https://github.com/facebookresearch/ssl_watermarking

Watermarking Images in Self-Supervised Latent Spaces

Pierre Fernandez[1,2], Alexandre Sablayrolles[1], Teddy Furon[2], Hervé Jégou[1], Matthijs Douze[1]

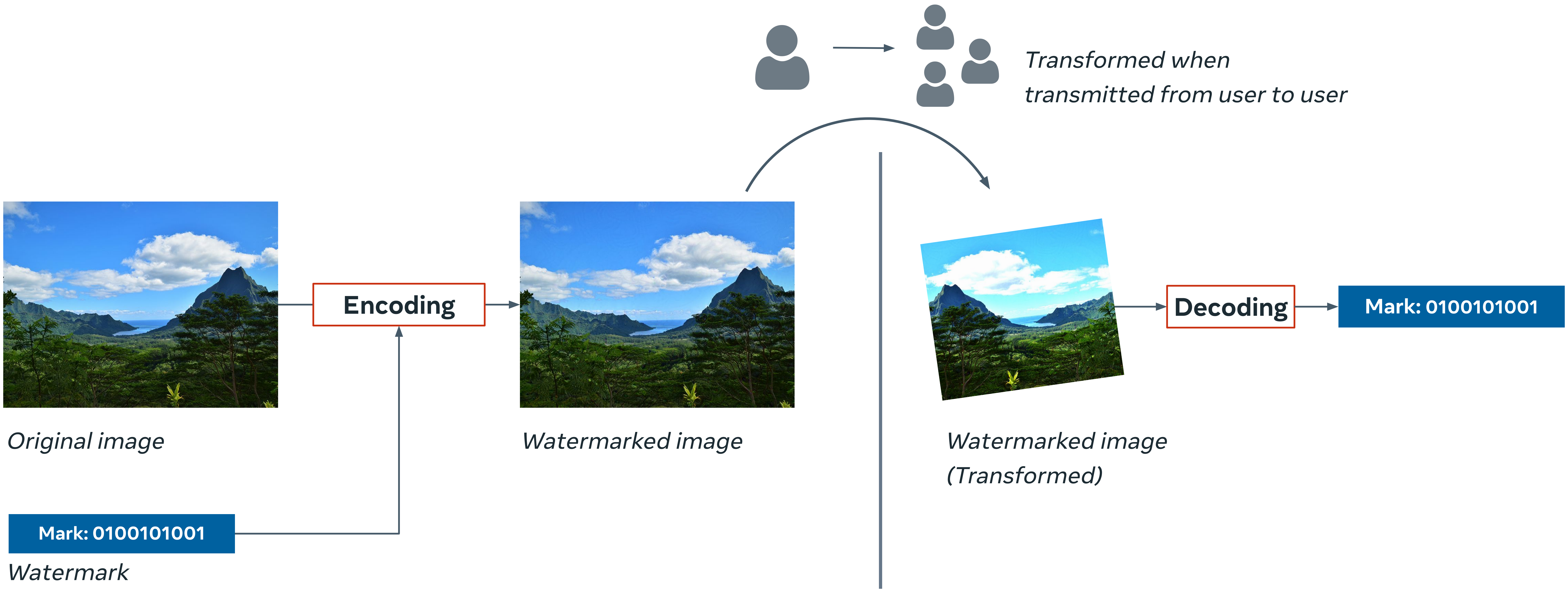
[1] **Meta AI**

[2] **Univ. Rennes, Inria, CNRS, IRISA**

icassp 2022
Singapore, China, Online

Inria Meta AI

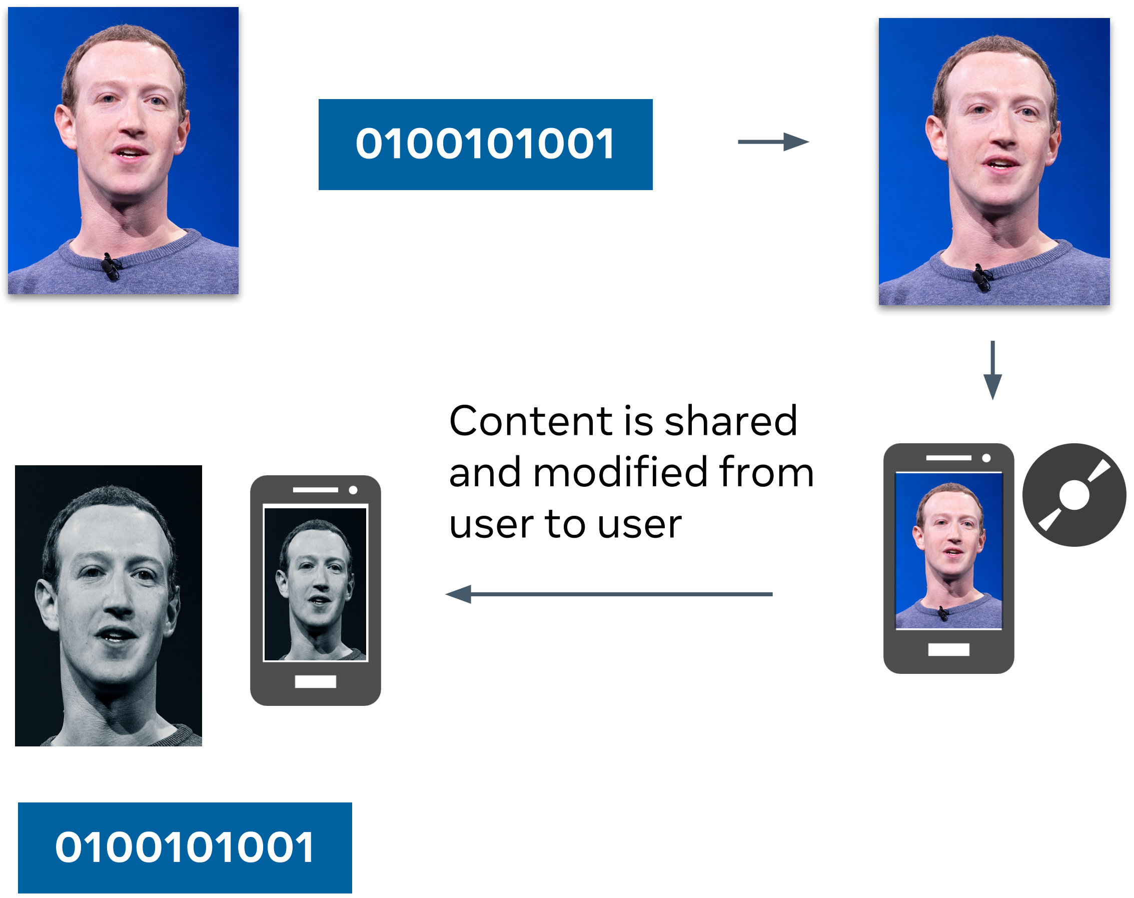
Motivation: Watermarking



Introduction

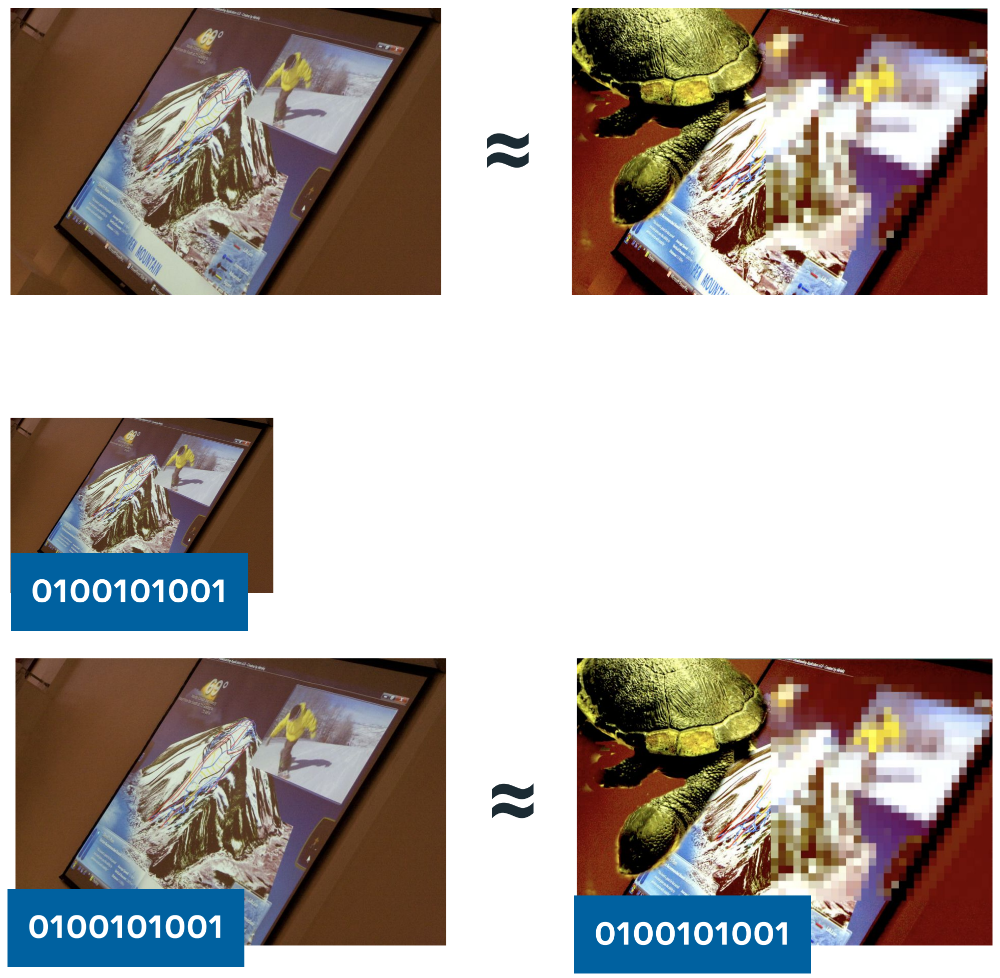
Motivation: Concrete Applications

- **Copyright Protection**



Mark remains unchanged → **content authenticity**

- **Content Tracing**



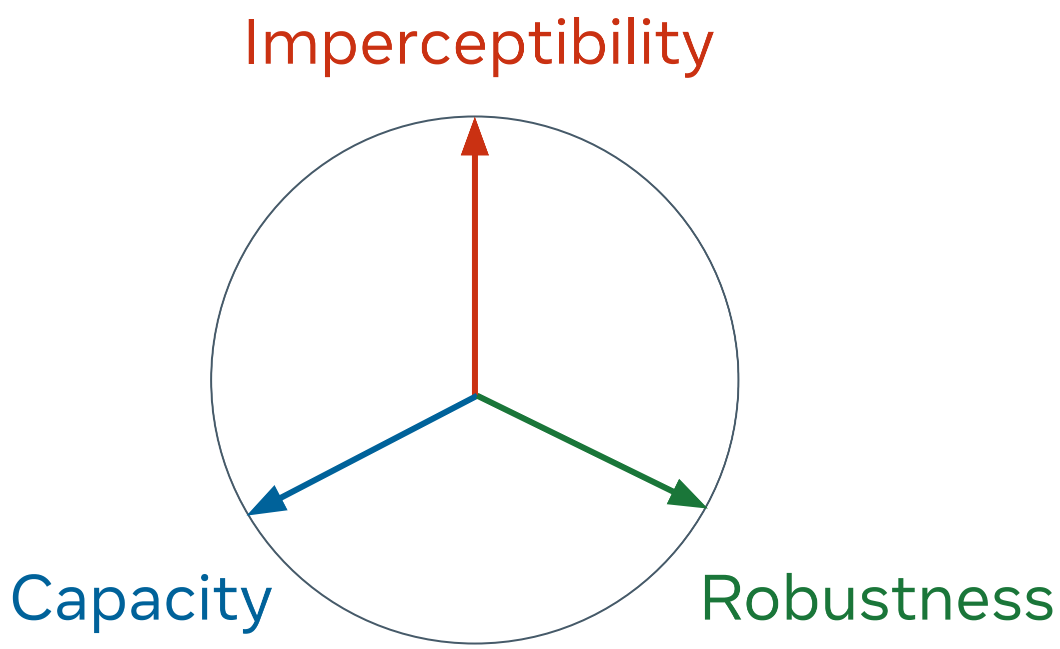
→ **Hard**

→ **Easier**

From: Matthijs Douze et al. 2021. "The 2021 Image Similarity Dataset and Challenge" In arXiv

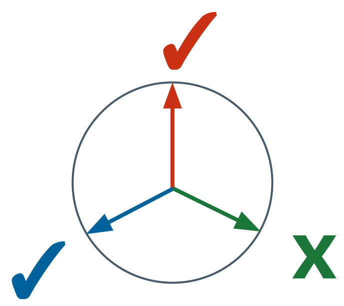
Watermarking: 3 Objectives

- 3 axes:

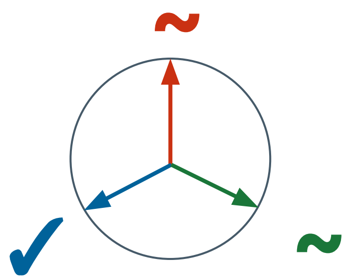


1. Imperceptibility
   Image distortion must be low
2. Capacity
   The message to hide can be long enough
3. Robustness
   The message must be recovered even if the image is transformed

- Existing approaches:
  - [📄 Cox et al. 2007] → **High capacity** but **robustness** specialized for **some transformations**



  - [📄 Luo et al. 2020] → **Deep Learning** methods specialized for robust watermarking
    Still **lack robustness + guarantees**



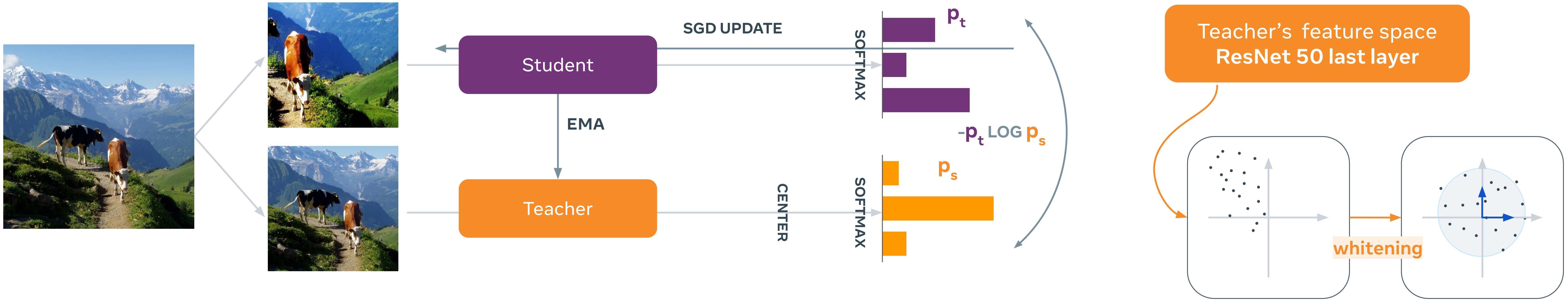
Self-Supervised Learning

- Motivation:
  - **use intrinsic robustness** of SSL neural networks to image transformations
  - does not suffer from **semantic collapse** of supervised learning (learns more than ImageNet classes only)

    → Latent space with more **capacity**

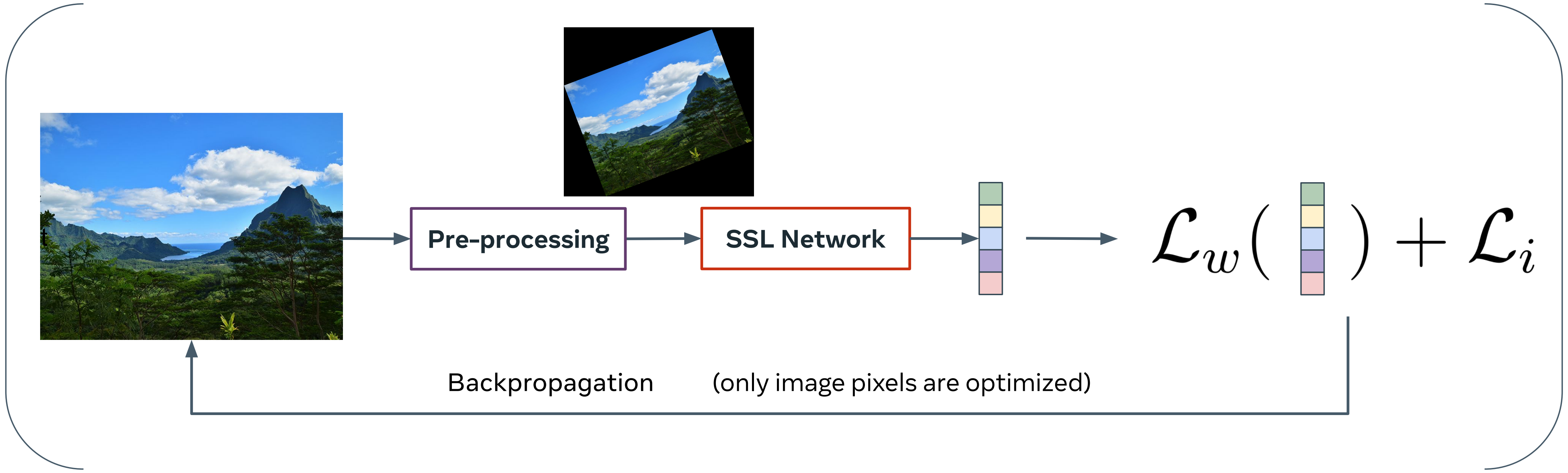
- SSL with DINO

  [📄 *Mathilde Caron et al. 2021, “Emerging Properties in Self-Supervised Vision Transformers.”*
]

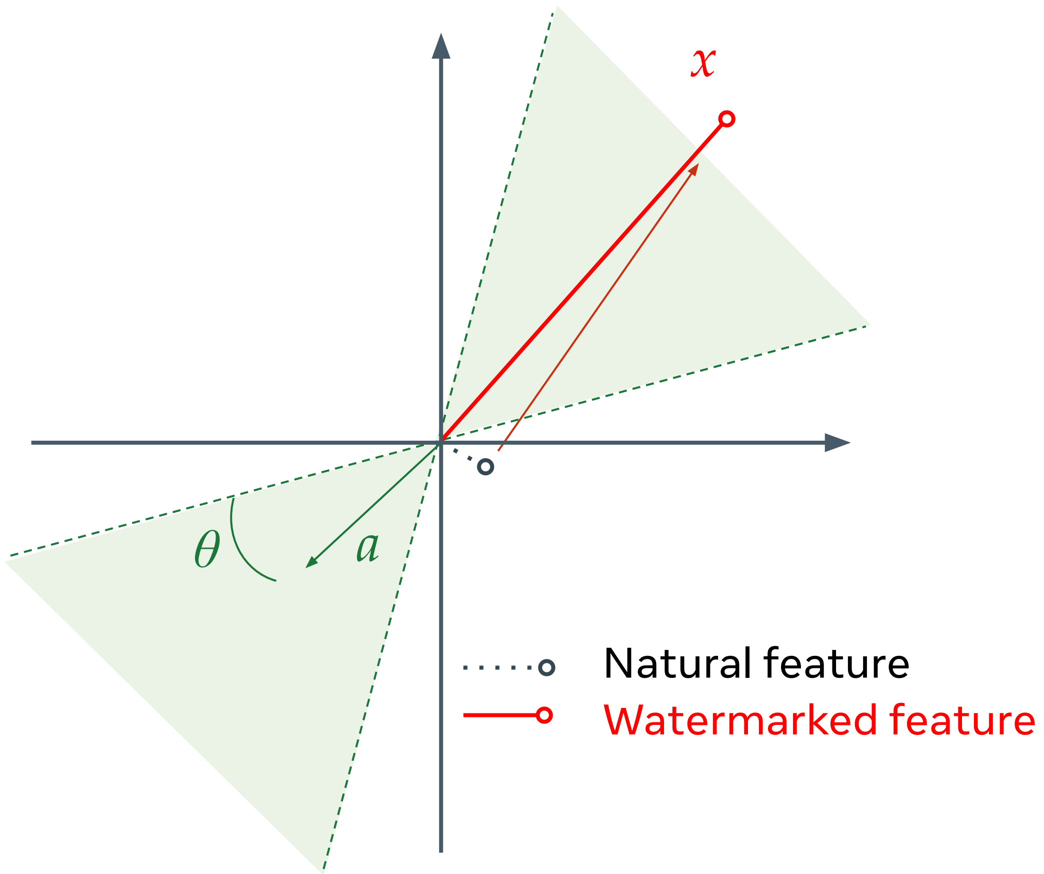


Watermarking in the Latent Space of a Network

- Method:
  - Mark in the **latent space** of a **self-supervised neural network**
  - **Simulate transformation** at marking time in the **pre-processing** module



several iterations



Example of 0-bit encoding with our method

Pre-processing → SSL filter + PSNR clipping
→ Data augmentation

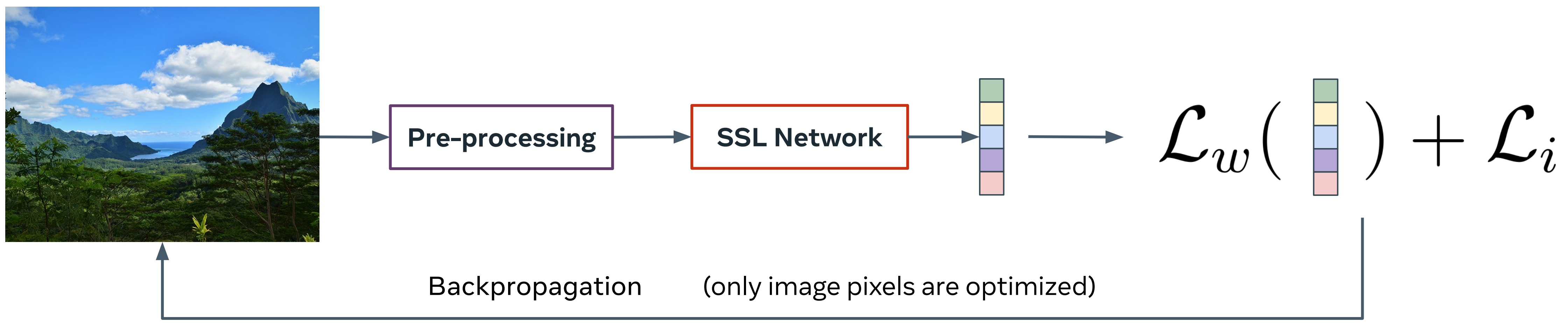
$$-\mathcal{L}_w(x) = (x^\top a)^2 - \|x\|^2 \cos^2 \theta$$

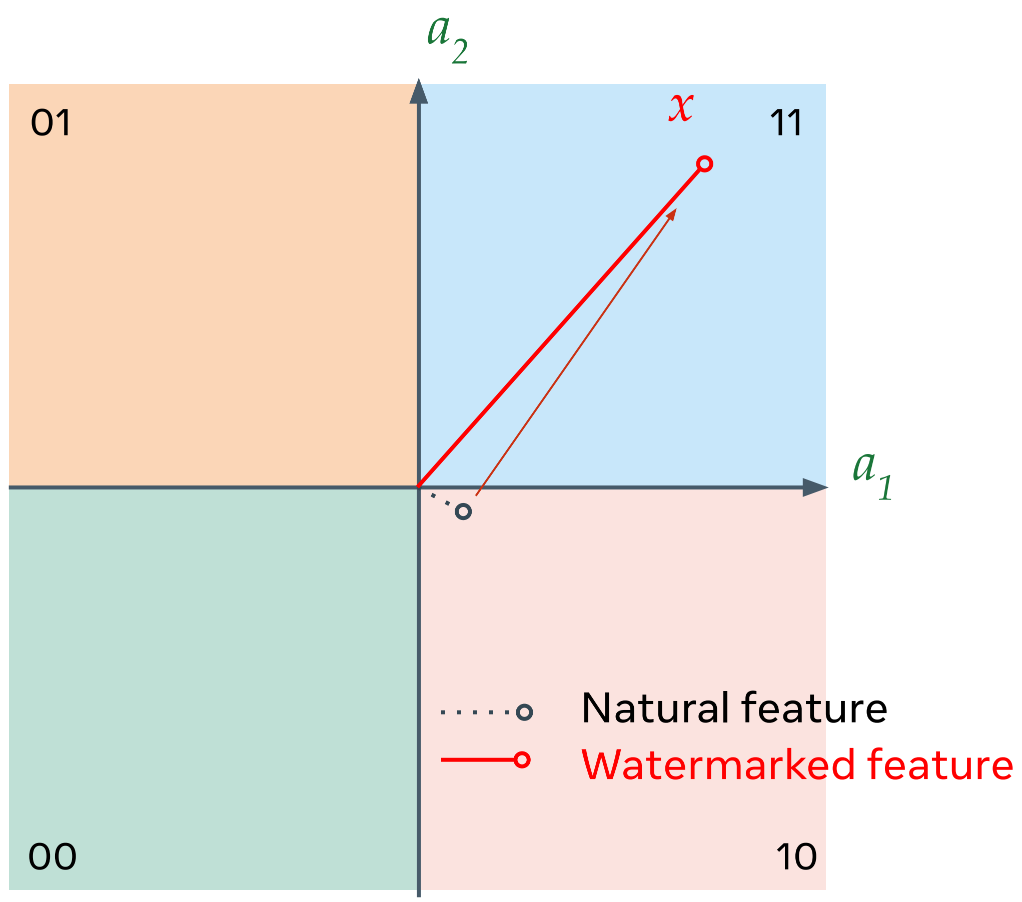
SSL Network → **Fixed**

Method baked on [Vukotić et al.. 2020 "Are Classification Deep Neural Networks Good for Blind Image Watermarking?"]

Watermarking in the Latent Space of a Network

- Method:
  - Mark in the **latent space** of a **self-supervised neural network**
  - **Simulate transformation** at marking time in the **pre-processing** module





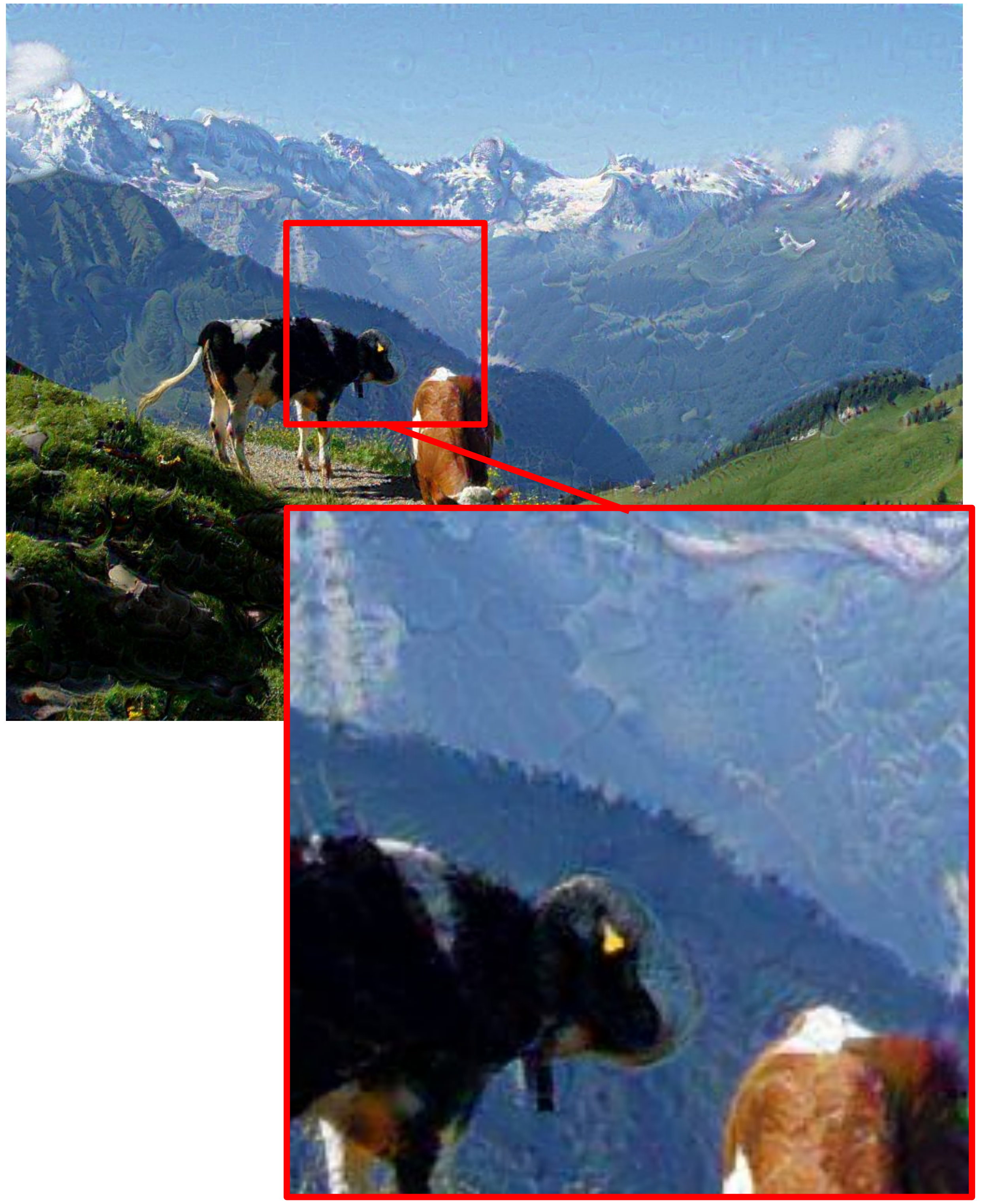
Example of 2-bits encoding with our method

$$\mathcal{L}_w(x) = \frac{1}{k}\sum_{i=1}^{k} \max\left(0, \mu - (x^\top a_i).m_i\right)$$

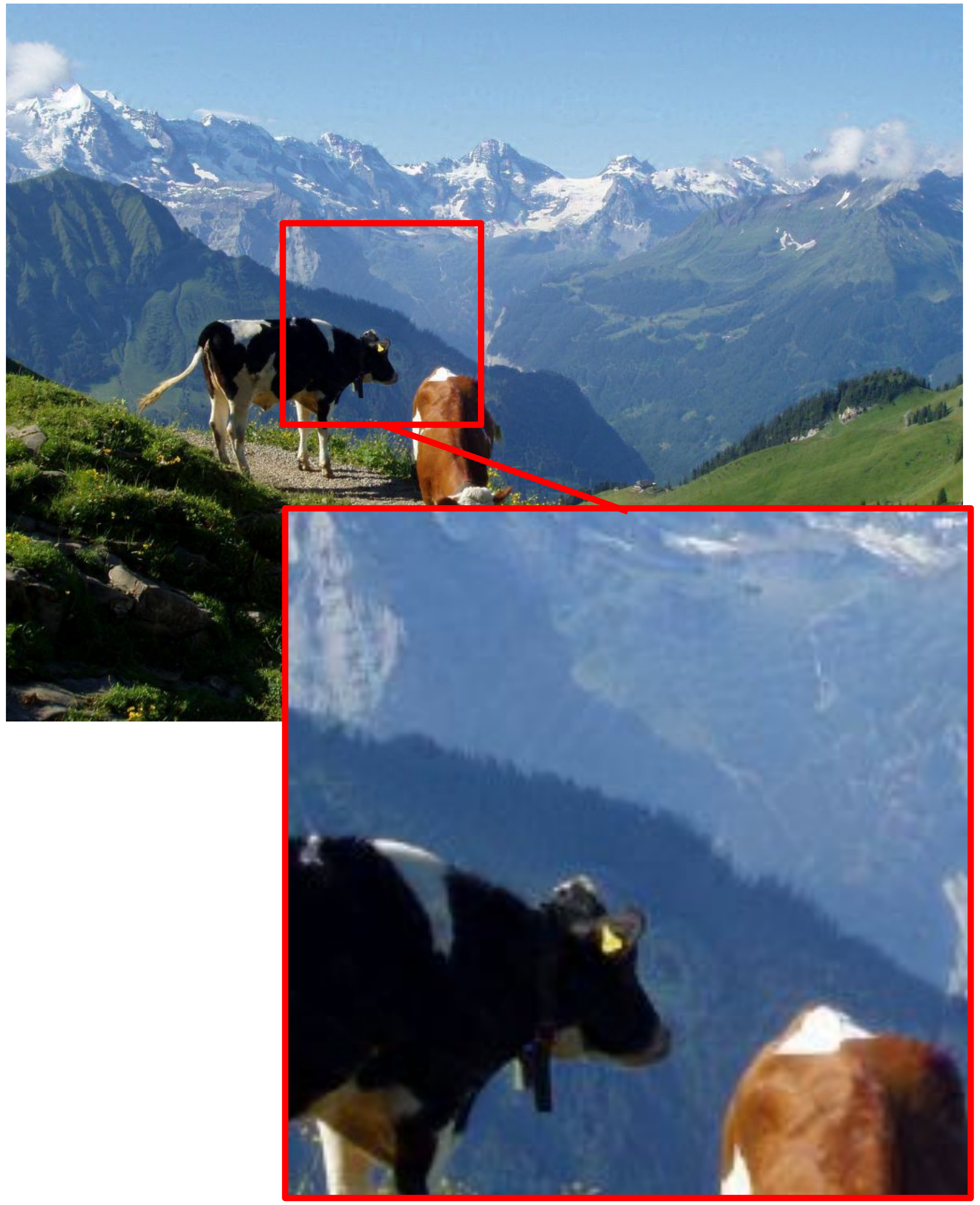
Method baked on [Vukotić et al.. 2020 "Are Classification Deep Neural Networks Good for Blind Image Watermarking?"]

Examples

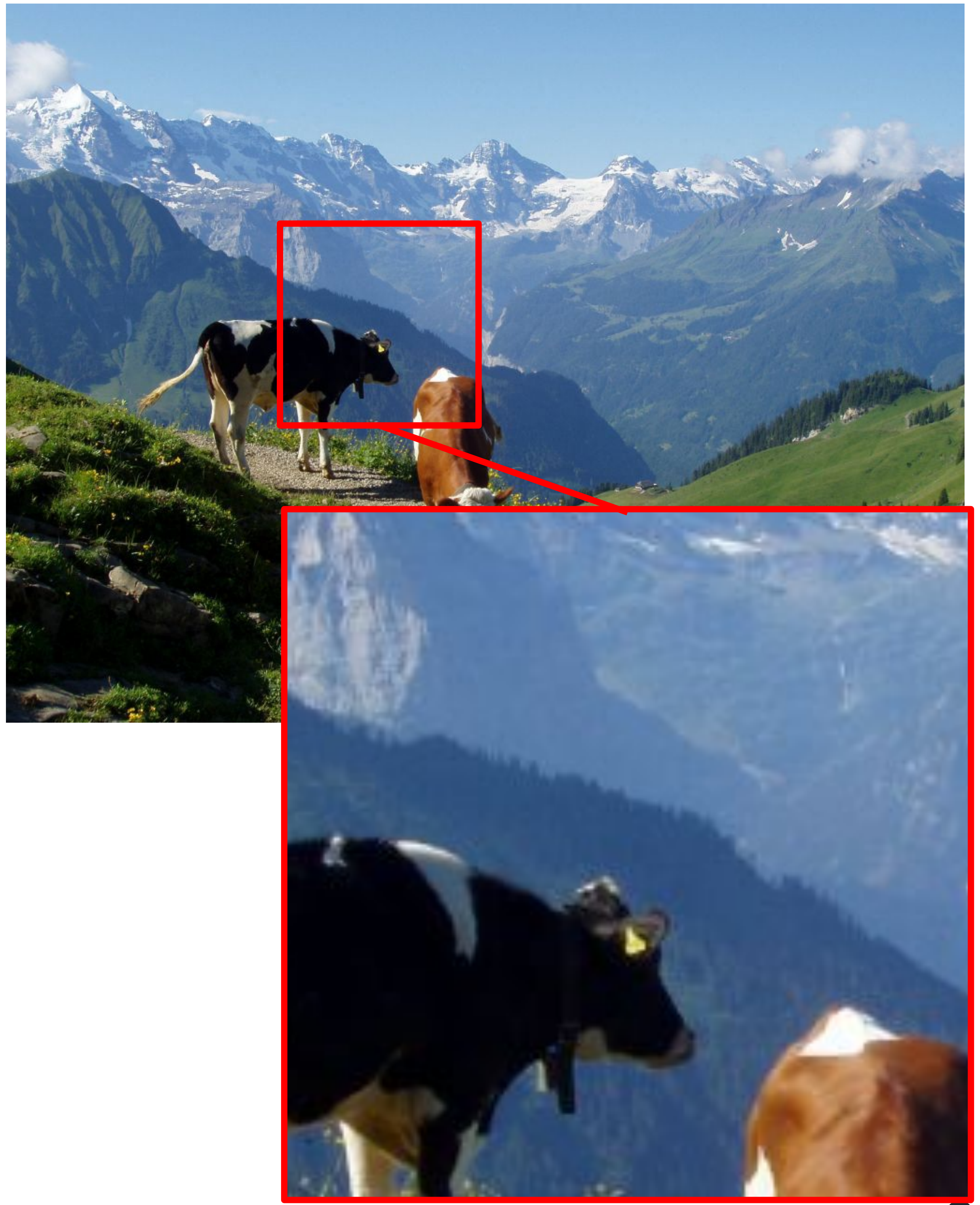
PSNR=28db



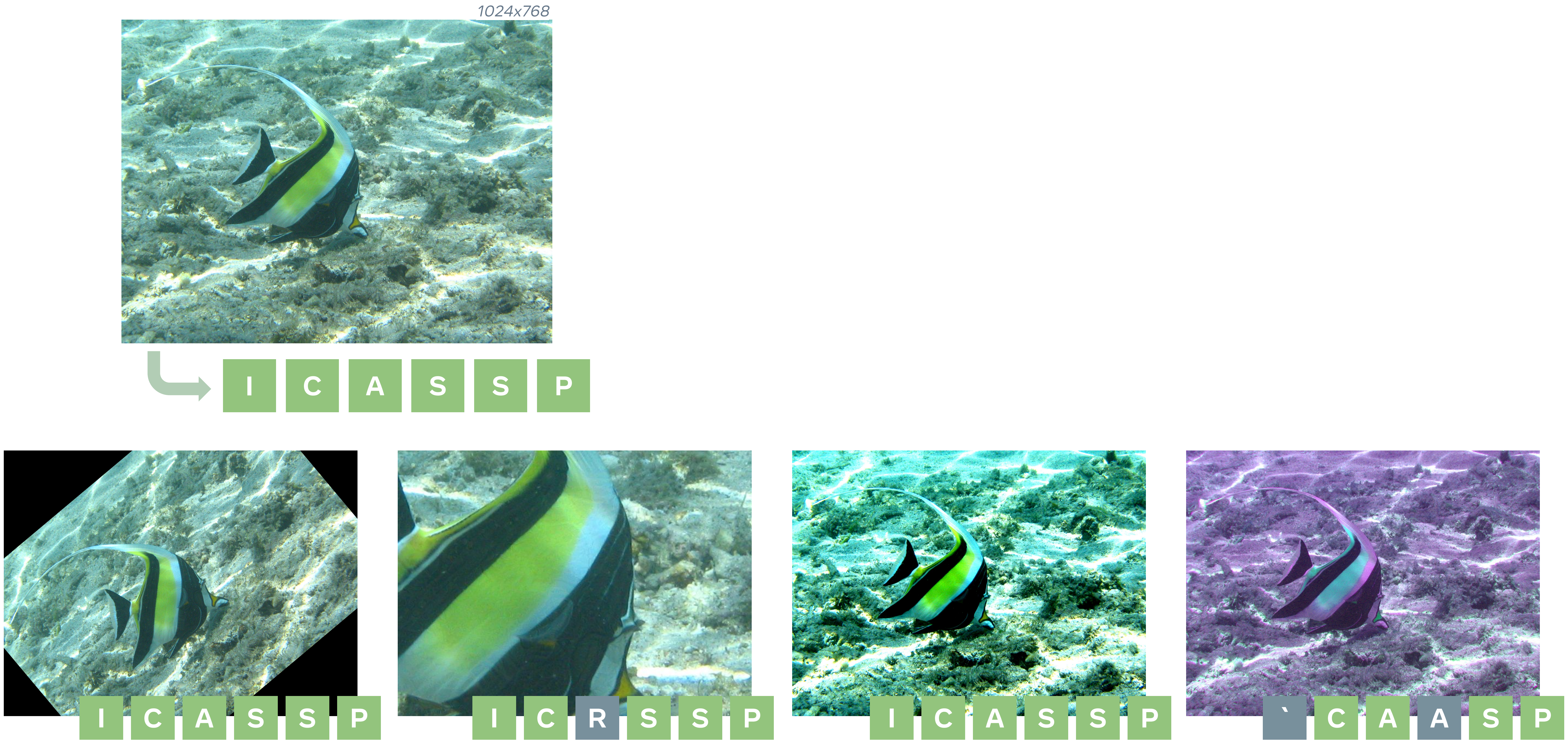
PSNR=40db



PSNR=52dB



Examples



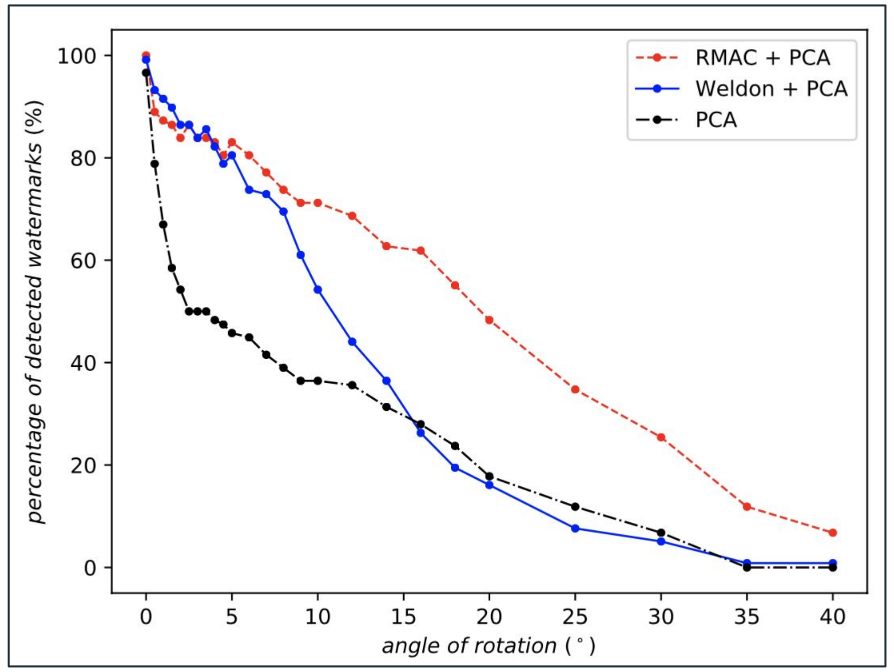
Influence of SSL and Data-augmentation

- **0-bit -** example of the **robustness to rotation**

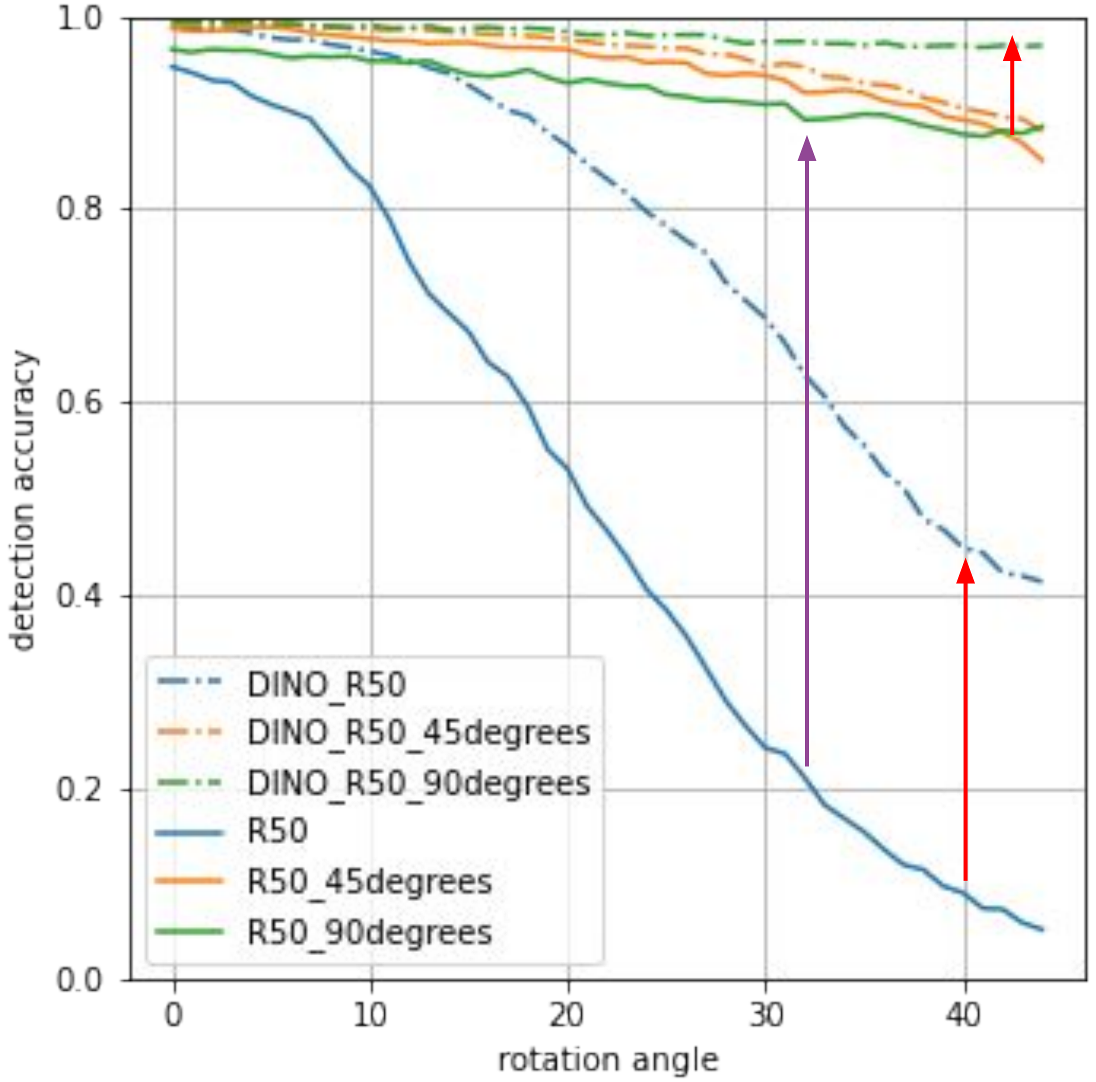
(i): Data augmentation both at network's training and marking time

(ii): SSL → better semantic space

Setup: 1k images from YFCC, PSNR=42dB, FPR=10^{-6}



True positive rate of the detection against the angle applied before detection



Previous work from:

[Vukotić et al.. 2020 "Are Classification Deep Neural Networks Good for Blind Image Watermarking?"]

VS state-of-the-art

- **0-bit**

True positive rate for different attacks on the watermarked images:

Transformation	Id.	Rot. (25)	Crop (0.5)	Crop (0.1)	Resize (0.7)	Blur (2.0)	JPEG (50)	Bright. (2.0)	Contr. (2.0)	Hue (0.25)	Meme	Screen
Ours	**1**	**1**	**1**	**0.98**	**1**	**1**	**0.97**	**0.96**	**1**	**1**	**1**	**0.97**
Vukotic et al.	**1**	≈ 0.3	≈ 0.1	≈ 0.0	-	-	**≈ 1.0**	-	-	-	-	-
Vukotic et al. (our implementation)	**1**	0.27	1	0.02	1	0.25	**0.96**	**0.99**	**1**	**1**	0.98	0.86

Setup: 118 images from CLIC, PSNR=42dB, FPR=10^{-3}

VS state-of-the-art

- **Multi-bit**

Bit accuracies in % for different attacks on the watermarked images:
(50% is no better than chance)

Attack	None	JPEG (Q=50)	Blur (σ=1)	Resize (0.7)	Crop (0.1)	Hue (0.2)
Ours on COCO - not resized	100	96	100	100	82	97
Ours on COCO resized to 128x128	**100**	**85**	**99**	84	45	**95**
HiDDeN - Zhu et al. 2018 on COCO resized to 128x128	**100**	77	**99**	85	**100**	75
Dist. Agnostic - Luo et al. 2020 on COCO resized to 128x128	**100**	82	93	**88**	98	94

Setup: 1k images from COCO, PSNR=33dB, K_{bits}=30

Key takeaways and future work

- **Self-supervised embedding** spaces are **excellent** **[DINO]**
- **Data augmentation** at **training** <u>AND</u> **marking** time
- **Significant improvement** over state-of-the-art on **0-bit watermarking**
  Multi-bit extension: **on-par** with SOTA

Future work:

- Watermarking in **forward** pass only
- **Specific** training for watermarking