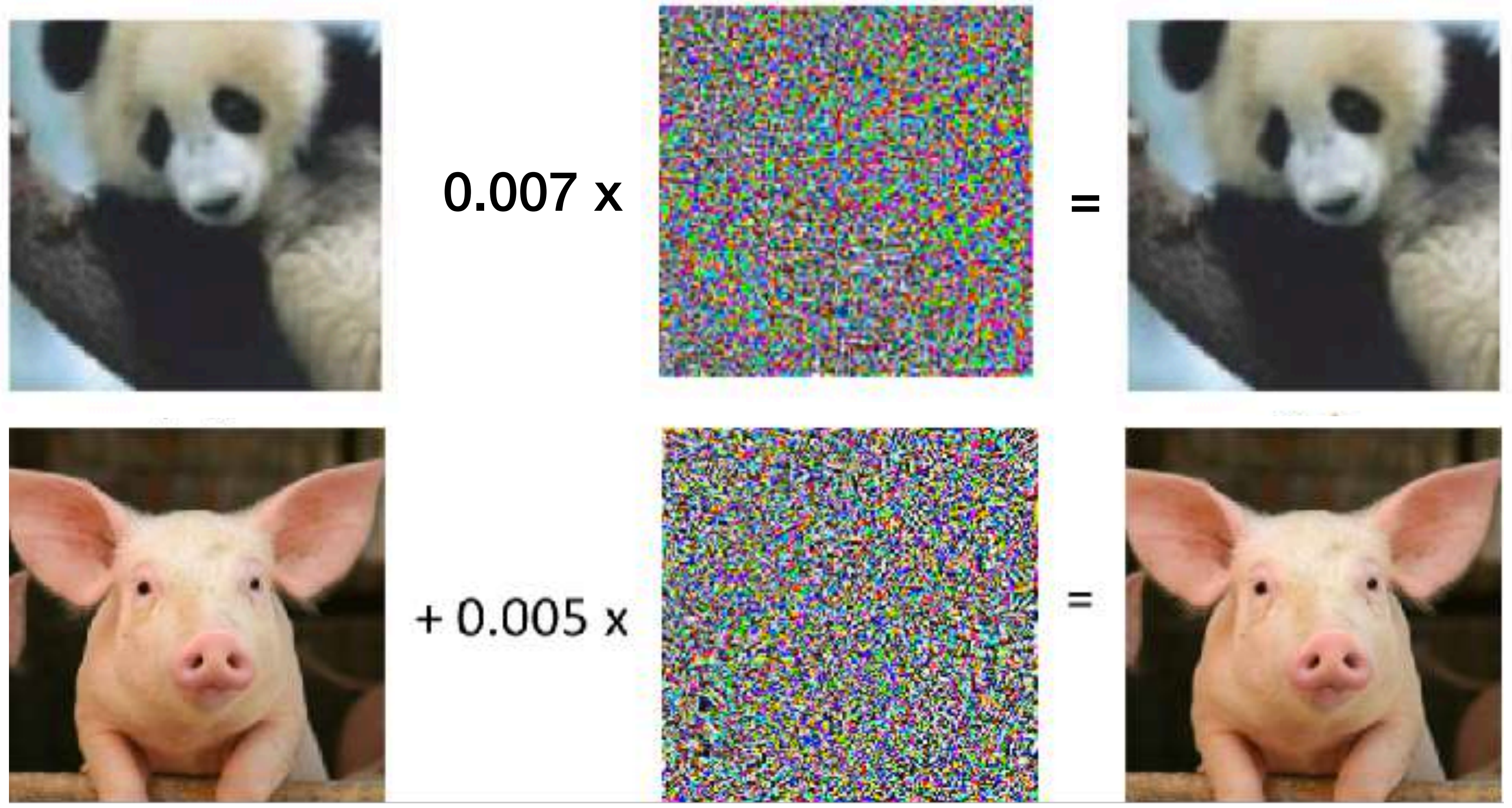
Beyond Classification: Extending and Leveraging Adversarial Examples

Yossi Adi
Bar-Ilan University
Feb 25, 2019

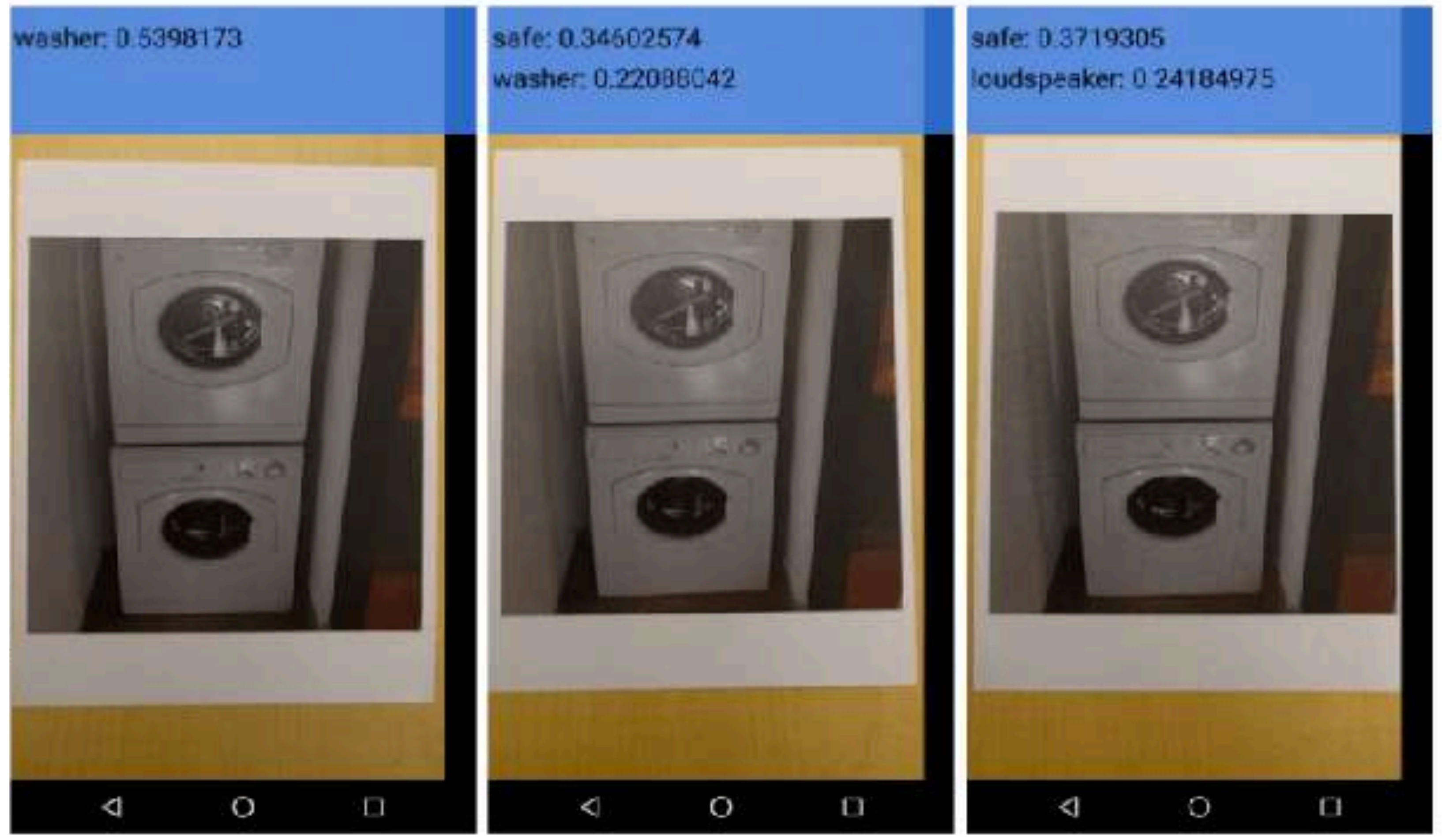
Beyond Classification: Extending and Leveraging Adversarial Examples

- Adversarial Examples - Overview
- Generating Adversarial Examples for Structured Tasks
- Houdini: Fooling Deep Structured Visual and Speech Recognition Models with Adversarial Examples
- Defences and Detection
- Steganography

Szegedy, Christian, et al. (2013)



Kurakin, A., Goodfellow, I., & Bengio, S. (2016)



(b) Clean image (c) Adv. image, $\epsilon = 4$ (d) Adv. image, $\epsilon = 8$

Brown, Tom B., et al. (2017)

Sharif, Mahmood, et al. (2017)

Eykholt, Kevin, et al. (2018)

Athalye, A., & Sutskever, I. (2018)

Most of the work was done for **images**
and for **classification** tasks

Recall

Solving: $$\eta = \arg \max_{\eta : \|\eta\|_p \leq \epsilon} \left(\nabla_x \bar{\ell}(x, y; \theta) \right)^\top \eta$$

$$\tilde{x} = x + \epsilon \cdot \mathrm{sign}(g) \qquad p = \infty$$

$$\tilde{x} = x + \epsilon \cdot g \qquad p = 2$$

Where, $g = \nabla_x \bar{\ell}(x, y; \theta)$

Measuring Performance

Target Predicted

The task loss function $\ell(y, \hat{y})$

Examples:

- Word Error Rate:

It is easy to recognize speech
It is easy to wreck a nice beach

1 substitutions
2 insertions

- Intersection Over Union:

Surrogate Loss Function

Negative Log Likelihood:

$$\bar{\ell}_{NLL}(x, y, \theta) = -log\ \mathbb{P}(y = y_t | x_t; \theta)$$

Hinge Loss (SVM like):

$$\bar{\ell}_{hinge}(x, y, \theta) = \ell(y, \hat{y}) - g_\theta(x, y) + g_\theta(x, \hat{y})$$

Does not necessarily have **connection** to the **task loss**

Houdini: Fooling Deep Structured **Visual** and **Speech** Recognition Models with **Adversarial Examples**

Let's look at the Hinge Loss Function

$$\bar{\ell}_{hinge}(x, y, \theta) = \ell(y, \hat{y}) - g_\theta(x, y) + g_\theta(x, \hat{y})$$

Network score for the target label

Network score for the predicted label

Houdini: Surrogate Loss Function

$$\boxed{\bar{\ell}_H(x, y; \theta) = \mathbb{P}_{\gamma \sim \mathcal{N}(0,1)} \Big[g_\theta(x, y) - g_\theta(x, \hat{y}) < \gamma \Big] \cdot \ell(y, \hat{y})}$$

Network score for the target label

Network score for the predicted label

Task Loss

Cisse, M. M., ***Adi, Y.****, Neverova, N., & Keshet, J. (NIPS, 2017).*

Houdini properties I

Gradients can be found analytically

$$\nabla_g \left[\mathbb{P}_{\gamma\sim\mathcal{N}(0,1)}\left[\gamma < \overbrace{g_\theta(x,\hat{y}) - g_\theta(x,y)}^{\delta g(\hat{y},y)}\right]\ell(y,\hat{y})\right]$$

$$= \nabla_g \left[\frac{1}{\sqrt{2\pi}}\int_{\delta g(\hat{y},y)}^{\infty} e^{-v^2/2} dv\right]\ell(y,\hat{y})$$

$$\boxed{\nabla_g\left[\bar{\ell}_H(\hat{y},y)\right] = \begin{cases} -\frac{1}{\sqrt{2\pi}}e^{-|\delta g(y,\hat{y})|^2/2}\ell(y,\hat{y}), & g = g_{\theta(x,y)} \\ \frac{1}{\sqrt{2\pi}}e^{-|\delta g(y,\hat{y})|^2/2}\ell(y,\hat{y}), & g = g_{\theta(x,\hat{y})} \\ 0, & otherwise \end{cases}}$$

*Cisse, M. M., **Adi, Y.**, Neverova, N., & Keshet, J. (NIPS, 2017).*

Houdini properties II

Lower Bound to the task loss (can be helpful for adversarial)

$$\begin{aligned}\bar{\ell}_H &= \mathbb{P}_{\gamma \sim \mathcal{N}(0,1)} \Big[g_\theta(x, y) - g_\theta(x, \hat{y}) < \gamma \Big] \cdot \ell(y, \hat{y}) \\ &\leq \ell(y, \hat{y})\end{aligned}$$

Cisse, M. M., ***Adi, Y.****, Neverova, N., & Keshet, J. (NIPS, 2017).*

Houdini

- We generate adversarial examples using Houdini to three different structured tasks
    - Image Segmentation
    - Pose Estimation
    - Automatic Speech Recognition

Image Segmentation

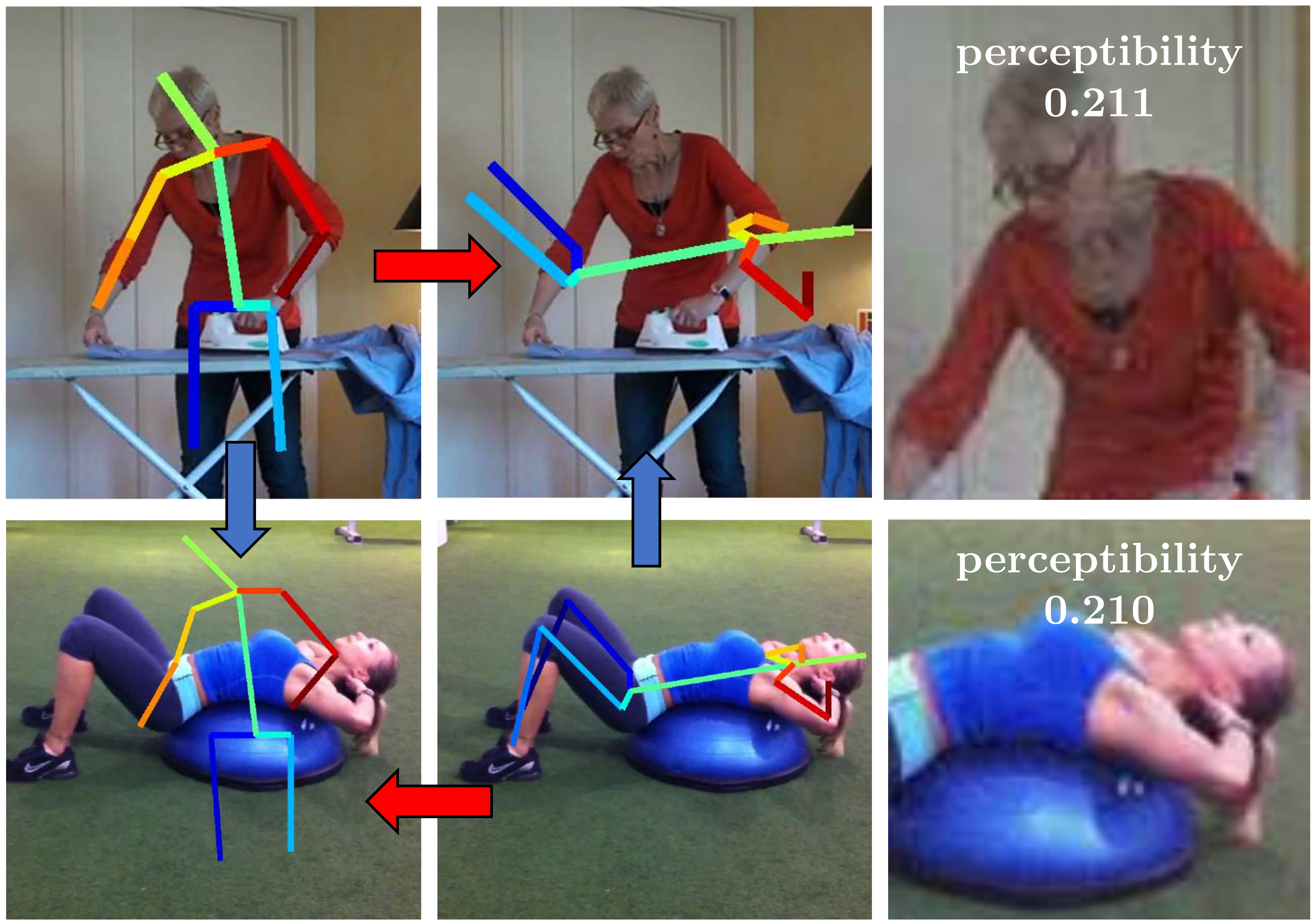
source image | initial prediction | perturbed image | noise | adversarial prediction

Image Segmentation

Image Segmentation

Image Segmentation

Pose Estimation



Automatic Speech Recognition

Original wav form:
Transcription:
if she could only see Phronsie for just one moment

Adversarial wav form:
Transcription:
if she ou down take shee throwns purhdress luon ellwon

Automatic Speech Recognition

	$\epsilon = 0.3$		$\epsilon = 0.2$		$\epsilon = 0.1$		$\epsilon = 0.05$	
	WER	CER	WER	CER	WER	CER	WER	CER
CTC	68	9.3	51	6.9	29.8	4	20	2.5
Houdini	96.1	12	85.4	9.2	66.5	6.5	46.5	4.5

Automatic Speech Recognition

Adversarial examples are defined to be indistinguishable to humans eye/ear

ABX Testing: We generated 100 audio samples of adversarial examples and performed an ABX test with about 100 humans.

Automatic Speech Recognition

<u>Ground truth Transcription</u>
The fact that a man can recite a poem does not show he remembers any previous occasion on which he has recited it or read it.

<u>G-Voice transcription of the original example:</u>
The fact that a man can decide a poem does not show he remembers any previous occasion on which he has work cited or read it.

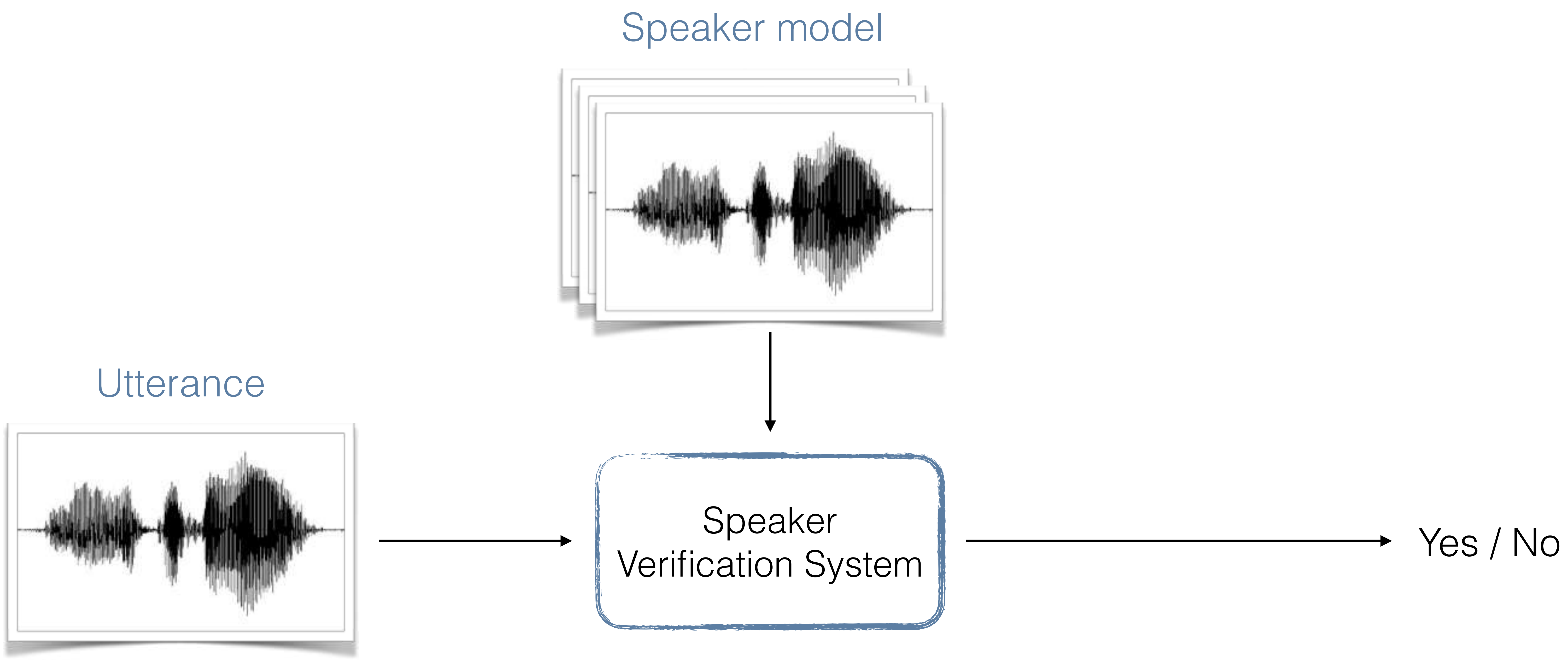
<u>G-Voice transcription of the adversarial example:</u>
The fact that I can rest I'm just not sure that you heard there is any previous occasion I am at he has your side it or read it.

<u>Ground truth Transcription</u>
Her bearing was graceful and animated she led her son by the hand and before her walked two maids with was lights and silver candlesticks

<u>G-Voice transcription of the original example:</u>
The bearing was graceful an animated she let her son by the hand and before her walked two maids with was lights and silver candlesticks

<u>G-Voice transcription of the adversarial example:</u>
Marry was grateful then admitted she let her son before the walks to Mays would like slice furnace filter count six.

Speaker Verification



Model

$$\hat{y} \in \{0, 1\}$$

Binary classifier $\uparrow$

$$s \in [0, 1]$$

Cosine-similarity $\uparrow$

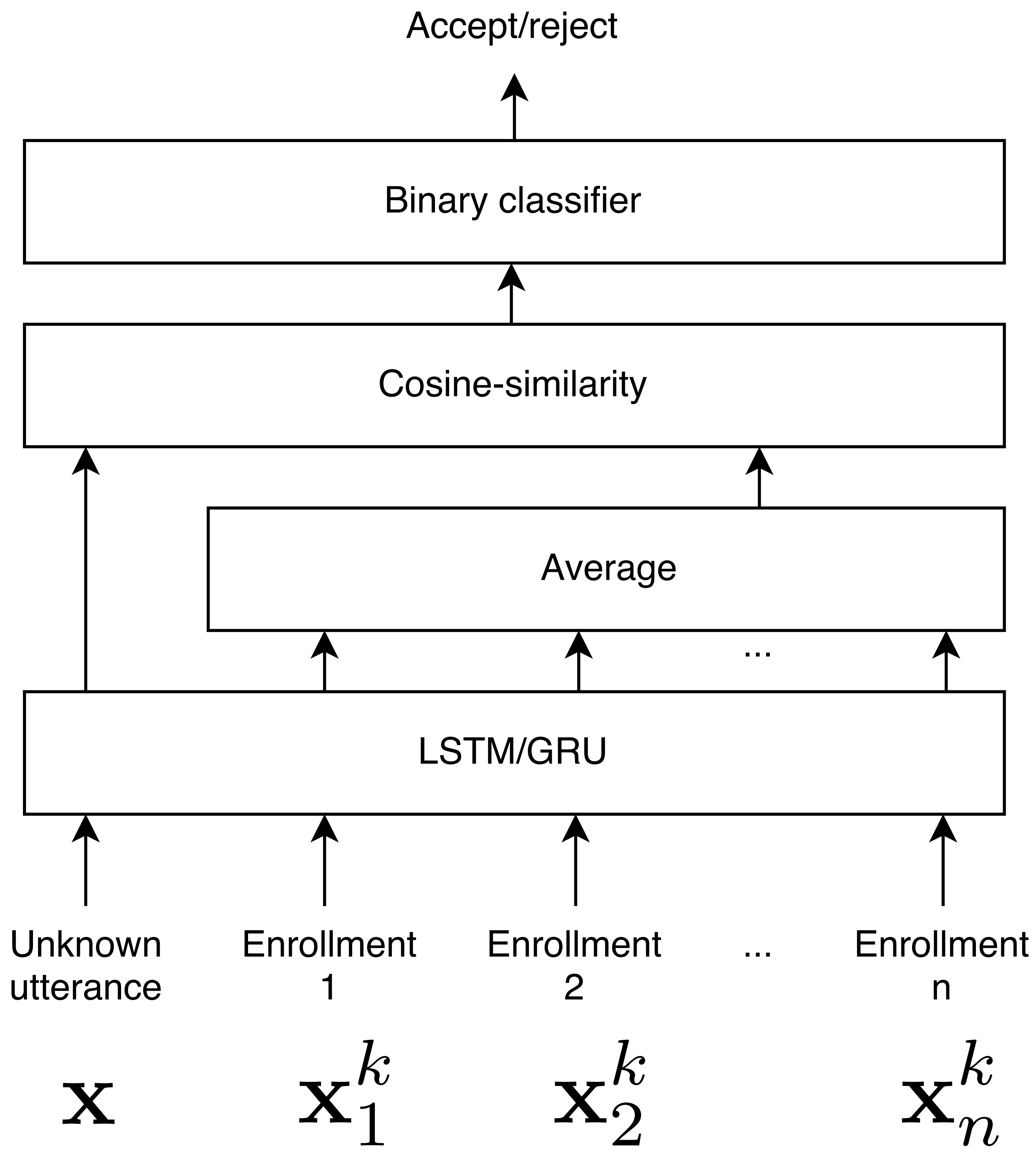
$$(\mathrm{RNN}(\mathbf{z}), \frac{1}{n}\sum_{i=0}^{n} \mathrm{RNN}(\mathbf{Z}^k))$$

RNN $\uparrow$

$$(\mathbf{z}, \mathbf{Z}^k)$$

Feature-extraction $\uparrow$

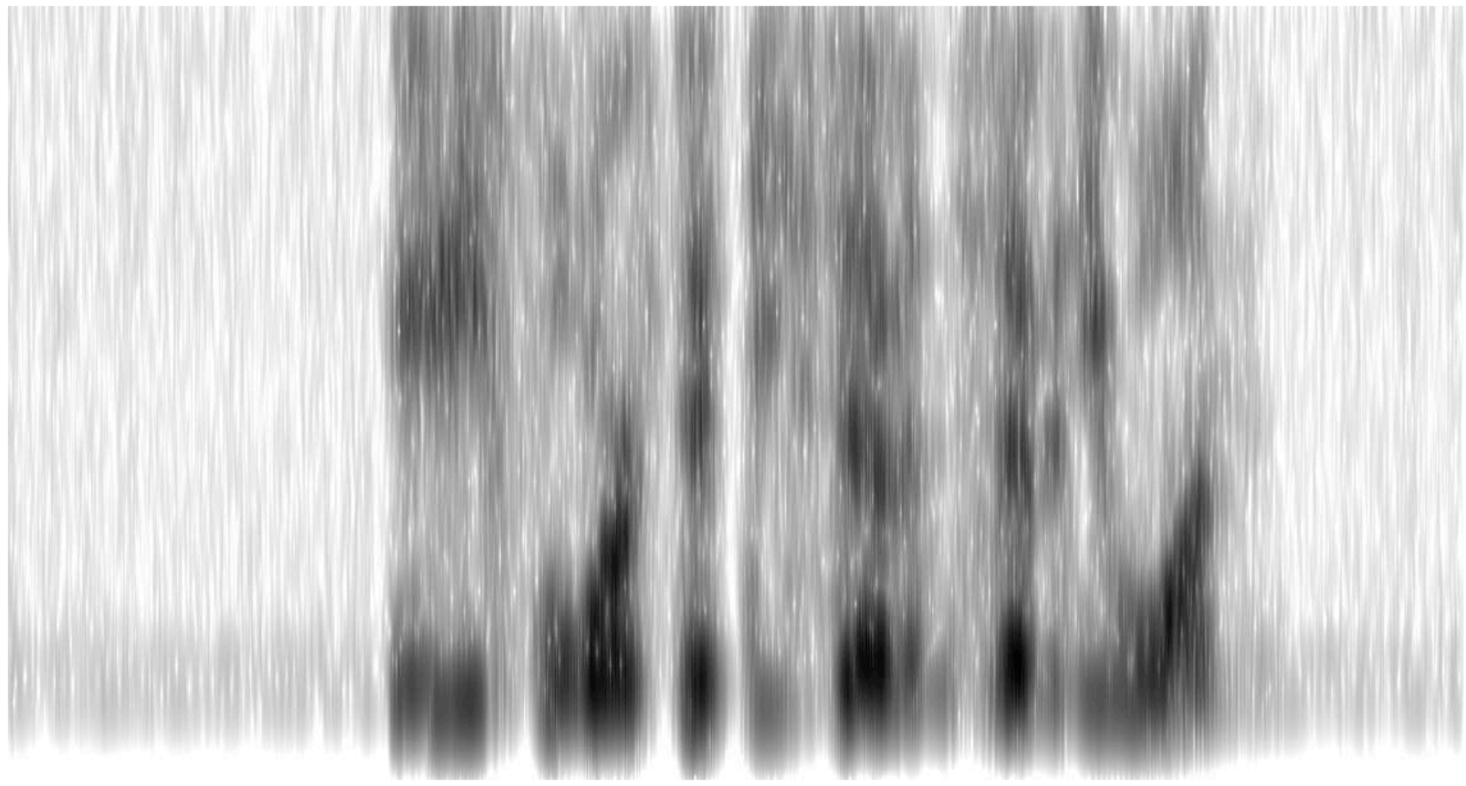
$$(\mathbf{x}, \mathbf{X}^k)$$



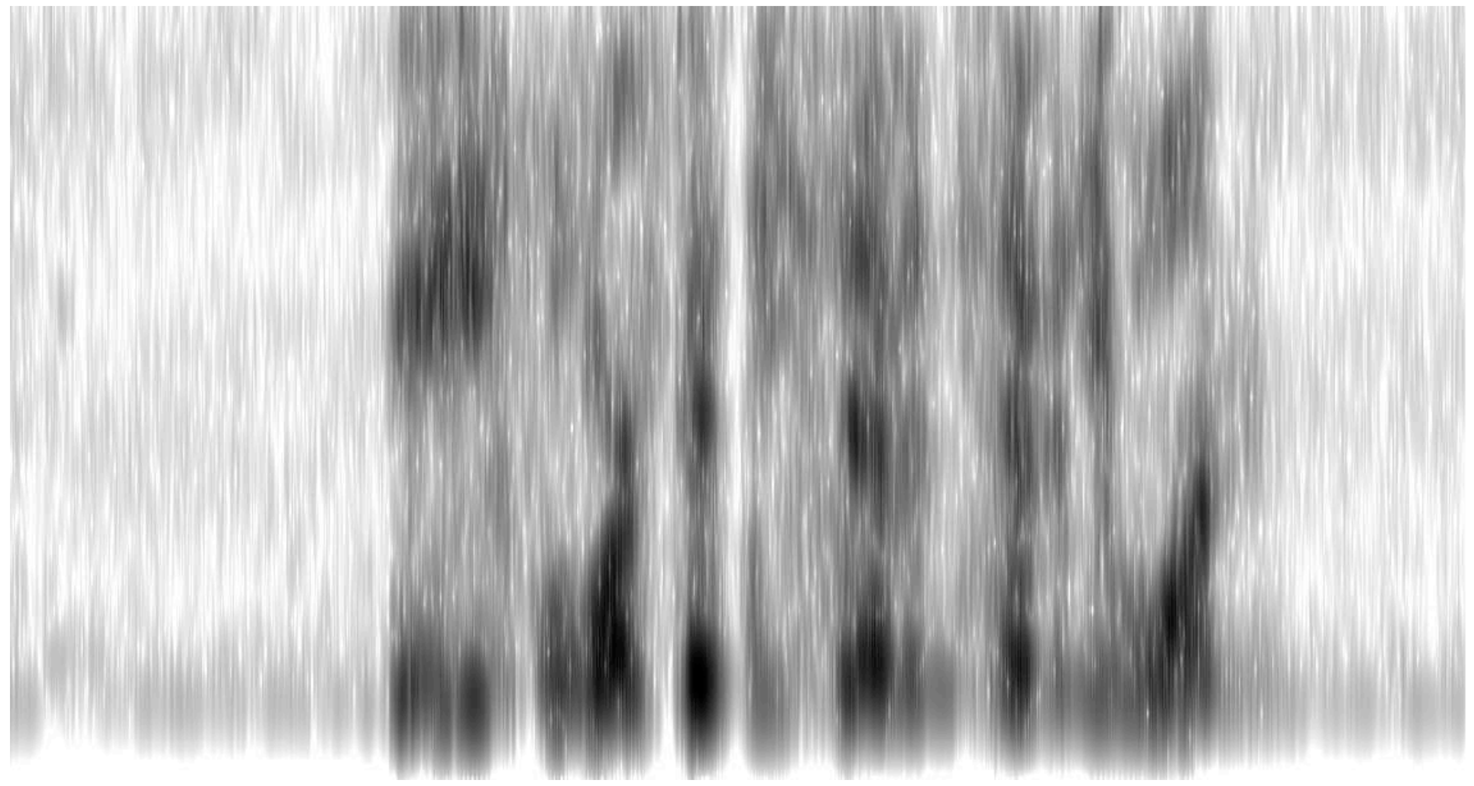
Heigold, Georg, et al. *(ICASSP, 2016).*

Speaker Verification

Original



Adversarial



*Kreuk, F., **Adi, Y.**, Cisse, M., & Keshet, J. (ICASSP, 2018).*

Defences and **Detection**

Defenses

- Adversarial Training
- Adversarial Logit Pairing
- Denoising Auto-Encoder
- Regularization Methods (Parseval Networks)
- Input Transformation

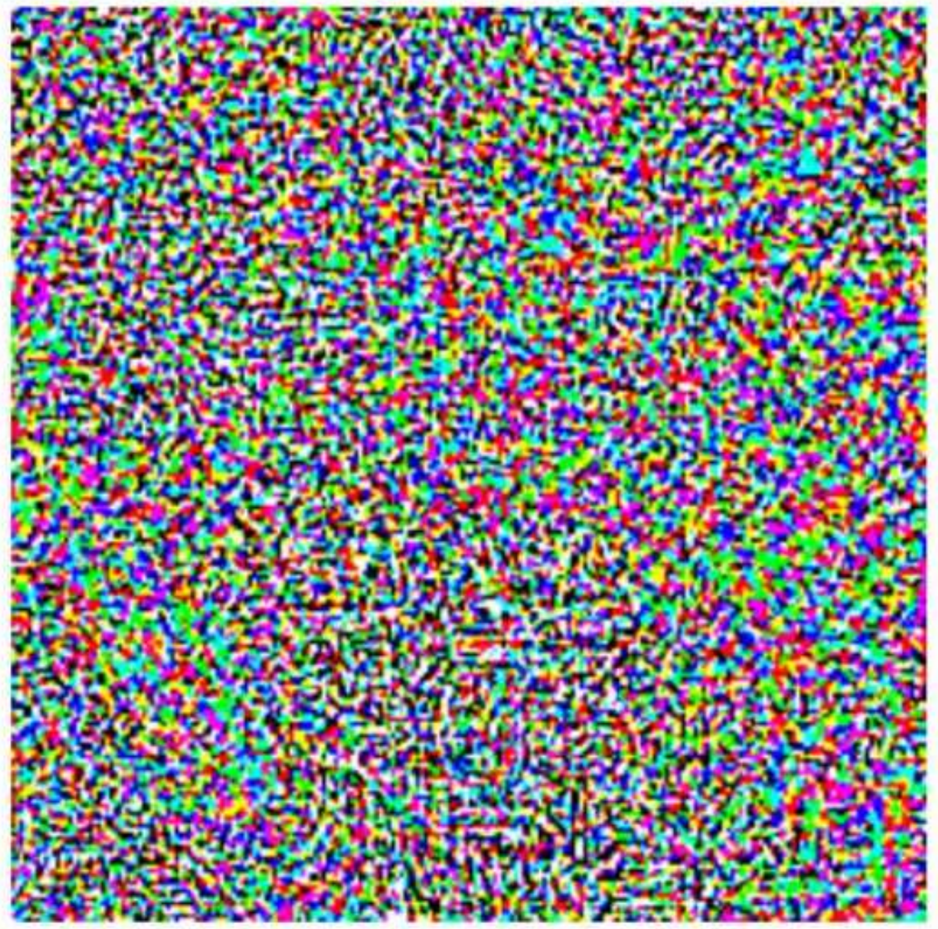
Recall

“gibbon”
99.3%

=

“panda”
57.7% confidence

$+ .007 \times$



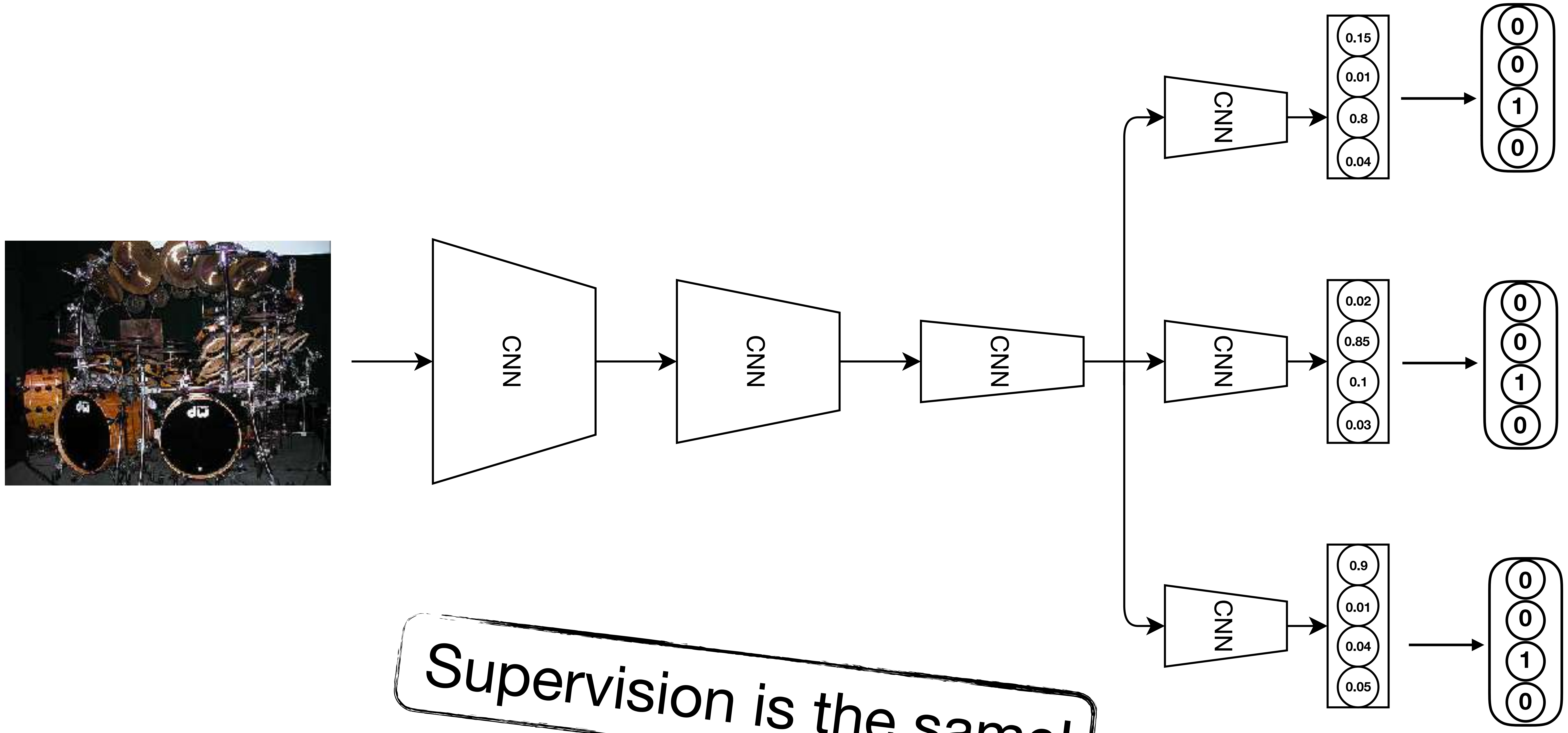
“nematoda”
8.2% confidence

Detecting Adversarial Examples

- Using softmax outputs turns out to be useful for detecting adversarial examples
- Similar approaches use an ensemble of classifiers to detect adversarial examples
- We would like to adopt the second approach while using shared representation
    - Closely related to detecting out-of-distribution/wrongly classified examples

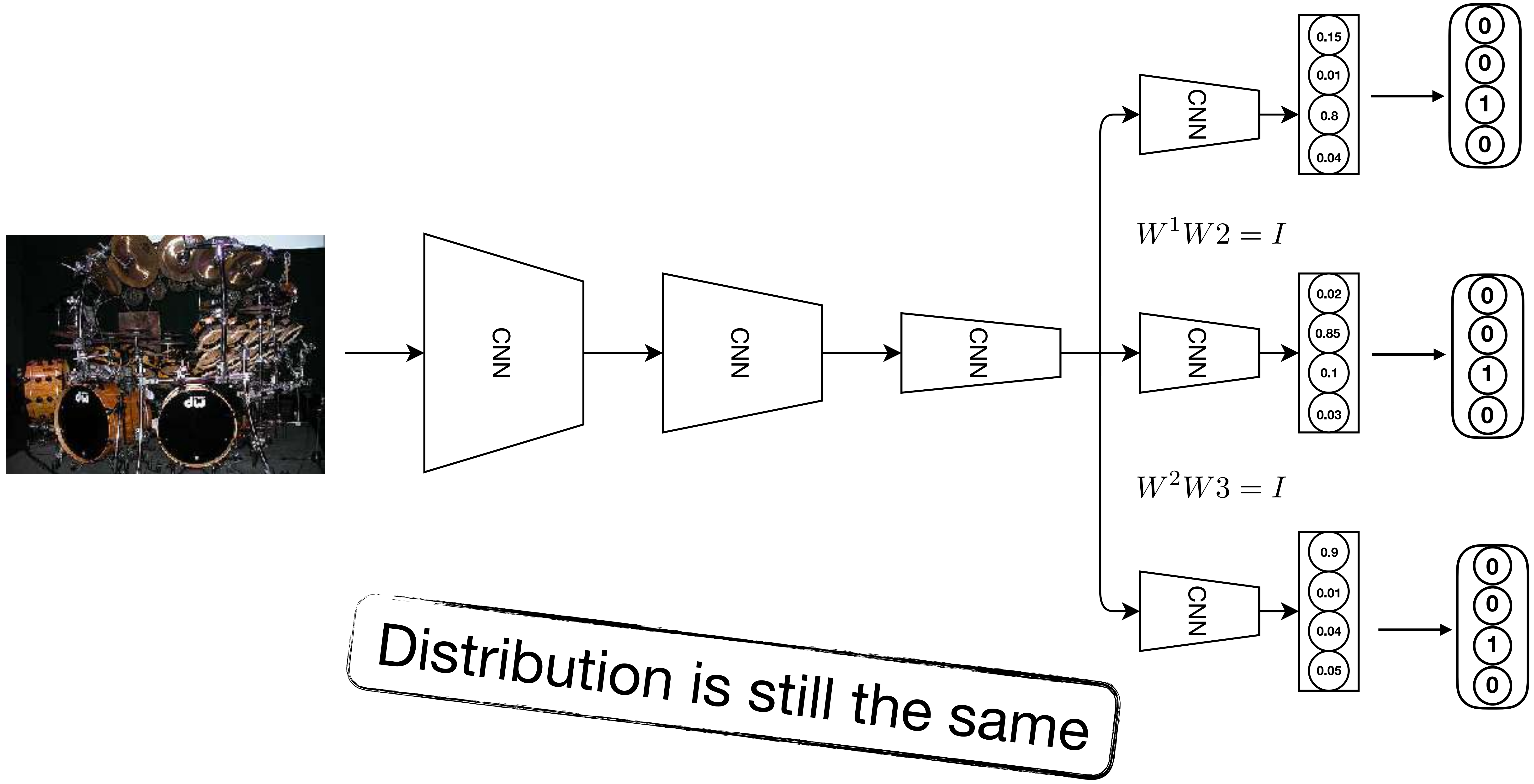
Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. (NIPS, 2017).
Bagnall, Alexander, Razvan Bunescu, and Gordon Stewart. (arXiv, 2017).

Detecting Adversarial Examples



Supervision is the same!

Detecting Adversarial Examples



Littwin, E. and Wolf, L., (CVPR, 2016).

Detecting Adversarial Examples

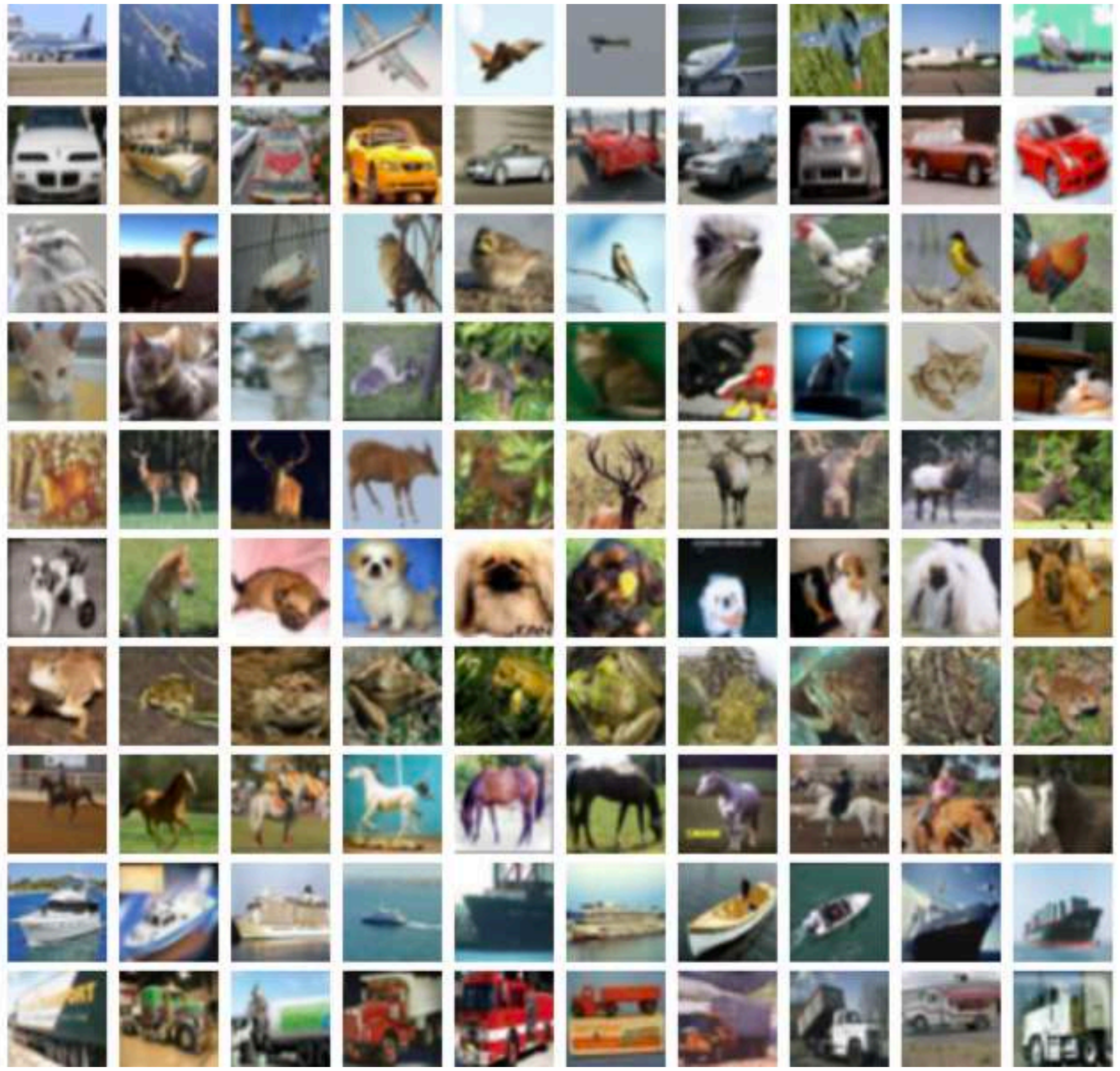
- Idea:
  - Generate word embeddings to the classes for different corpuses
  - Train the classifier to output those representations
  - Semantic hierarchy between the labels
  - “Ensemble like” training

*Shalev, G., **Adi, Y.**, & Keshet, J. (NeurIPS 2018).*

Proposed Model

K

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck



<0.1, 0.03, 0.89, … , >, …, <0.4, 0.91, 0.11, … , >

<0.2, 0.15, 0.72, … , >, …, <0.5, 0.21, 0.66, … , >

…

…

…

…

…

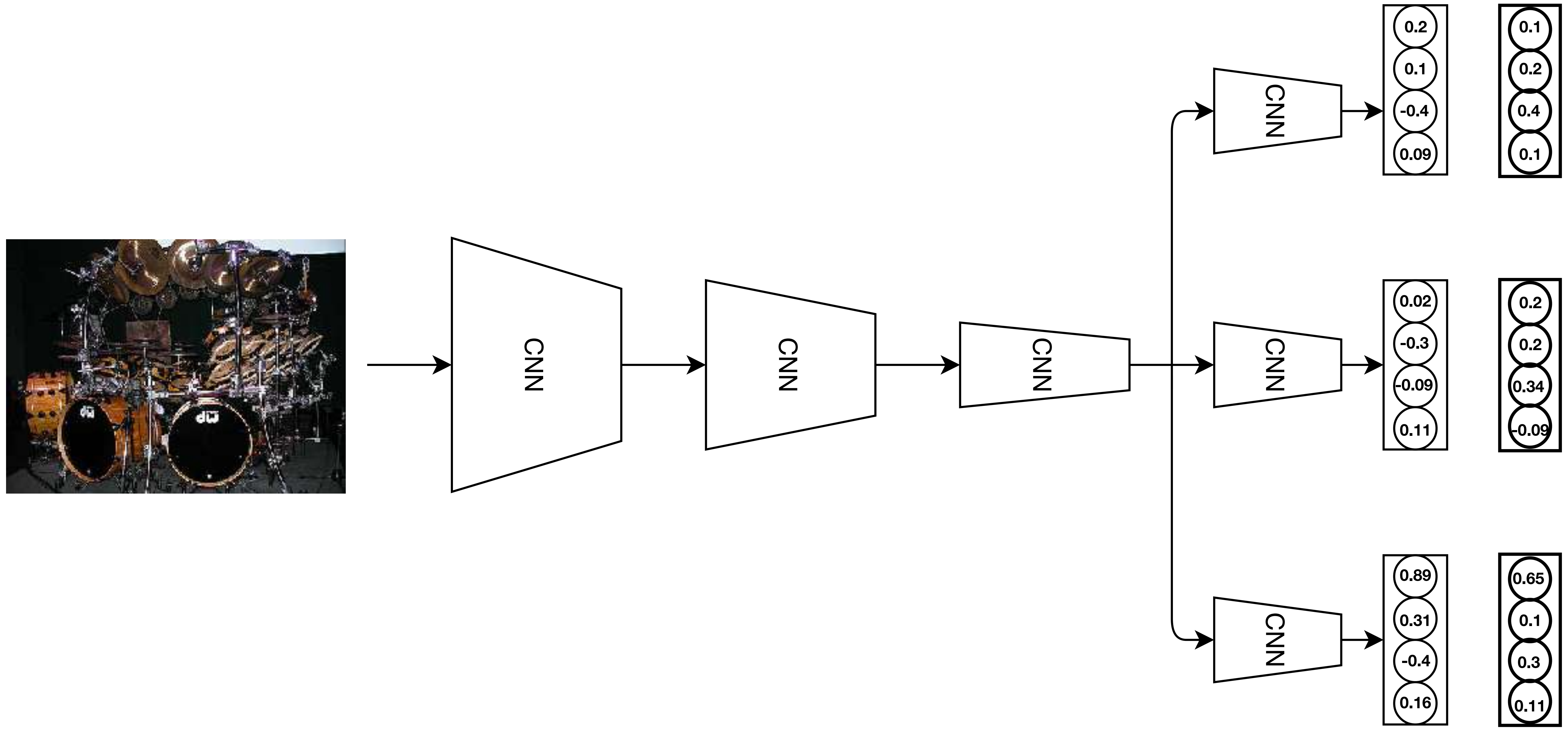
…

…

<0.41, 0.5, 0.34, … , >, …, <0.2, 0.67, 0.29, … , >

Num labels

Proposed Model



“Drums” Word Embedding 1 (wiki db)

“Drums” Word Embedding 2 (music magazines db)

“Drums” Word Embedding 3 (twitter db)

*Shalev, G., **Adi, Y.**, & Keshet, J. (NeurIPS 2018).*

Proposed Model - Formally

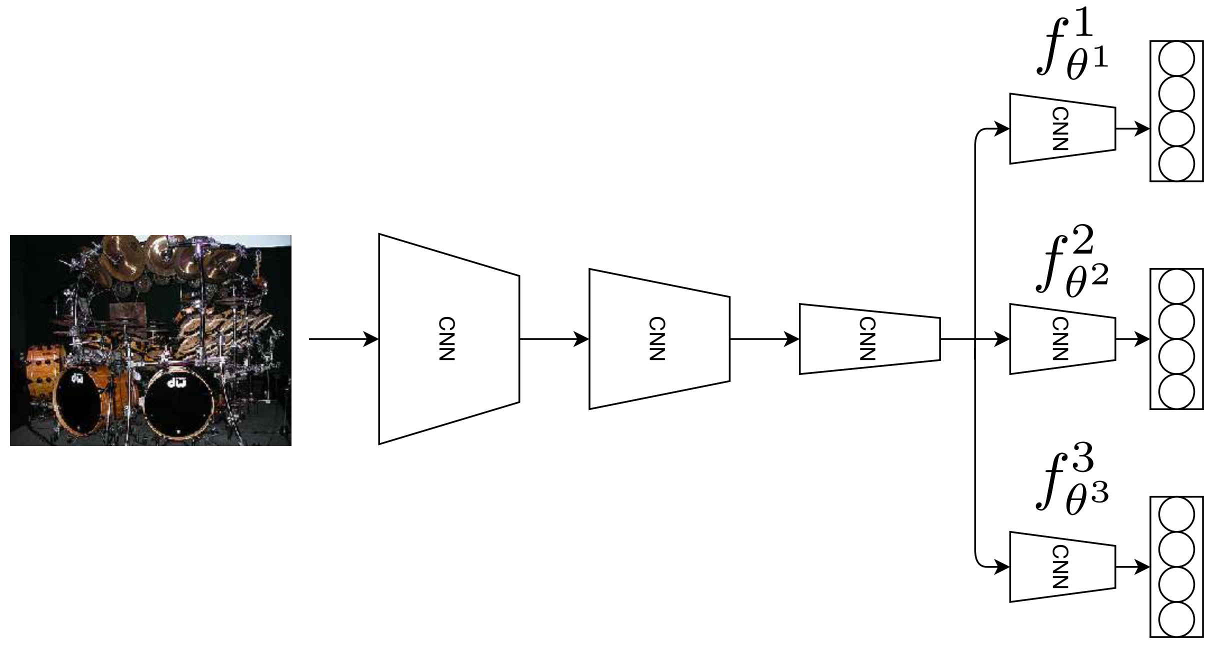
- Given a training set of: $\mathcal{S}_{\text{train}} = \{(\mathbf{x}_i, \mathbf{e}^1(y_i), ..., \mathbf{e}^K(y_i))\}_{i=1}^M$

- Our goal is to minimize the following surrogate-loss function:

$$\bar{\ell}(\mathbf{x}, y; \theta) = \sum_{k=1}^{K} d_{\cos}(\mathbf{e}^k(y), \boldsymbol{f}^k_{\theta^k}(\mathbf{x})).$$

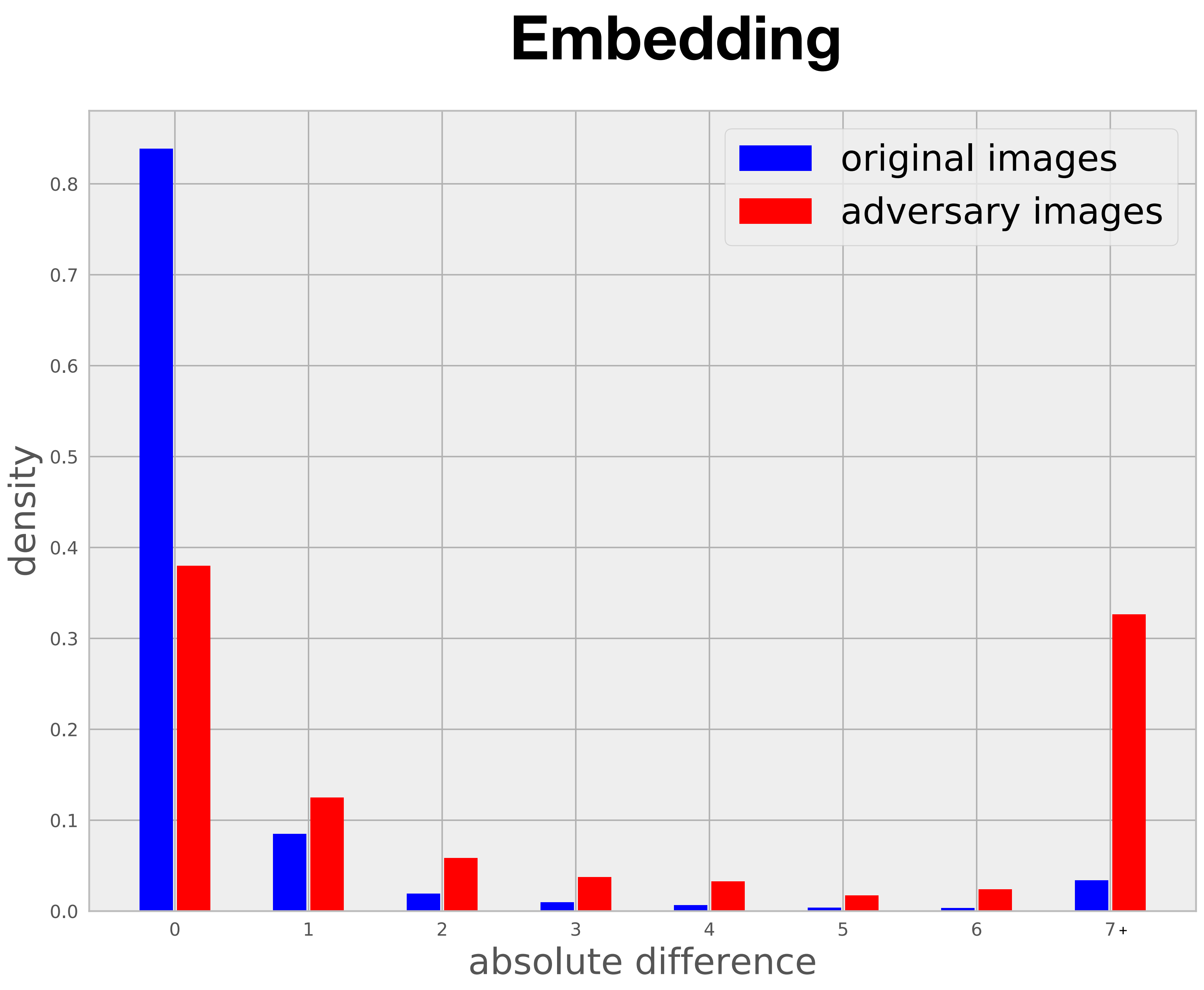
- At inference time, classify new example as follows:

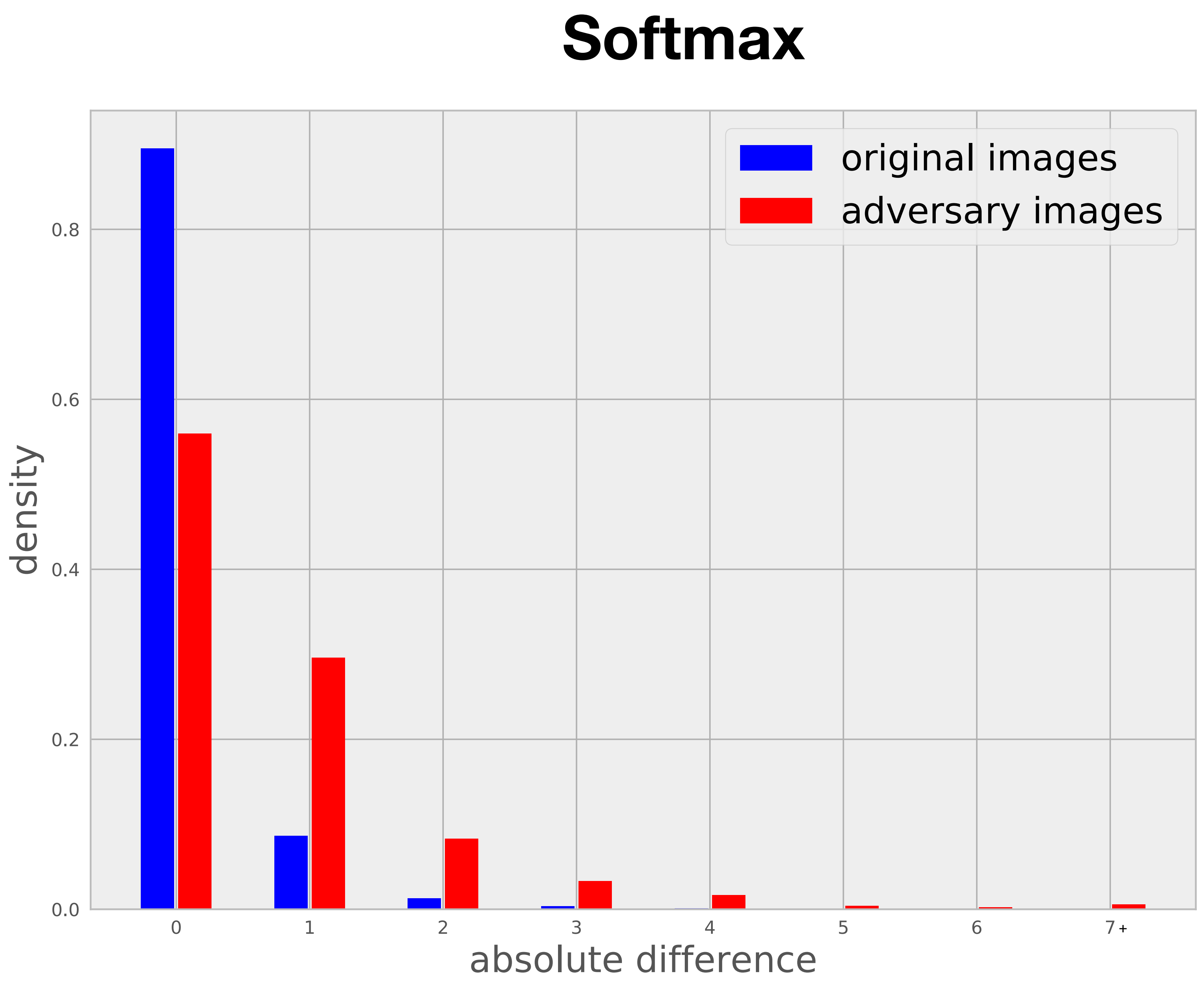
$$\hat{y} = \arg\min_{y \in \mathcal{Y}} \sum_{k=1}^{K} d_{\cos}(\mathbf{e}^k(y), \boldsymbol{f}^k_{\theta^k}(\mathbf{x})).$$



Shalev, G., **Adi, Y.**, & Keshet, J. (NeurIPS 2018).

Results (ii) - Adversarial Examples

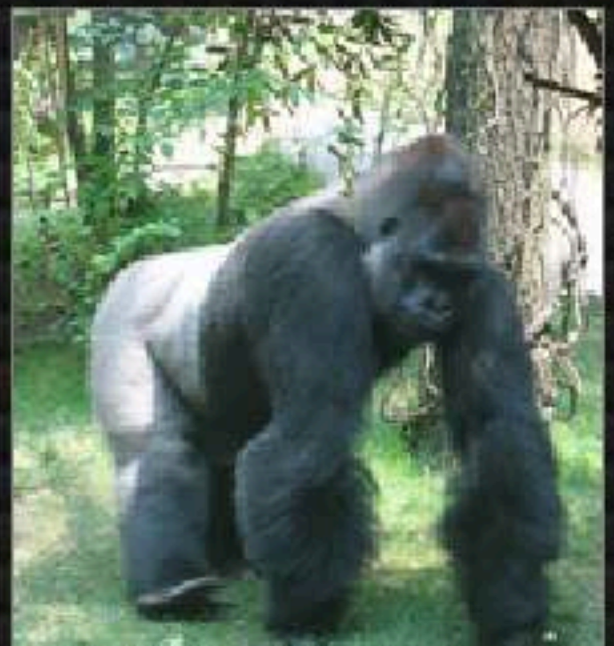




- *To qualify that, we fixed the false rejection rate in both methods to be 3%. In this setting, the ensemble reaches 15.41% detection rate while our model reaches 28.64% detection rate*

Can we use this for our own **benefit**?

DeepCAPTCHA test

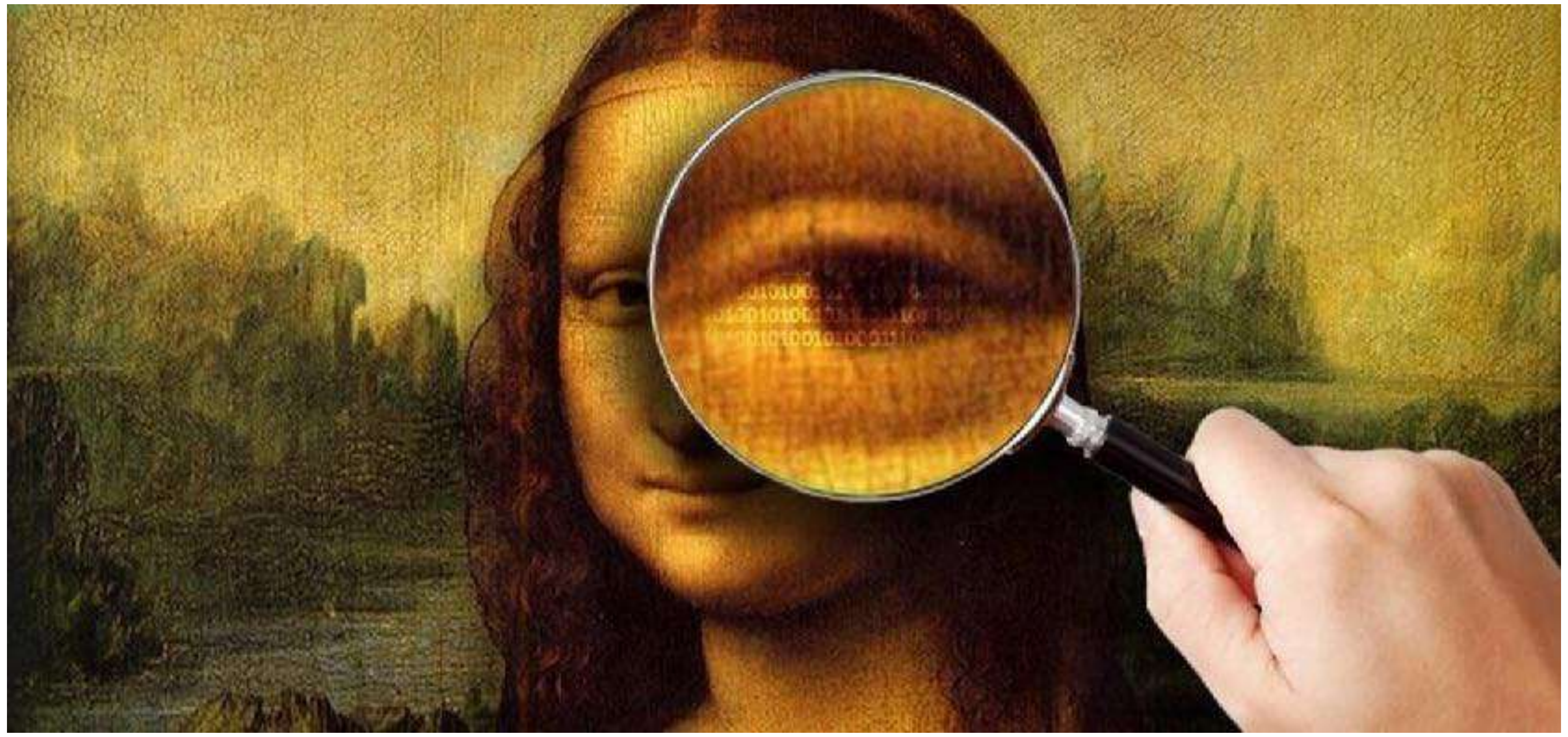


Osadchy, Margarita, et al. (IACR Cryptology, 2016)

Can we use this mechanism to **transmit hidden information** instead of causing misclassification?

Steganography

- *steganos* meaning "covered, concealed, or protected", and *graphein* meaning “writing”
- Steganography is the practice of *concealing* a message within another message
- Goal: hide the existence of the hidden message



Steganography Example

Cover Image

Image to Hide

Cover + Hidden Image

Extracted Image

Steganography

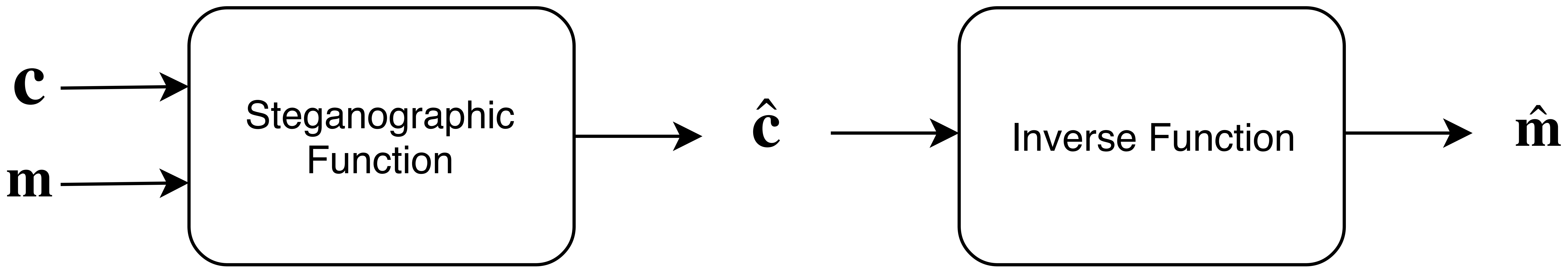
- Traditional steganography used Invisible Inks, Tattoos, etc. (Herodotus, Ancient romans, World War II, etc.)

- Modern (computer) steganography is based on two observations:
  - Some kinds of digital data can be altered without losing functionality (images, audio, etc.)
  - Human inability to distinguish minor changes in such digital data imparts redundancy, which can be exploited

Problem Settings

- Consider two messages, *carrier* ($\mathbf{c}$) and *message* ($\mathbf{m}$)
- The steganographic system gets as input: $\mathbf{c}$ and $\mathbf{m}$, and outputs $\hat{\mathbf{c}}$ and $\hat{\mathbf{m}}$
- The output of the steganography system should fulfill the following requirements:
  - $\hat{\mathbf{c}}$ and $\hat{\mathbf{m}}$ should be perceptually similar to $\mathbf{c}$ and $\mathbf{m}$
  - $\hat{\mathbf{m}}$ should be recoverable from $\hat{\mathbf{c}}$
  - A human listener should not be able to detect the presence of the hidden message $\mathbf{m}$ in $\hat{\mathbf{c}}$

Steganography



Steganography Example

Carrier $\mathbf{c}$

Message $\mathbf{m}$

Modified Carrier $\hat{\mathbf{c}}$

Extracted Message $\hat{\mathbf{m}}$

Steganography

- Modern approaches for steganography are based on some signal redundancy
- The most common approach is Least Significant Bit (LSB) Encoding & Decoding

Carrier Image

R(1100 1010)
G(0010 0110)
B(1110 1110)

Message Image

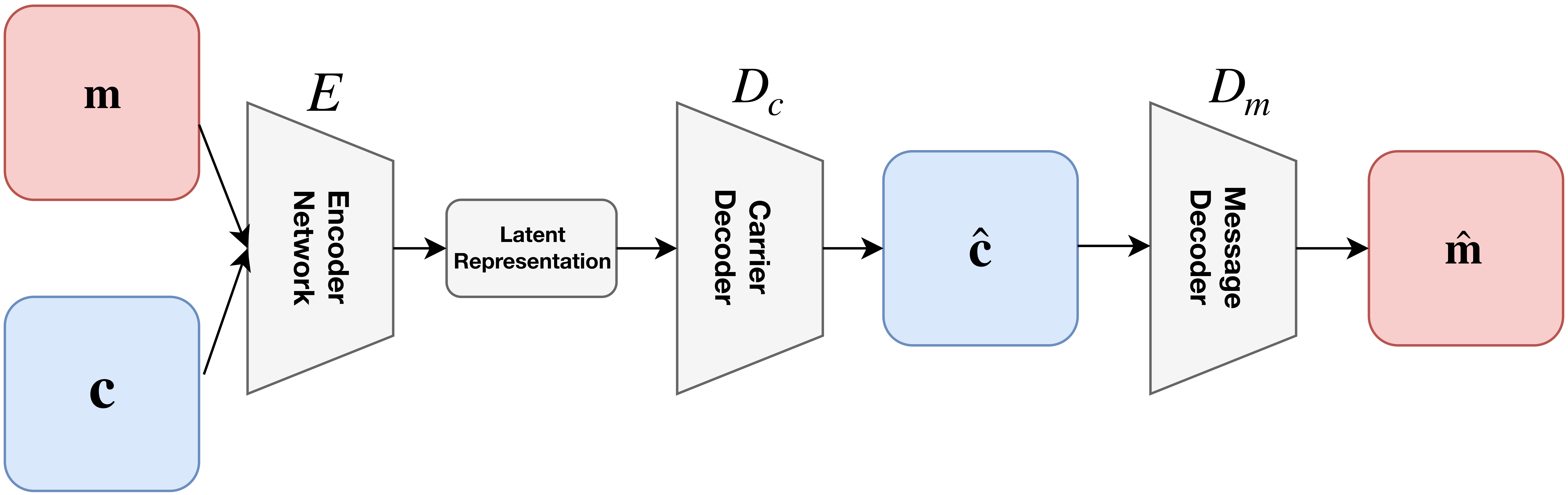
R(1000 1010)
G(1100 0001)
B(1111 0110)

Modified Carrier Image

R(1100 1000)
G(0010 1100)
B(1110 1111)

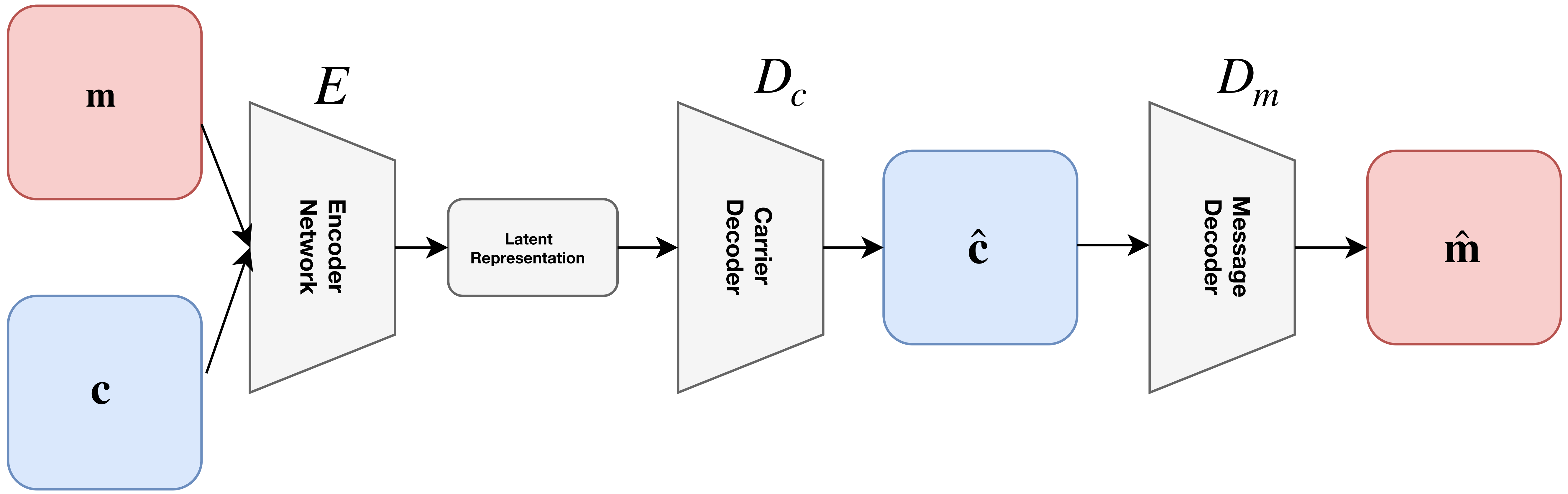
Neural network as a **steganographic function**

Image Steganography



Shumeet Baluja, (NeurIPS, 2017).

Image Steganography

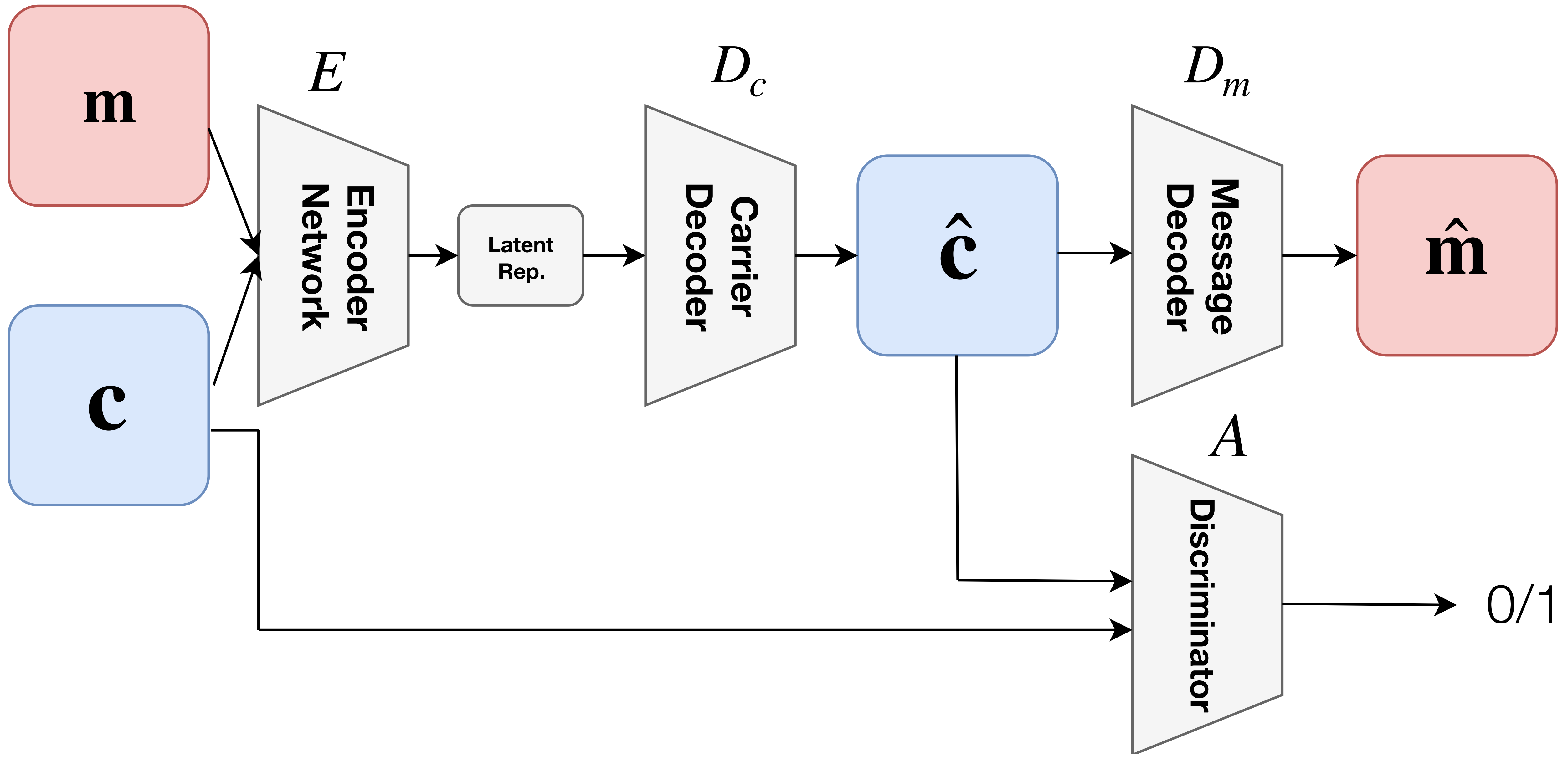


Objective:

$$\mathcal{L}(\mathbf{c}, \mathbf{m}) = \lambda_c \|\mathbf{c} - D_c(E(\mathbf{c}, \mathbf{m}))\|_2^2 + \lambda_m \|\mathbf{m} - D_m(D_c(E(\mathbf{c}, \mathbf{m})))\|_2^2$$

Shumeet Baluja, (NeurIPS, 2017).

Image Steganography

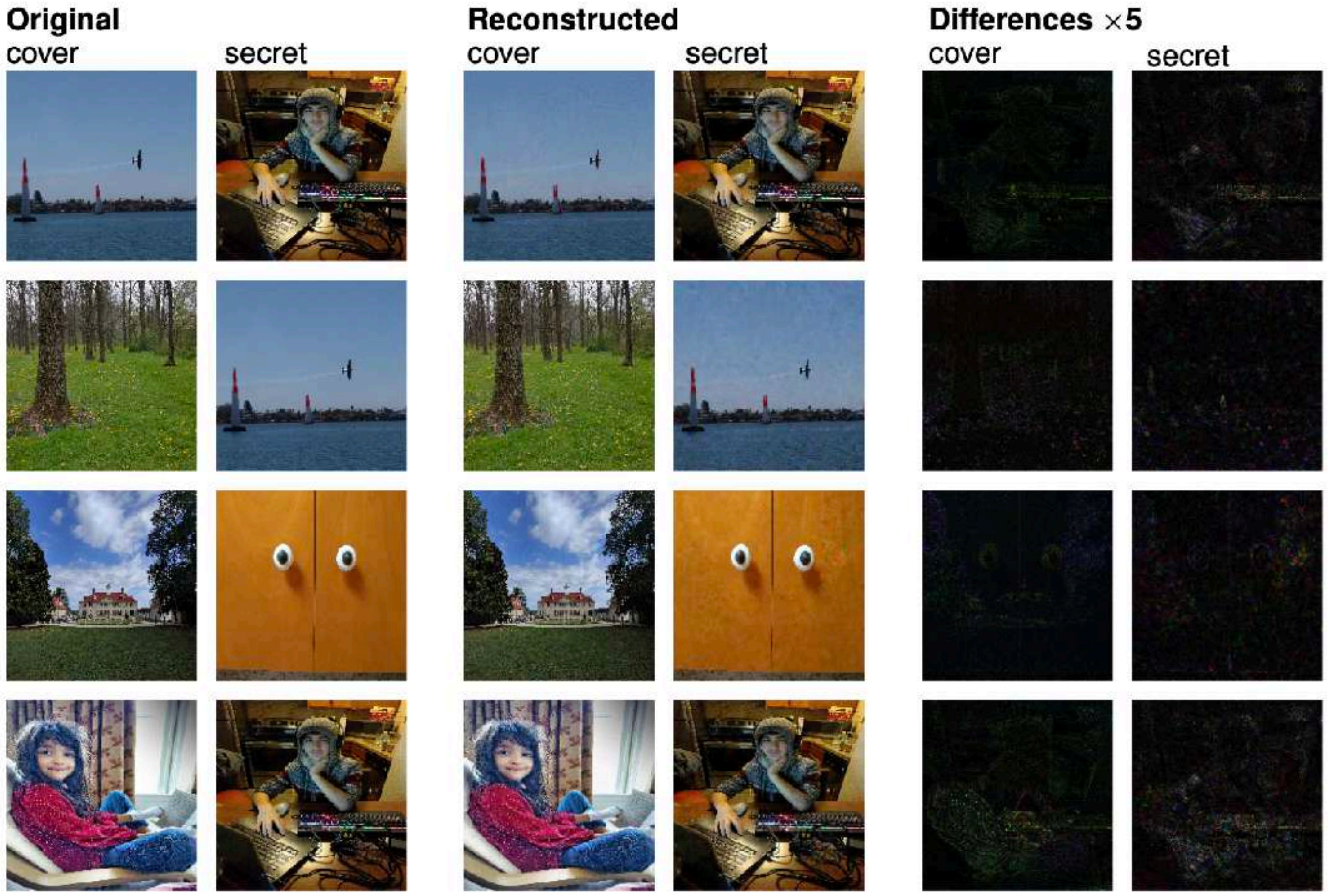


<u>Objective</u>:

$$
\begin{aligned}
\mathcal{L}(\mathbf{c}, \mathbf{m}) = {} & \lambda_c ||\mathbf{c} - D_c(E(\mathbf{c}, \mathbf{m}))||_2^2 \\
& + \lambda_m ||\mathbf{m} - D_m(D_c(E(\mathbf{c}, \mathbf{m})))||_2^2 \\
& + \lambda_g(-\log A(\hat{\mathbf{c}}))
\end{aligned}
$$

$$\mathcal{L}_{dis}(\mathbf{c}, \hat{\mathbf{c}}) = -\log(A(\mathbf{c})) - \log(1 - A(\hat{\mathbf{c}}))$$

Examples



Shumeet Baluja, (NeurIPS, 2017).

Examples

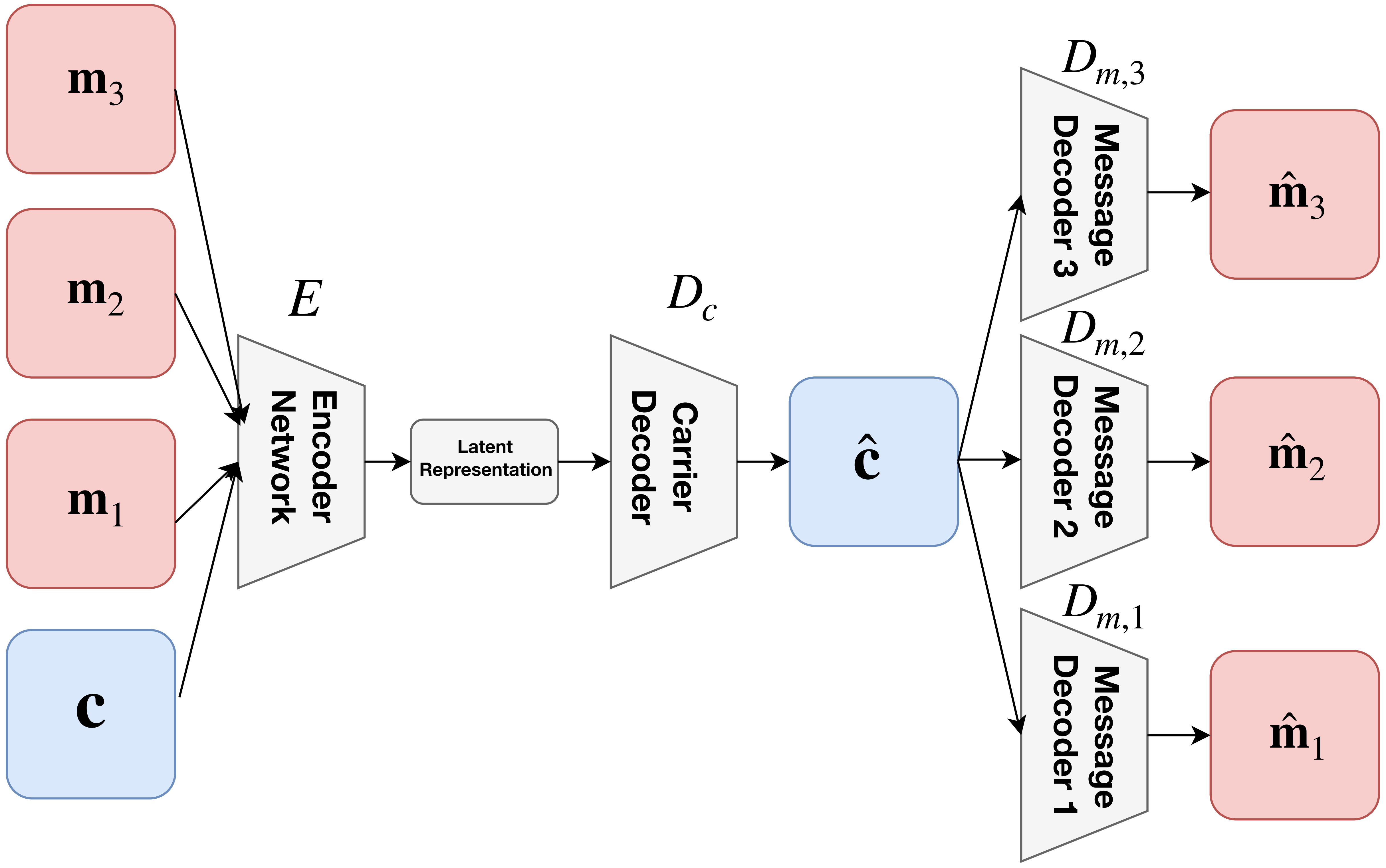
Shumeet Baluja, (NeurIPS, 2017).

Examples

Shumeet Baluja, (NeurIPS, 2017).

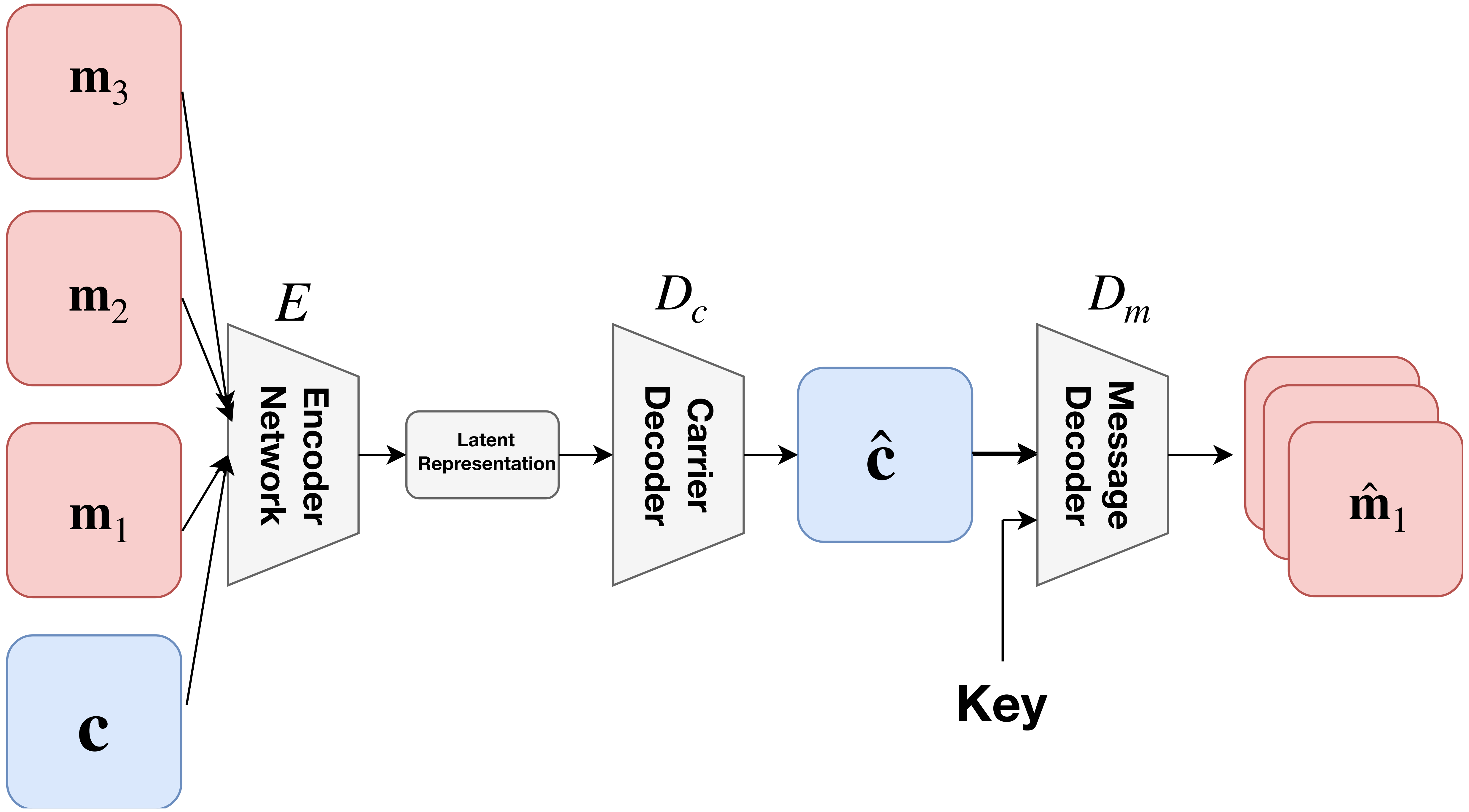
Can we embed **more than one message?**

Multiple Decoders



$$\mathcal{L}(\mathbf{c}, \{\mathbf{m}_i\}_{i=1}^k) = \lambda_c \|\mathbf{c} - D_c(E(\mathbf{c}, \{\mathbf{m}_i\}_{i=1}^k))\|_2^2 + \lambda_m \sum_{i=1}^{k} \|\mathbf{m}_i - D_{m,i}(D_c(E(\mathbf{c}, \{\mathbf{m}_i\}_{i=1}^k)))\|_2^2$$

Conditional Decoder



$$\mathcal{L}(\mathbf{c}, \{\mathbf{m}_i\}_{i=1}^{k}) = \lambda_c \|\mathbf{c} - D_c(E(\mathbf{c}, \{\mathbf{m}_i\}_{i=1}^{k}))\|_2^2 + \lambda_m \sum_{i=1}^{k} \|\mathbf{m}_i - D_m(D_c(E(\mathbf{c}, \{\mathbf{m}_i\}_{i=1}^{k})), q_i)\|_2^2$$

Summary and Future Work

- Adversarial examples
  - Structured Tasks
  - Speaker Verification
- Defences and Detection
- Steganography

Thanks!