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

Summary

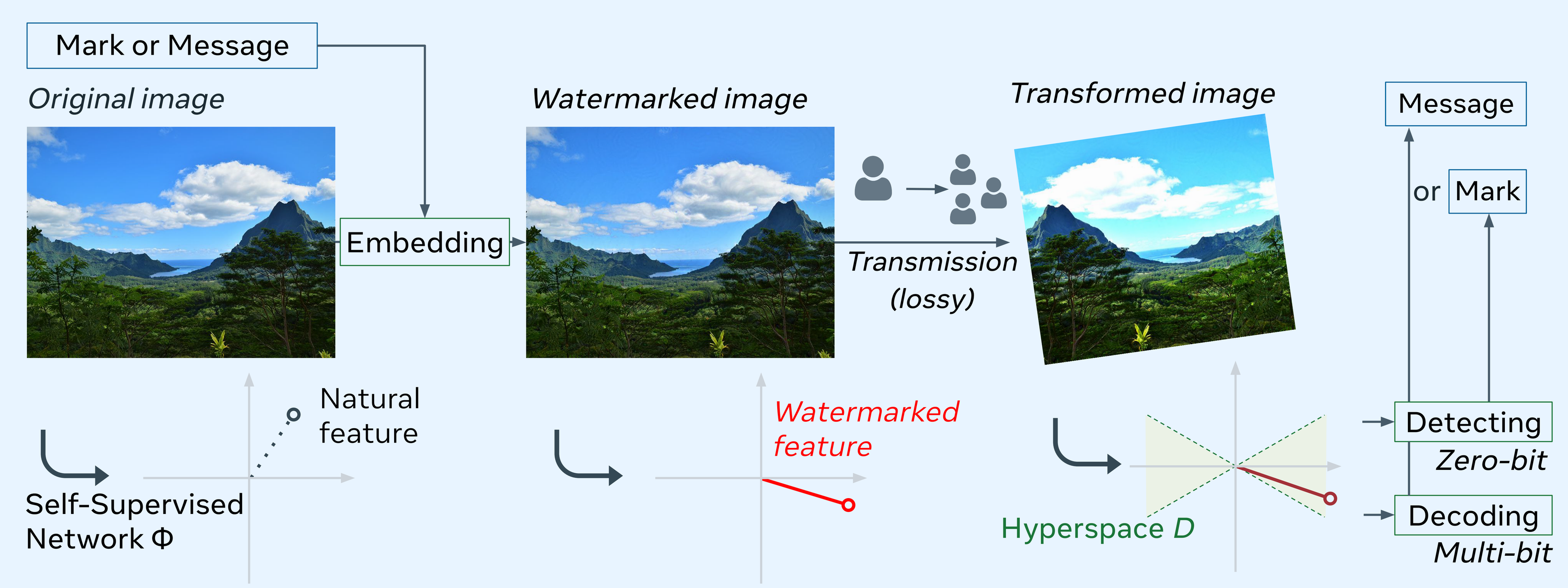
Context: Robust watermarking and data hiding

- Trade-offs: imperceptibility, payload, robustness
- Deep watermarking architectures require heavy training & lack robustness

Our contributions:

- Encode marks or binary messages in the latent space of any pre-trained network
- Leverage data augmentation at marking time
- Self-supervision → excellent embedding spaces

Method overview



The method is made of:

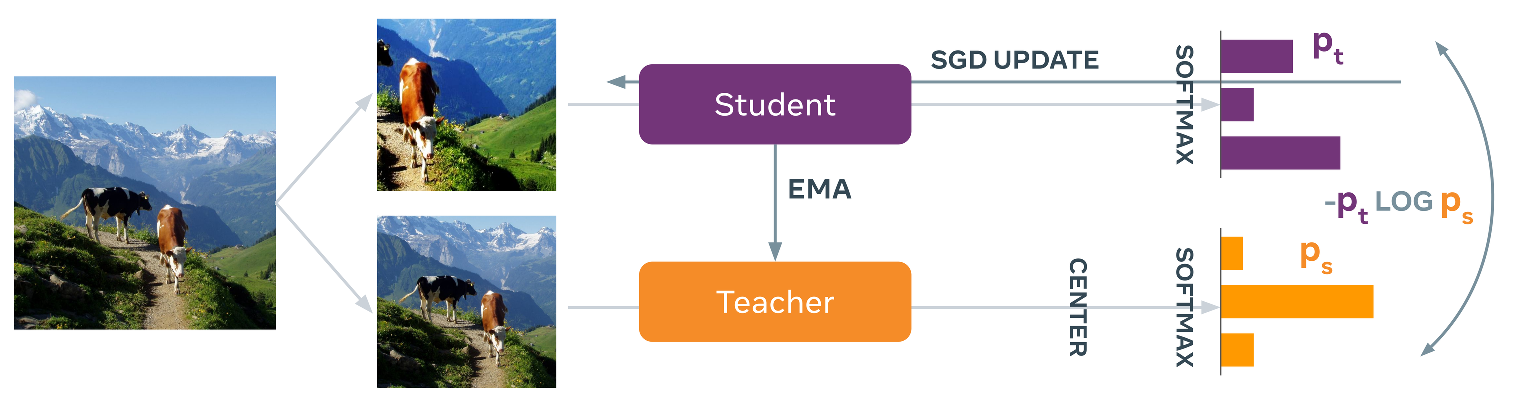
- A neural network trained with self-supervision that extracts features from images
- An embedding process that shifts the features into a well-specified region of the latent space
- A decoding step that happens in the same latent space

Feature extraction

Self-Supervised Pre-Training

Teacher-Student approach with DINO [1]:

- different augmented views of the same image, stronger for student than teacher
- pretext task: match output of student and teacher



Motivations behind the use of SSL

+ leverage inherent robustness to data augmentations.
+ SSL is fine grained (captures more than classes only) and does not suffer from the semantic collapse that happens because of supervised learning
  → latent space with *more bandwidth*.

Latent space normalization with whitening

Features output by the neural network are not well distributed.
→ Apply PCA whitening transformation *at marking time* for the features to have zero mean and identity covariance.

Qualitative results

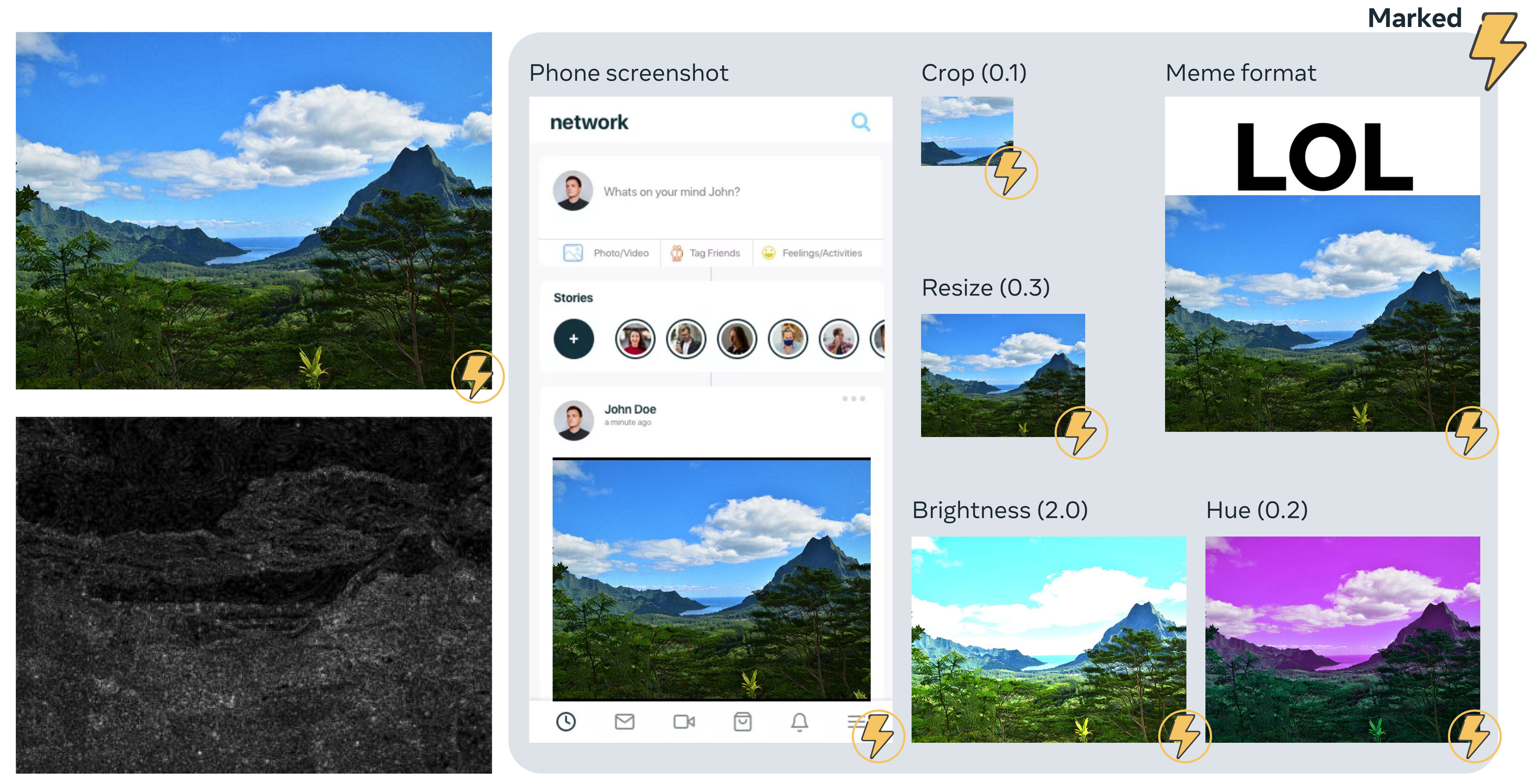


Figure 1. Image (800x600) watermarked at PSNR=40 dB and FPR=10^{-6}, and some attacked versions of the image, where the mark is detected by the hypercone detector

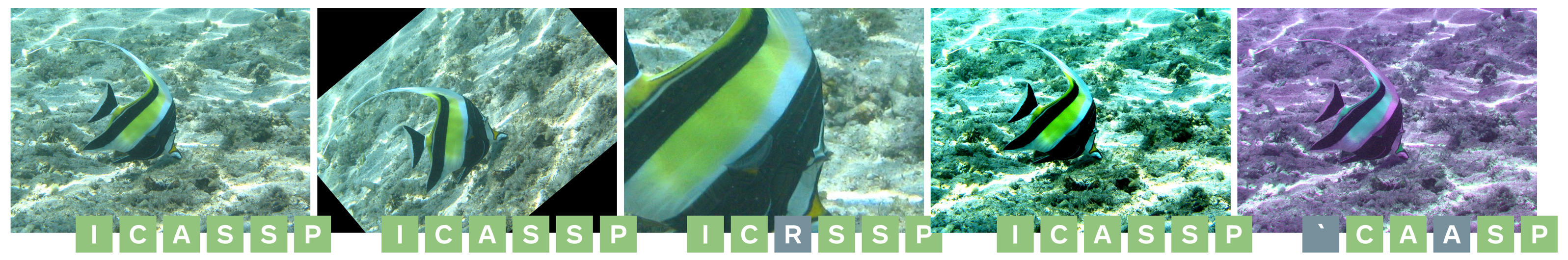


Figure 2. Image (1024x768) watermarked at PSNR=40 dB and a payload of 30 bits, and decoded messages

Embedding process

Goal: take image I_0 and output visually similar I_w carrying the mark/message.
Gradient descent over image pixels:

Algorithm One iteration of the embedding algorithm

Impose perceptual constraints (SSIM and PSNR filters) $\triangleright\ I_w \xleftarrow{\text{constraints}} I_w$
Sample data-augmentation and apply it to the image $\triangleright\ I_w \leftarrow \mathrm{Tr}(I_w, t)\ ;\ t \sim \mathcal{T}$
Compute loss (ϕ is the feature extractor) $\triangleright\ \mathcal{L} \leftarrow \lambda \mathcal{L}_w(\phi(I_w)) + \|I_w - I_0\|$
Update the image with GD $\triangleright\ I_w \leftarrow I_w + \eta \times \mathrm{Adam}(\mathcal{L})$

Hypercone detector

Secret key $a \in \mathcal{F}$; $\|a\| = 1$, dual hypercone: $\mathcal{D} := \left\{ x \in \mathbb{R}^d : \|x^T a\| > \|x\| \cos(\theta) \right\}$

Objective function: "how far the feature lies from the hypercone"

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

Theoretical guarantees on the False Positive Rate (FPR):

$$\mathrm{FPR} := \mathbb{P}\left(\phi(I) \in \mathcal{D} \mid \text{"key } a \text{ is uniformly distributed"}\right) = 1 - I_{\cos^2(\theta)}\left(\frac{1}{2}, \frac{d-1}{2}\right)$$

Hyperspace decoding

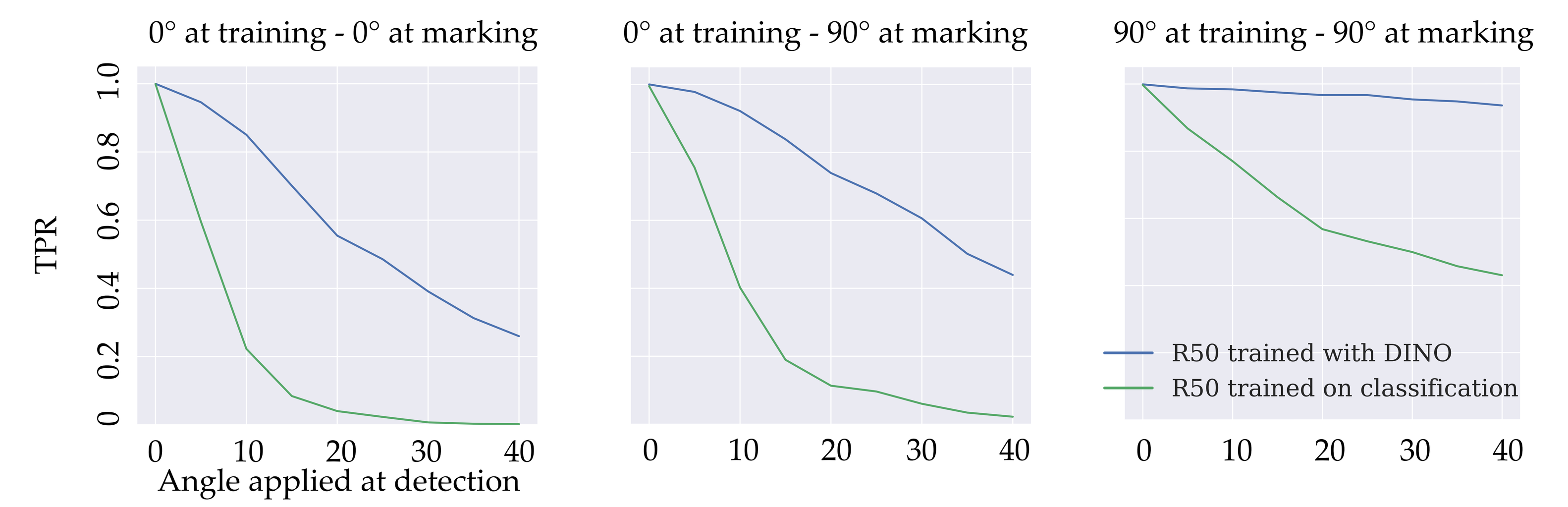
Secret key: randomly sampled orthogonal family of carriers $a_1, ..., a_k \in \mathbb{R}^d$.
Modulation of message $m = (m_1, ..., m_k) \in \{-1, 1\}^k$ into the signs of the projection of the feature $\phi(I)$ against each of the carriers.
Decoded message:

$$\hat{m} = D(I) = \left[\mathrm{sign}\left(\phi(I)^\top a_1\right), ..., \mathrm{sign}\left(\phi(I)^\top a_k\right) \right].$$

Objective function: hinge loss with margin $\mu \geq 0$ on the projections

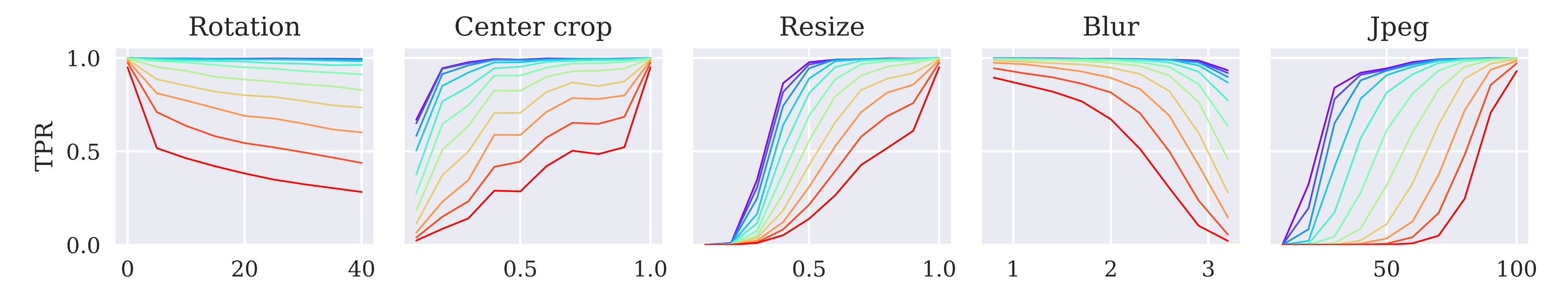
$$\mathcal{L}_w(x) = \frac{1}{k} \sum_{i=1}^{k} \max\left(0, \mu - (x^\top a_i).m_i\right).$$

Impact of SSL and data augmentation



True Positive Rate (TPR) of detection on 1k images from YFCC, at PSNR= 40dB and FPR= 10^{-6}, against different rotation angles.
→ SSL alone greatly improves watermarks' robustness against attacks.
→ Adding augmentation during both network's training and marking stages also does.

Trade-off on image quality



TPR of detection at FPR= 10^{-6} against different attacks.
PSNR ranging from 52 dB to 32 dB. Lower PSNR → more robustness.
Remarks: Similar trade-offs w.r.t. FPR and payload - Applies for multi-bit.

Our approach VS the state of the art

Zero-bit watermarking

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	Screenshot
Ours	1.00†	**1.00**†	1.00†	**0.98**†	1.00†	**1.00**†	0.97	0.96†	1.00†	1.00†	1.00	**0.97**
[3] (⋆)	1.0†	≈ 0.3†	≈ 0.1†	≈ 0.0†	-	-	≈ 1.0	-	-	-	-	-
[3] (⋆⋆)	1.00†	0.27†	1.00†	0.02†	1.00†	0.25	0.96	0.99	1.00	1.00	0.98	0.86

Table 1. (⋆) best results in [3], (⋆⋆) our implementation of [3]. † denotes augmentations used at pre-training.

TPR on 118 CLIC images, at PSNR≥ 42 and FPR= 10^{-6} → Noticeable improvement w.r.t. [3].

Multi-bit watermarking (data hiding)

Transformation	Identity	JPEG (50)	Blur (1.0)	Crop (0.1)	Resize (0.7)	Hue (0.2)
Ours	0.00‡	**0.15**	**0.01**‡	0.45‡	0.16‡	**0.05**
HiDDeN [4]	0.00‡	0.23‡	**0.01**‡	**0.00**‡	0.15	0.29
Dist. Agnostic [2]	0.00‡	0.18‡	0.07‡	0.02‡	**0.12**	0.06

Table 2. ‡ denotes transformations used in the embedding process.

Bit Error Rate (BER) on 1k COCO images resized to 128x128, at PSNR≥ 33, and with a payload of 30 bits. → Results comparable to [2, 4], better for JPEG (never seen at train nor at mark time).

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *ICCV*, 2021.

[2] Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. Distortion agnostic deep watermarking. In *CVPR*, 2020.

[3] Vedran Vukotić, Vivien Chappelier, and Teddy Furon. Are classification deep neural networks good for blind image watermarking? *Entropy*, 2020.

[4] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *ECCV*, 2018.