\boldsymbol{X}), y)$$

Learned parameters

Ground-truth label of $\boldsymbol{X}$

Input is fixed,
Adjust parameters

Generating an adversarial image against $C(\cdot)$

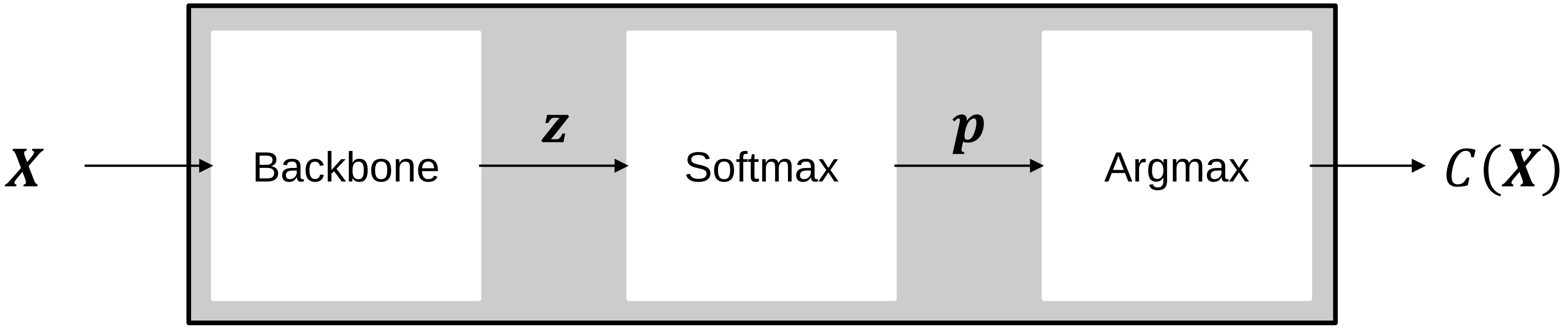


Fixed parameters,
Adjust input

$$\dot{\boldsymbol{X}} = \underset{\dot{\boldsymbol{X}}}{\operatorname{argmax}}\, J(C(\dot{\boldsymbol{X}}), y)$$

Adversarial image

Generating an adversarial image against $C(\cdot)$



$$\dot{\boldsymbol{X}} = \underset{\dot{\boldsymbol{X}}}{\operatorname{argmax}}\, J\left(C\left(\dot{\boldsymbol{X}}\right), y\right) \text{ s.t. } \begin{cases} \dot{\mathbf{X}} \text{ is still an image} \\ \text{Perceptually similar} \end{cases}$$

Generating an adversarial image against $C(\cdot)$

- Parameters are fixed, maximising the gradients of the loss function wrt input

$$\boldsymbol{X} \longrightarrow \text{Backbone} \xrightarrow{\boldsymbol{z}} \text{Softmax} \xrightarrow{\boldsymbol{p}} \text{Argmax} \longrightarrow C(\boldsymbol{X})$$

Difficult and
Not mathematically possible

$$\dot{\boldsymbol{X}} = \underset{\dot{\boldsymbol{X}}}{\operatorname{argmax}}\, J(C(\dot{\boldsymbol{X}}), y) \text{ s.t. } \begin{cases} \dot{\mathbf{X}} \text{ is still an image} \\ \text{Perceptually similar} \end{cases}$$

So how we perturb images?

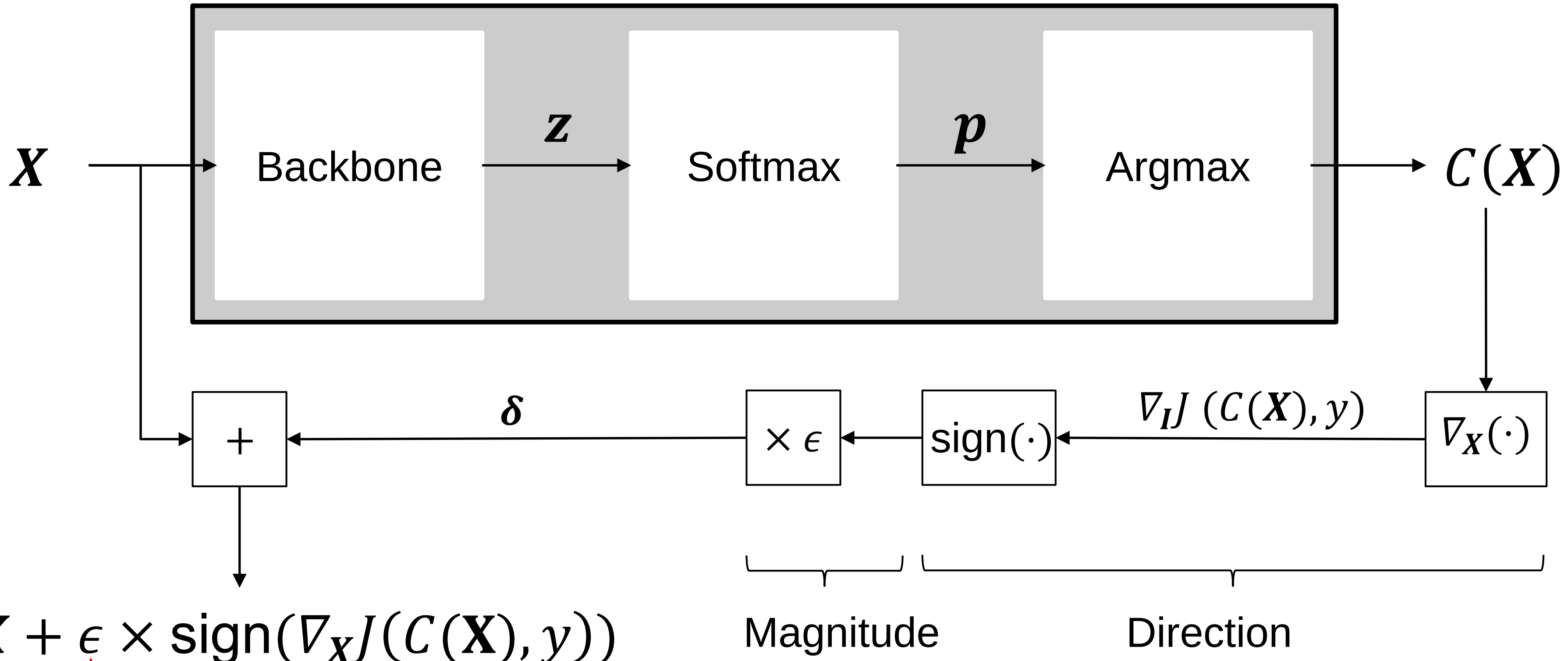
- Possible space of allowed perturbations
  - Norm-bounded perturbations
  - Content-based perturbations

Norm-bounded perturbations

- Minimize l_p-norm
  - Maximum change for each pixel, l_∞
    - Fast Gradient Sign Method (FGSM)[ICLR'14]
    - Basic Iterative Method (BIM)[ICLR'17]
    - Robust and private BIM (RP-BIM)[TMM'20]
  - Maximum energy change, l_2
    - DeepFool (l_2)[CVPR'16]
    - Carlini-Wagner (CW)[S&P'17]
  - Maximum number of perturbed pixels, l_1
    - SparseFool[CVPR'19]
    - JSMA[EuroS&P'16]

ICLR'14: Goodfellow et al, "Explaining and harnessing adversarial examples".
ICLR'17: Kurakin et al, "Adversarial examples in the physical world".
TMM'20: Sanchez-Matilla et al, "Exploiting vulnerabilities of deep neural networks for privacy protection".
CVPR'16: Moosavi-Dezfool et al, "DeepFool: A simple and accurate method to fool deep neural networks ".
S&P'17: Carlini and Wagner, "Towards evaluating the robustness of neural networks".
CVPR'19: Modas et al, "SparseFool: a few pixels make a big difference".
EuroS&P'16: Papernot et al. "The limitations of deep learning in adversarial settings".

Fast Gradient Sign Method, untargeted

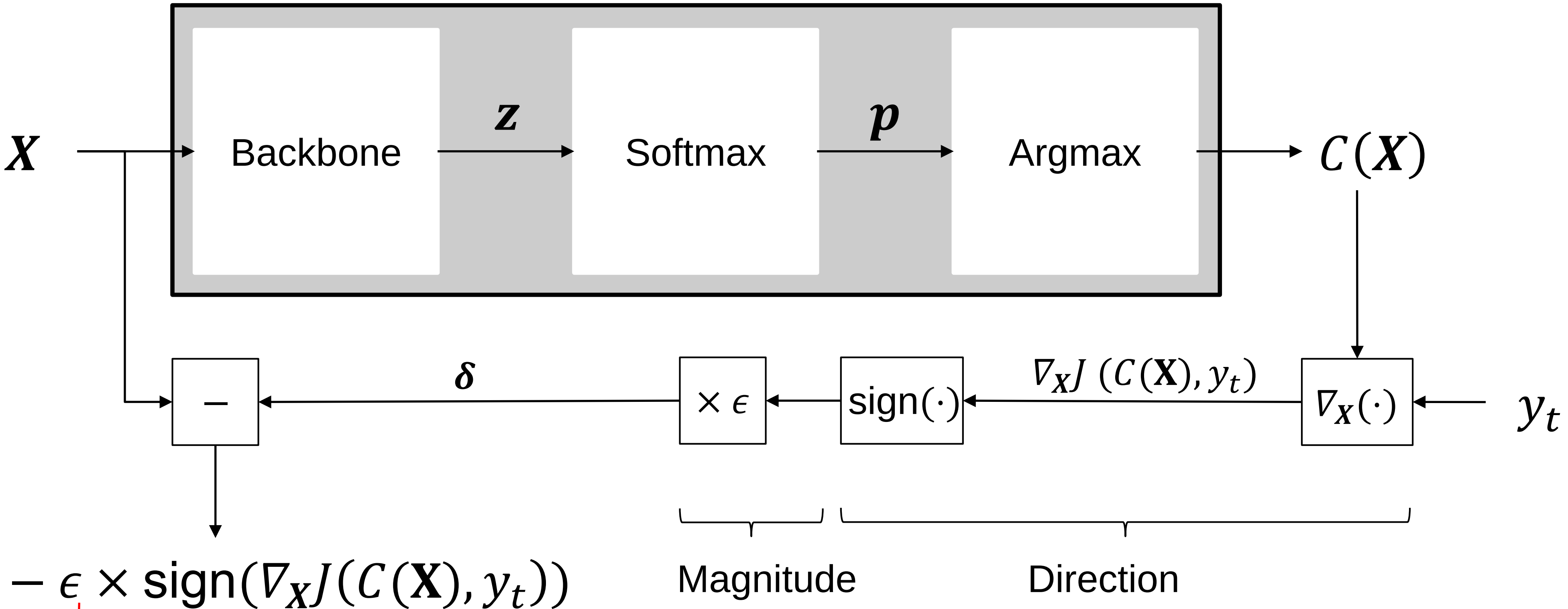


$$\dot{\boldsymbol{X}} = \boldsymbol{X} + \epsilon \times \mathrm{sign}(\nabla_X J(C(\mathbf{X}), y))$$

Increase or decrease least significant bit of each pixel intensity

ICLR'14: Goodfellow et al, "Explaining and harnessing adversarial examples".

Fast Gradient Sign Method, targeted



$$\dot{X} = X - \epsilon \times \mathrm{sign}(\nabla_X J(C(\mathbf{X}), y_t))$$

Increase or decrease least significant bit of each pixel intensity

ICLR'14: Goodfellow et al, "Explaining and harnessing adversarial examples".

Improving misleading of Fast Gradient Sign Method

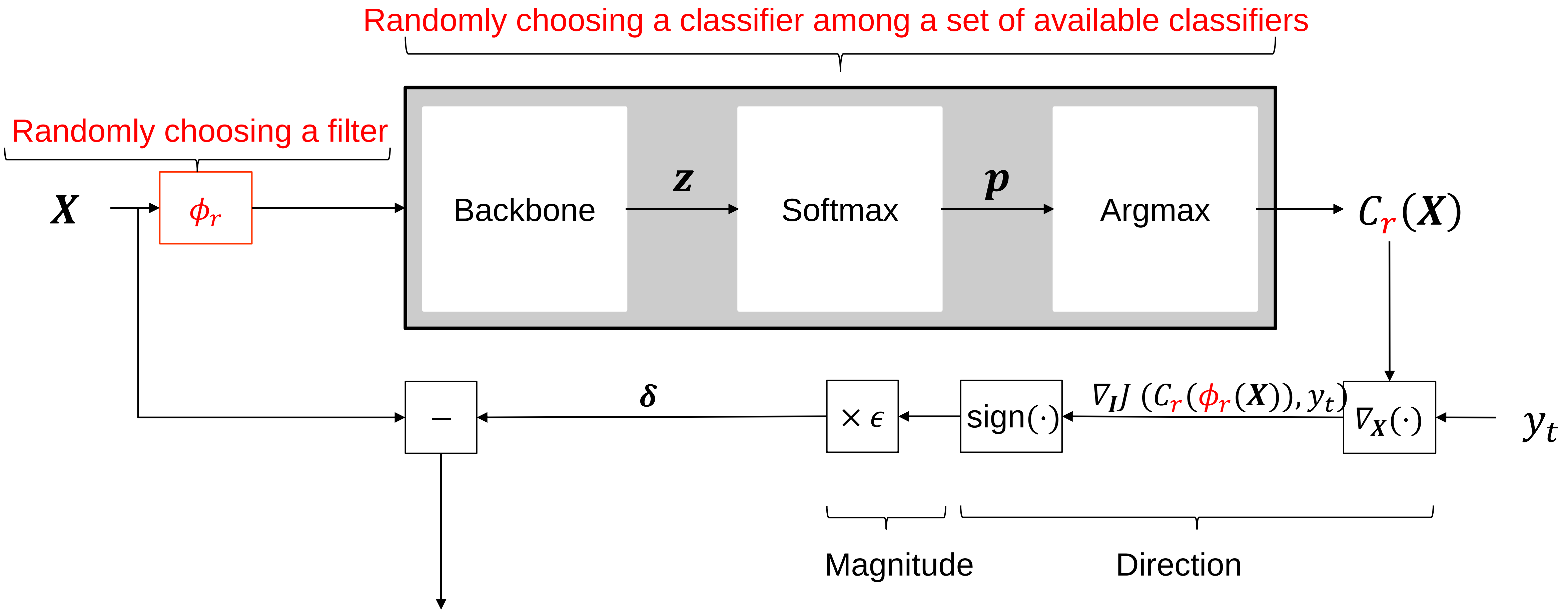
- Basic Iterative Method: Iteratively generate the adversarial perturbation
- Aggregates N perturbations

$$\dot{\boldsymbol{X}}_0 = \boldsymbol{X}$$

$$\dot{\boldsymbol{X}} = \mathrm{Clip}_{\epsilon}(\boldsymbol{X} + \sum_{n} \text{sign}\left(\nabla_{\dot{\boldsymbol{X}}_{n-1}} J\left(C\left(\dot{\boldsymbol{X}}_{n-1}\right), y\right)\right))$$

ICLR'17: Kurakin et al, "Adversarial examples in the physical world".

Improving robustness and transferability of FGSM/BIM



$$\dot{\boldsymbol{X}} = \boldsymbol{X} - \epsilon \times \text{sign}(\nabla_X J(C_r(\phi_r(\boldsymbol{X})), y_t))$$

TMM'20: Sanchez-Matilla et al, "Exploiting vulnerabilities of deep neural networks for privacy protection".

In summary, norm-bounded perturbations are

- Unnoticeable to human eyes
  - Small perturbations

- Not transferable to unseen classifiers
  - Overfitted to a specific classifier
  - Small magnitudes

- Not robust to defenses
  - High-frequency

In summary, norm-bounded perturbations are

- Unnoticeable to human eyes
  - Small perturbations

- Not transferable to unseen classifiers
  - Overfitted to a specific classifier
  - Small magnitudes

- Not robust to defenses
  - High-frequency

Neglect image content

How image contents can help adversarial perturbations?

- Introduce a larger range of perturbations based on image content
  - Colour
    - SemanticAdv[CVPR-W'17]
    - ColorFool[CVPR'20]
    - ACE[BMVC'20]

- Structure and details
    - EdgeFool[ICASSP'20]
    - FilterFool[Arxiv'20]

CVPR-W'17: Hosseini and Poovendran, "Semantic adversarial examples".
CVPR'20: Shahin Shamsabadi et al, "ColorFool: Semantic adversarial colorization".
BMVC'20: Zhao et al, "Adversarial Color Enhancement: Generating Unrestricted Adversarial Images by Optimizing a Color Filter".
ICASSP'20: Shahin Shamsabadi et al, "EdgeFool: An adversarial image enhancement filter ".
Arxiv'20: Shahin Shamsabadi et al, "Semantically Adversarial Learnable Filters ".

Queen Mary
University of London

Large colour perturbations

- Untargeted adversarial colour changes
  - HSV colour space
  - Shifting hue and saturation
- Low-frequency colour perturbations
  - Transferable
  - Robust
  - Black-box
  - Unnatural-looking adversarial images

Original

SemanticAdv

CVPR-W'17: Hosseini and Poovendran, "Semantic adversarial examples".

Why adversarial colours may look unnatural?

- Images contain semantic regions
  - A set of identifiable pixels such as car or person

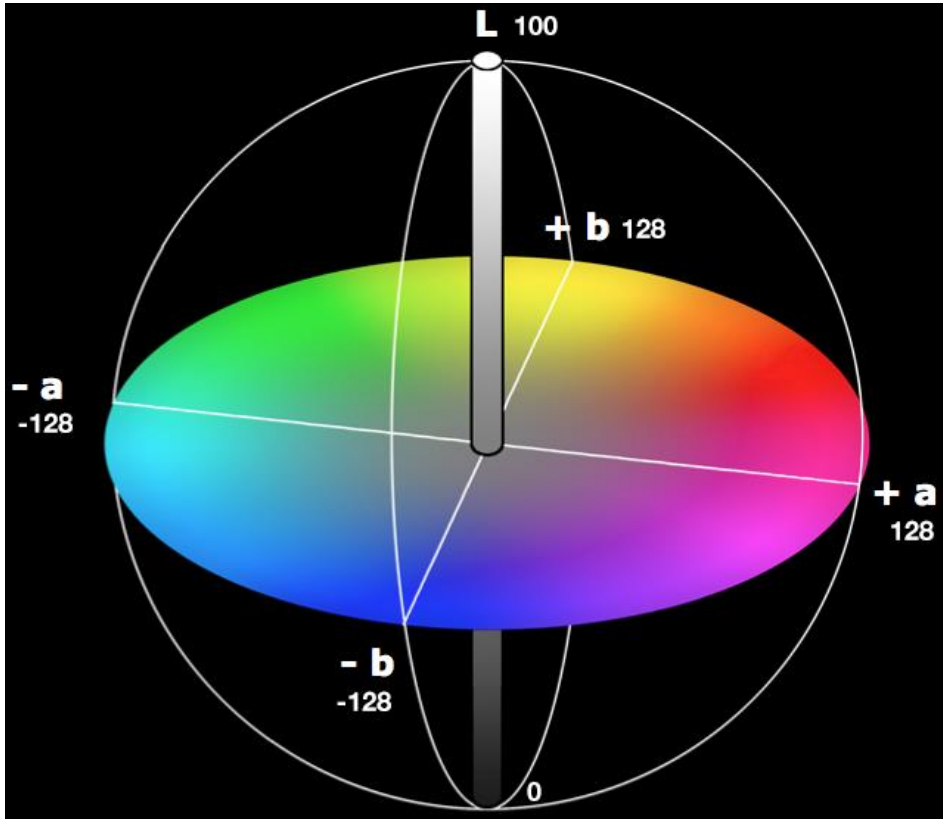
- Sensitive regions
  - Colour changes are noticeable
  - Person, Vegetation, Water and Sky

- Non-sensitive regions
  - Occur in a wide range of colours
  - Curtain, Wall and …

CVPR'20: Shahin Shamsabadi et al, “ColorFool: Semantic adversarial colorization”.

We also need a better perceptually motivated colour space

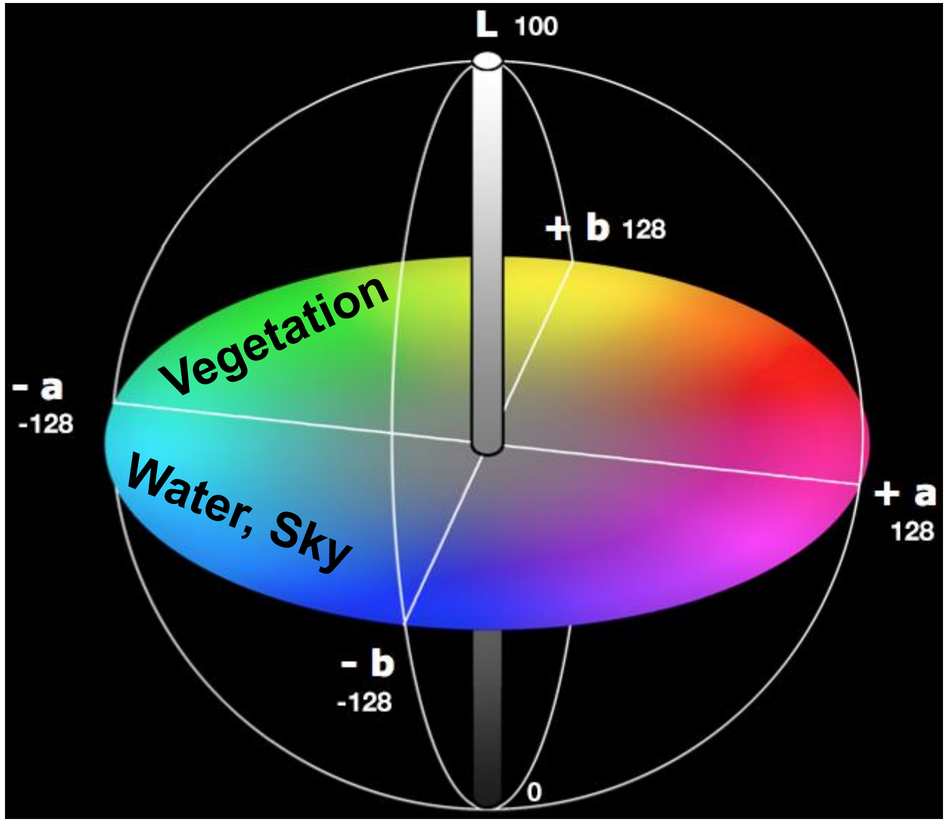
- Lab colour space
- Perceptually uniform
- Represent colour separately from lightness
  - Lightness
    - L channel: black (0) to white (100)
  - Colour information
    - a channel: green (-128) to red (+128)
    - b channel: blue (-128) to yellow (+128)



co_cop_QuantifyingColors.html

Semantic and perceptually motivated colour perturbations

- Perform in Lab colour space
- Modify de-correlated a and b colour channels
  - Semantic regions
  - Human perception
- Up to 1000 trials



co_cop_QuantifyingColors.html

Semantic and perceptually motivated colour perturbations

- Perform in Lab colour space
- Modify a and b colour channels
  - Semantic regions
  - Human perception
- Up to 1000 trials

Original

ColorFool-ed

CVPR'20: Shahin Shamsabadi et al, "ColorFool: Semantic adversarial colorization".

ColorFool

- Transferable
  - Randomness
  - Black-box
- Robust
  - Low frequency
- Natural-looking
  - Semantic regions
  - Human perception
- Maintain aspect ratio

Original

ColorFool-ed

CVPR’20: Shahin Shamsabadi et al, “ColorFool: Semantic adversarial colorization”.

Now let's see how we can implement ColorFool-ed images

Original

ColorFool-ed

https://github.com/smartcameras/ColorFool

CVPR'20: Shahin Shamsabadi et al, "ColorFool: Semantic adversarial colorization".

What about other semantic attributes of images?

=

Image structure

+

Image details

ICASSP’20: Shahin Shamsabadi et al, “EdgeFool: An adversarial image enhancement filter ”.

Queen Mary University of London

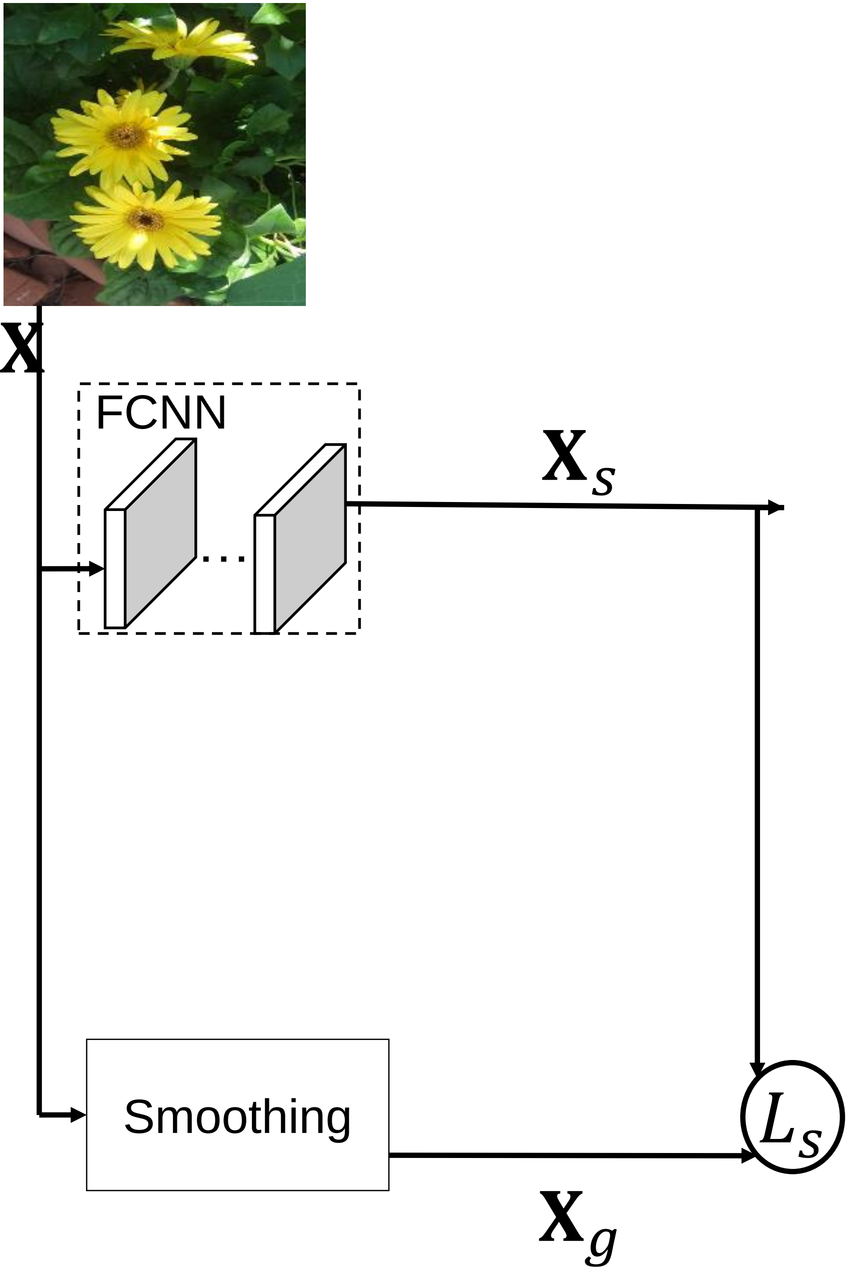
Can perturbations perform enhancement, not distortion?

- Yes
- Exploiting traditional image processing filters

Original

EdgeFool-ed

ICASSP'20: Shahin Shamsabadi et al, "EdgeFool: An adversarial image enhancement filter ".



Decompose structure and detail by training a Fully Convolutional Neural Network (FCNN)

Structure ($\mathbf{I}_s$)

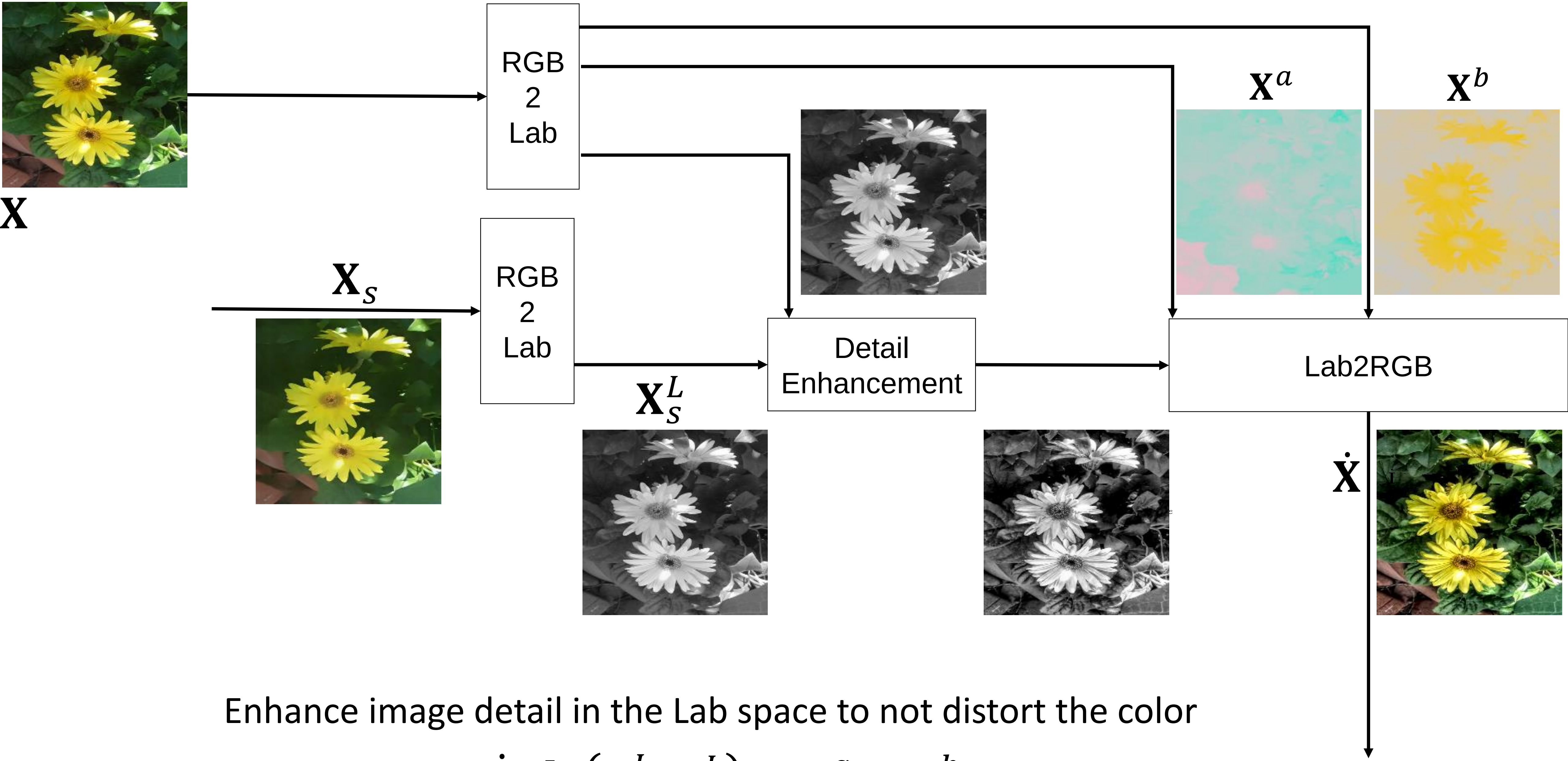
Detail

$$L_s = ||\mathbf{X}_s - \boldsymbol{X}_g||^2$$

Smoothing Loss Function

Image Structure learned by FCNN

Image Structure output by l_0 Image smoothing filter



Enhance image detail in the Lab space to not distort the color

$$\dot{\mathbf{X}}=[f\left(\mathbf{X}_d^L,\mathbf{X}_s^L\right),\quad \mathbf{X}^a,\quad \mathbf{X}^b]$$

$\mathbf{X}$

Guide the enhancement in an adversary manner via minimizing

$$L_{\mathrm{a}} = \dot{z}_y - \max\{\dot{z}_i, i \neq y\}$$

Adversarial
Loss Function

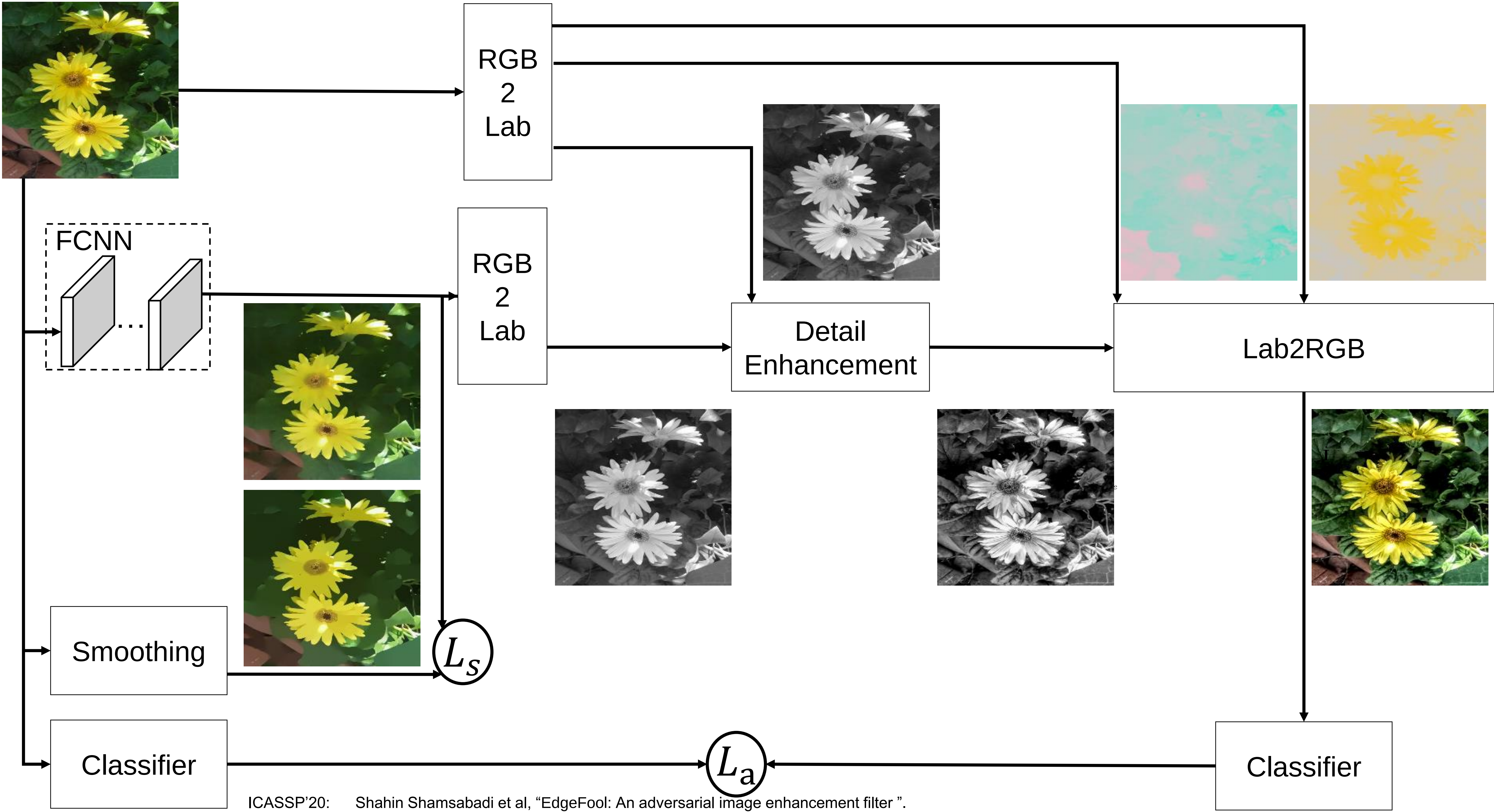
Score of belonging adversarial image
to the same class as the original image

$\dot{\mathbf{X}}$

Classifier

L_{a}

Classifier



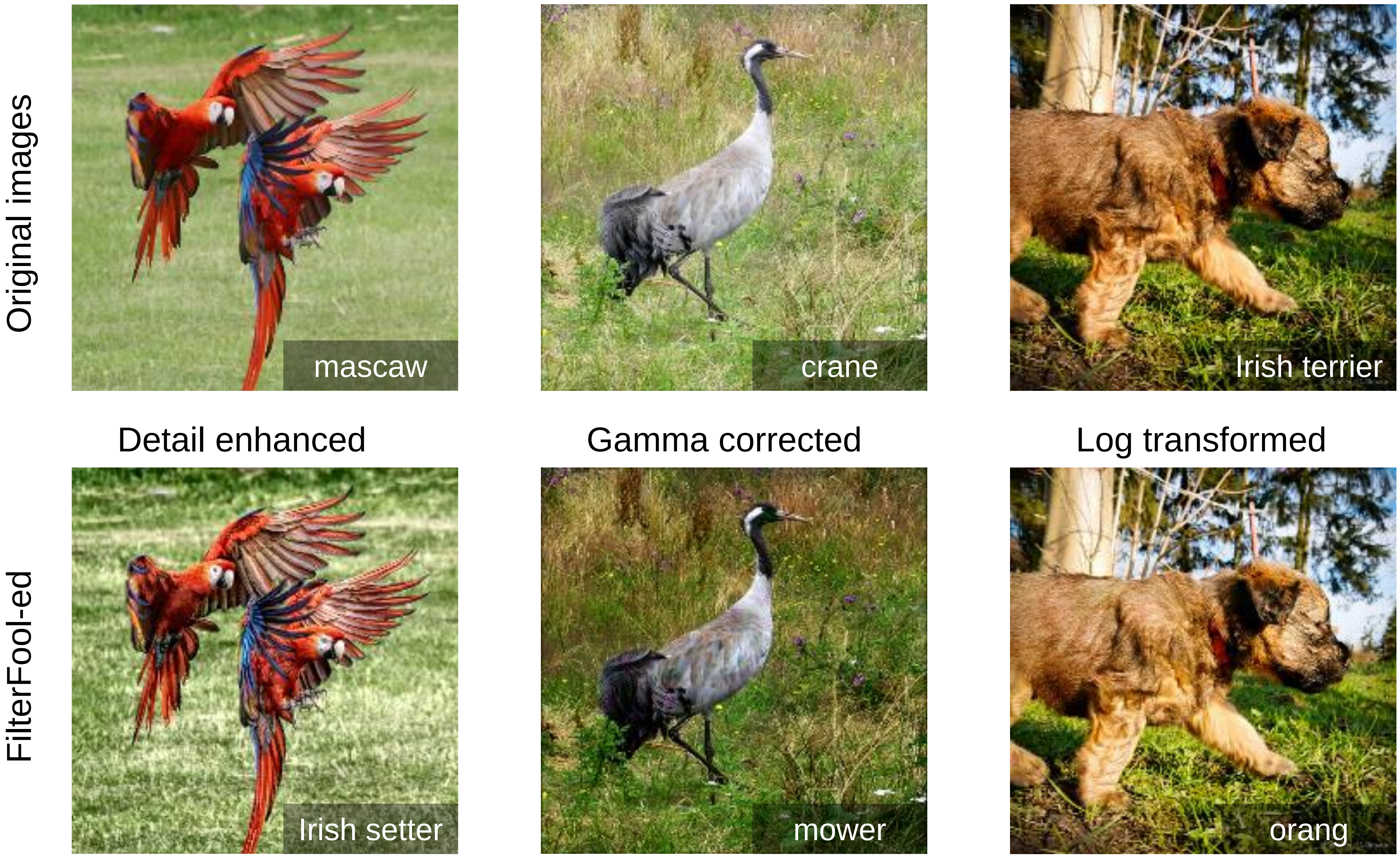
ICASSP'20: Shahin Shamsabadi et al, "EdgeFool: An adversarial image enhancement filter".

Residual learning: if you look after other enhancements

- Enhanced image outputted by the filter: $\boldsymbol{X}_e$
- The filter residual: $\boldsymbol{\delta}_e = \boldsymbol{X}_e - \boldsymbol{X}$
- Learnable adversarial residual: $\boldsymbol{\delta}$

$$\underbrace{L_{l_2}(\boldsymbol{\delta}, \boldsymbol{\delta}_e)}_{\text{Penalizes pixel-wise large differences}} + \overbrace{(1 - \mathrm{SSIM}(\boldsymbol{\delta}, \boldsymbol{\delta}_e))}^{\text{Prevent artefacts in untextured areas}} + \underbrace{L_a}_{\text{Guides the residual towards adversarial}}$$

Arxiv'20: Shahin Shamsabadi et al, "Semantically Adversarial Learnable Filters ".



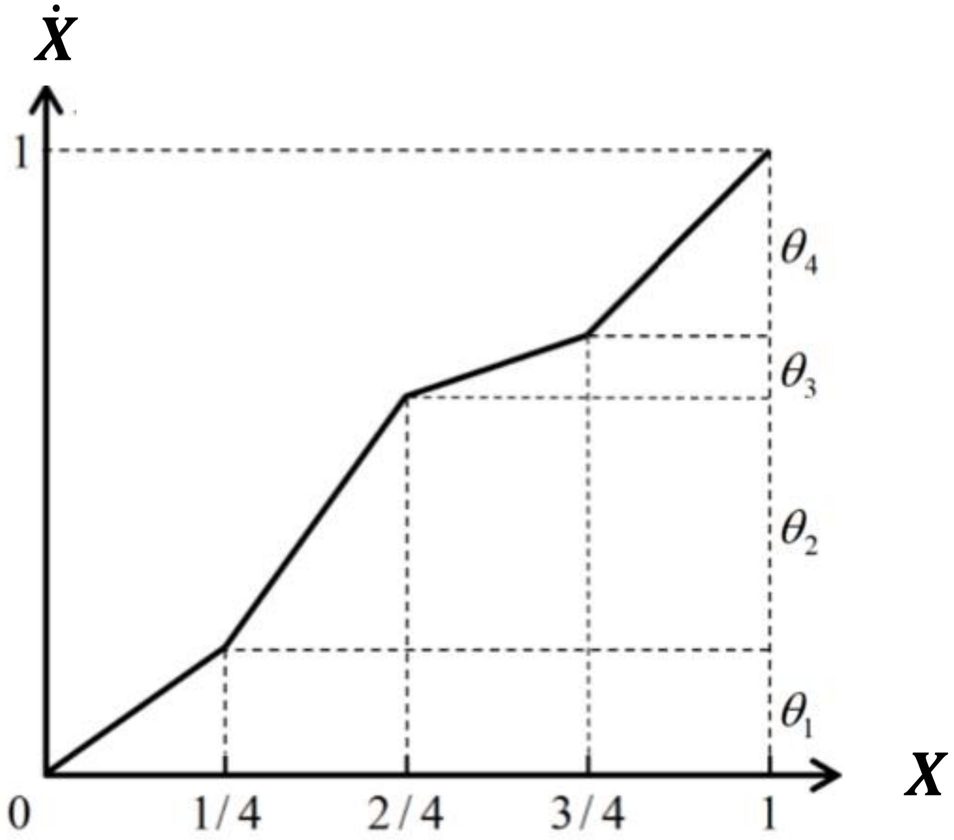
Arxiv'20: Shahin Shamsabadi et al, "Semantically Adversarial Learnable Filters ".

Others traditional image processing filters

- Adversary learn parameters of filters to output adversarial images

- Differentiable approximation of a colour filter

$$\dot{X}_x = \sum_{i=1}^{\left[\frac{X_x}{s}\right]} \theta_i + \frac{X_x (\mathrm{mod} s)}{s} \theta_{\left[\frac{X_x}{s}\right]}$$



BMVC'20: Zhao et al, "Adversarial Color Enhancement: Generating Unrestricted Adversarial Images by Optimizing a Color Filter".

Now let’s look at class labels

“screen”
?
“Television”

“computer keyboard”
?
“Typewriter keyboard”

“Shetland sheepdog”
?
“collie”

Arxiv’20: Shahin Shamsabadi et al, “Semantically Adversarial Learnable Filters ”.

Now let's look at class labels

Original:	"screen"	"computer keyboard"	"Shetland sheepdog"
Adversarial:	"Television"	"Typewriter keyboard"	"collie"

Arxiv'20: Shahin Shamsabadi et al, "Semantically Adversarial Learnable Filters ".

Semantic relationships between class labels

- Cluster $D = 1000$ ImageNet classes to $S = 11$ semantic classes
  - Dogs (130); Other mammals (88); Birds (59); Reptiles, fish, amphibians (60); Invertebrates (61); Food, plants, fungi (63); Devices (172); Structures, furnishing (90); Clothes, covering (92); Implements, containers, misc. objects (117); and Vehicles (68)

- Define mapping matrix $\boldsymbol{W} \in \{0,1\}^{D \times S}$

Semantically different classes than the original class

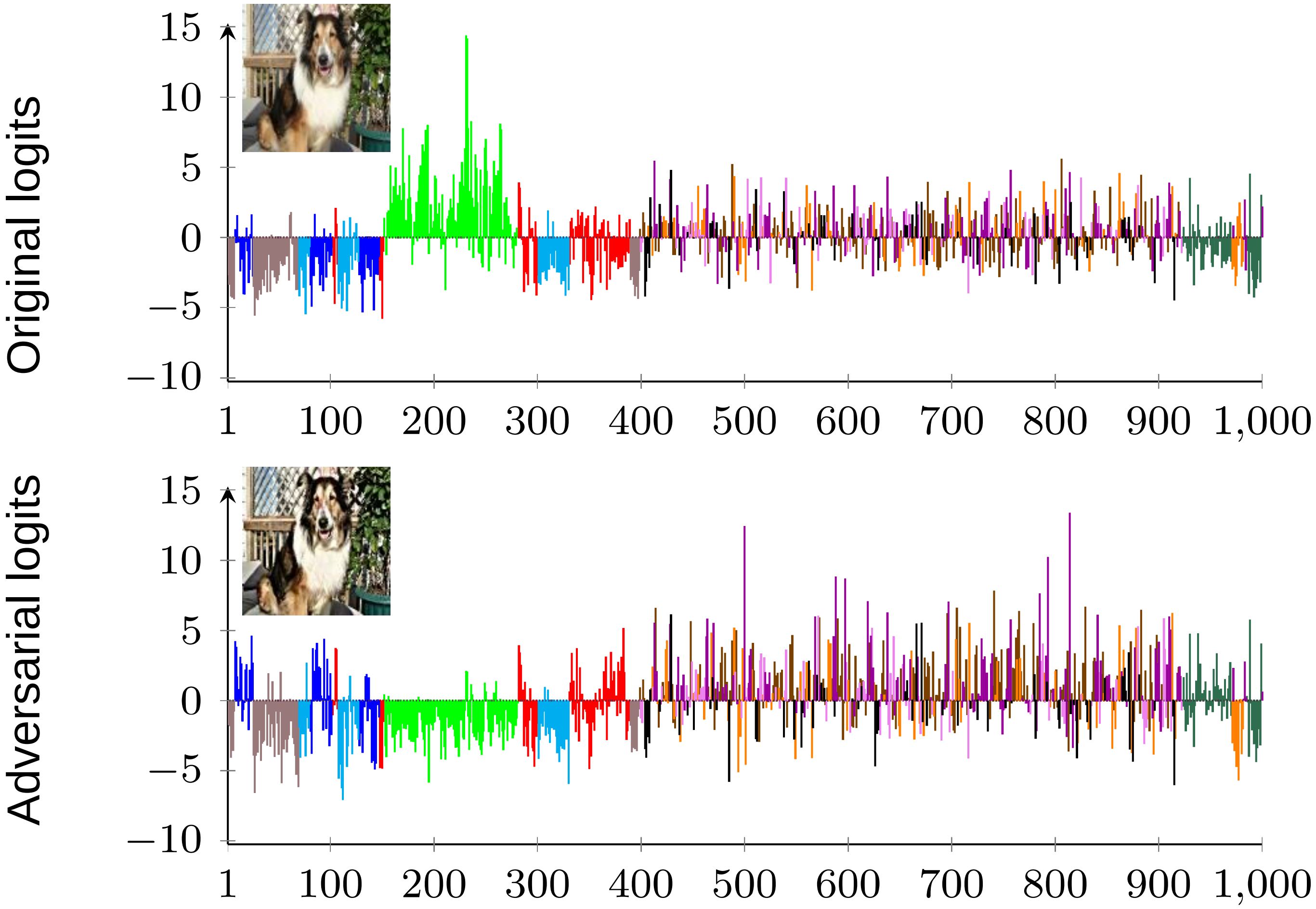
$$L_{\mathrm{S-Adv}}(\boldsymbol{X}, \dot{\boldsymbol{X}}) = \langle \mathrm{ReLU}(\dot{\boldsymbol{z}}), \boldsymbol{w}_s \rangle - \max(\dot{\boldsymbol{z}} \odot \hat{\boldsymbol{w}}_s)$$

Semantically similar classes to the original class

Arxiv'20: Shahin Shamsabadi et al, "Semantically Adversarial Learnable Filters ".

$<\cdot,\cdot>$: dot product $\odot$: Hadamard product

Semantic relationships between class labels



Dogs (130);
Other mammals (88);
Birds (59);
Reptiles, fish, amphibians (60);
Invertebrates (61);
Food, plants, fungi (63);
Devices (172);
Structures, furnishing (90);
Clothes, covering (92);
Implements, containers, misc. objects (117);
Vehicles (68).

$$L_{\mathrm{S-Adv}}(\boldsymbol{I}, \dot{\boldsymbol{I}}) = < \mathrm{ReLU}(\dot{\boldsymbol{z}}), \boldsymbol{w}_s > - \max\{\dot{\boldsymbol{z}} \odot \hat{\boldsymbol{w}}_s\}$$

Arxiv'20: Shahin Shamsabadi et al, "Semantically Adversarial Learnable Filters ".

Experimental settings

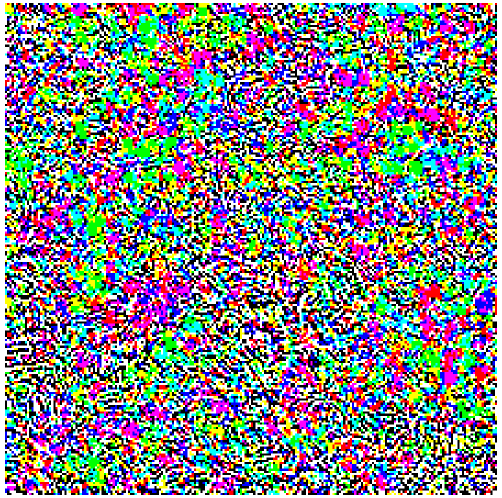
- Dataset
  - ImageNet: $3K$ images of $1k$ objects
- Classifiers under attacks
  - ResNet50, ResNet18, AlexNet
- Visualization
  - Perturbations
  - Adversarial images
  - Top5 predictions and classifier attention
- Success rate and transferability

$$\frac{\#\ \text{successful adversarial images}}{\#\ \text{total images}}$$

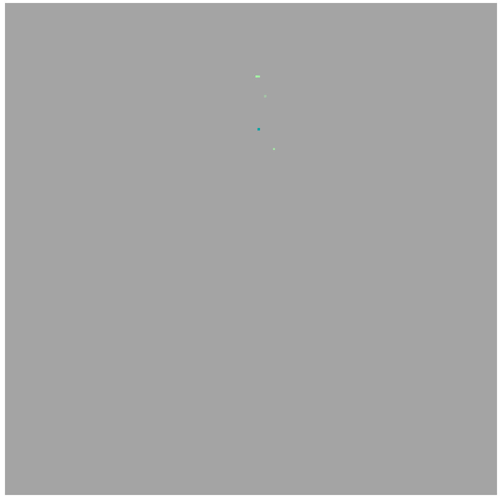
Adversarial perturbations

Original

Basic Iterative Method



SparseFool



SemanticAdv

ColorFool

EdgeFool

FilterFool

Adversarial images

Original

Basic Iterative Method

SparseFool

SemanticAdv

ColorFool

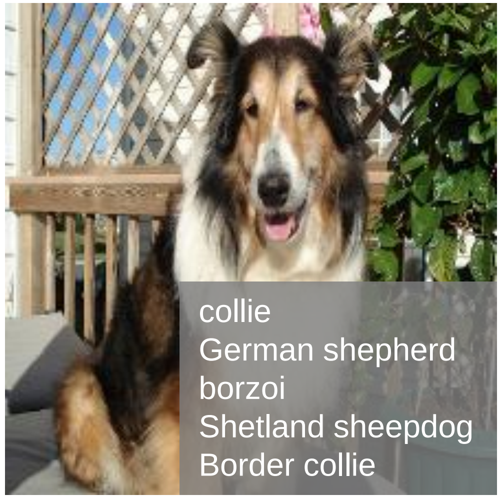
EdgeFool

FilterFool

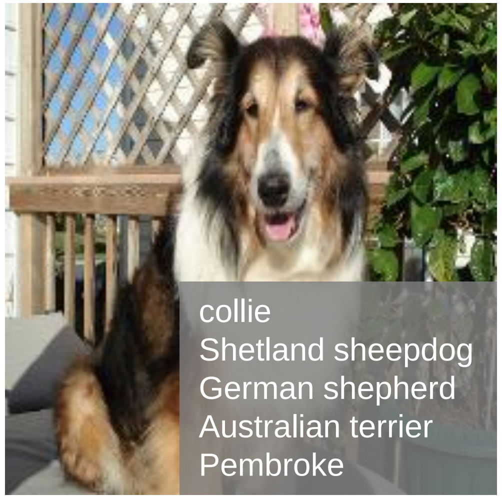
Top-5 predictions

Original

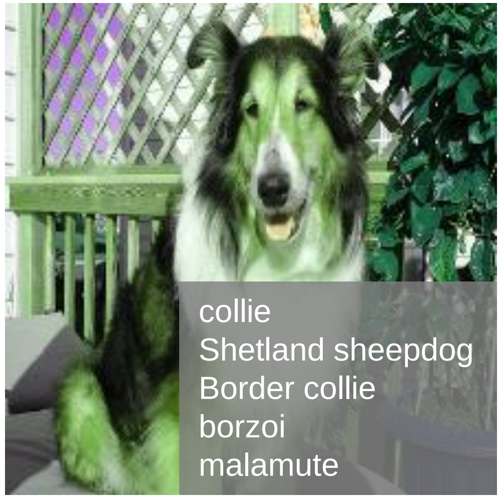
Basic Iterative Method



SparseFool



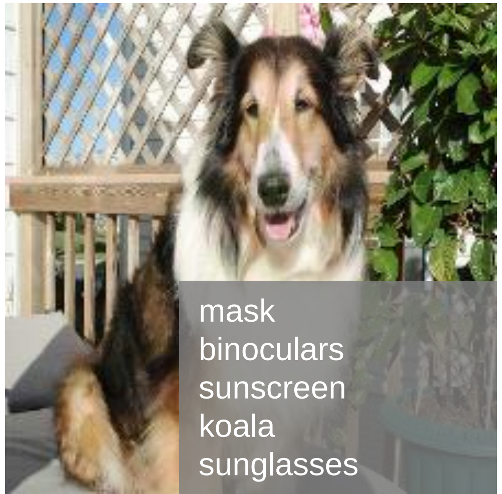
SemanticAdv



ColorFool

EdgeFool

FilterFool



Classifier attention

Original

Basic Iterative Method

SparseFool

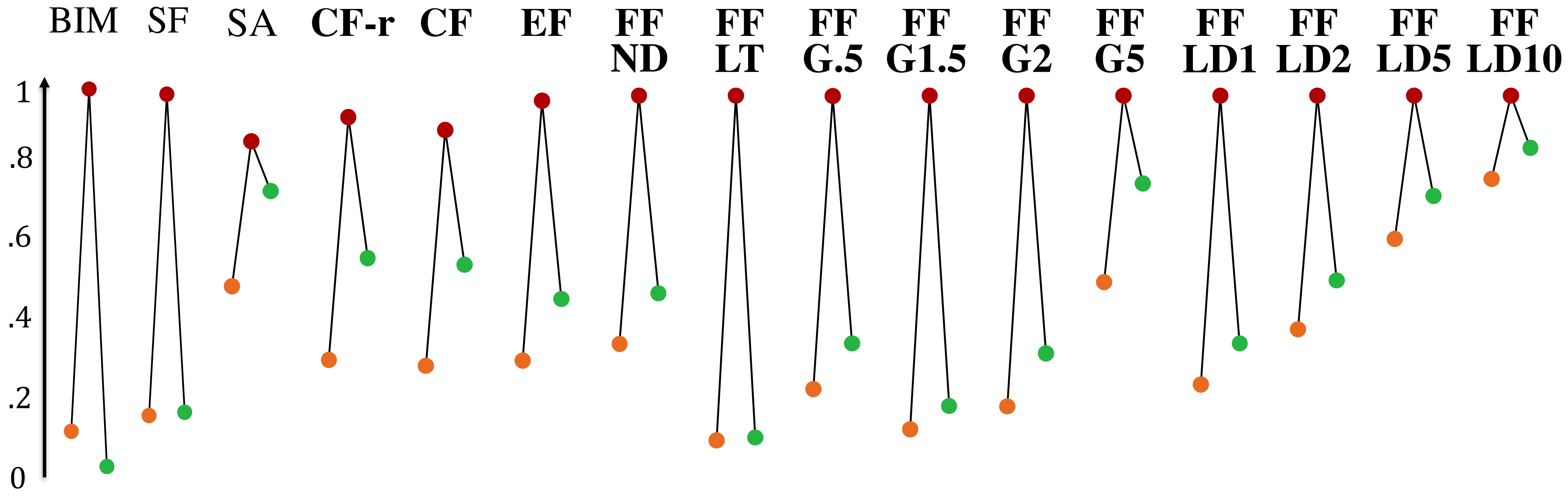
SemanticAdv

ColorFool

EdgeFool

FilterFool

Success rate and transferability



ResNet50 (seen)
ResNet18 (unseen)
AlexNet (unseen)

BIM: Basic Iterative Method, SF: SparseFool, SA: SemanticAdv,
EF: EdgeFool, FF: FilterFool, LT: Log transformation,
LD, ND: Linear and Nonlinear Detail enhancement,
G: Gamma correction

Conclusion

- Images are special data not like tabular data

- Improves unnoticeability, transferability and robustness
  - Large content-based perturbations
  - Structure, detail, colours and objects

- Protect privacy in photo sharing social media

Still much room for improvement!

- Define mathematically the possible set of adversarial images
  - Model human eyes
- Why classifiers are vulnerable to semantic changes?
  - Colour shifting
  - Detail enhancement
- Certified adversarial perturbations
- Make semantic changes outside of digital domains in physical world