https://doi.org/10.2352/EI.2024.36.4.MWSF-329

DeepSync: Affine Transform Recovery via Convolutional Neural Networks for Watermark Synchronization

Dimitris G. Chachlakis, Mohamed Yousuf, and Tomáš Filler, Digimarc Corporation, 8500 SW Creekside Pl, Beaverton, OR 97008, USA

Abstract

In the context of digital watermarking of images/video, template based techniques rely on the insertion of a signal template to aid recovery of the watermark after transforms (rotation, scale, translation, aspect-ratio) common in imaging workflows. Detection approaches for such techniques often rely on known signal properties when performing geometry estimation before watermark extraction. In deep watermarking, i.e., watermarking employing deep learning, focus so far has been on extraction methods that are invariant to geometric transforms. This results in a gap in precise geometry recovery and synchronization which compromises watermark recovery, including the recovery of information bits, i.e., the payload. In this work, we propose DeepSync, a novel deep learning approach aimed at enhancing watermark synchronization for both template-based and deep watermarks.

Introduction

Digital Watermarking (DW) is a field with a rich historical backdrop. DW systems deployed at scale are built on core signal processing and information theory principles which underline the communication systems we rely on today (e.g., satellite communications, cellular networks, etc.) [1]. Successful DW applications include copyright protection, content authentication, digital asset management, document security, recycling, factory automation, and consumer engagement, to name a few.

DW technology that can withstand geometric transforms, digital-to-analog and analog-to-digital conversions, compression, and noise [2] has proven to be useful in typical imaging and print workflows, as well as streaming and embedded applications. Besides the reliability of watermark extraction, this proven effectiveness is driven by extremely low false positive rates, highlighting DW's reliability and performance in diverse operational contexts. Moreover, the technology has a long-standing track record in large scale deployments including commodity embedded systems with minimal cost or impact to speed/throughput.

A successful watermarking framework is defined by its ability to strike a delicate balance between three key factors. (i) *Imperceptibility*: The watermarking process should embed a payload into an image while causing minimal perceptual changes to the original content. Maintaining perceptual quality ensures that the watermarked image remains visually indistinguishable from the original, enabling user acceptance and satisfaction. (ii) *Robustness*: In presence of image distortions, compression, or other common forms of image workflows/attacks, including compression, the watermark should exhibit resilience. The payload should be recoverable even after these manipulations, allowing for reliable information retrieval in real-world scenarios. (iii) *Capacity*: Payload capacity is the maximum amount of information bits that can be embedded and extracted for a given DW technique, without introducing artifacts or compromising image integrity. Payload capacity facilitates various applications such as data hiding, annotation, and content identification/provenance [3].

Various DW techniques have been developed and deployed at scale that strike a balance between these key factors. For example, frequency domain methods such as Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) manipulate the frequency components of digital content (can be image or video) to embed watermarks discreetly. Spread spectrum watermarking disperses watermark data (payload) across digital content, enhancing resilience against various attacks [1]. Template matching, i.e., a popular watermarking method that involves overlaying a predefined pattern or template onto digital content for watermark synchronization, can be combined with any of the above methods. The template can be composed of either a fixed or payload variable signal with known properties. Advanced methods may use a combination of the above.

The advent of Generative Artificial Intelligence (GenAI) has accelerated interest in DW technology, now seen as a key tenet for communicating content provenance [3]. This resurgence is further enhanced by government endorsements of DW, including a recent White House executive order and other global legislative efforts [4]. Consequently, research on the applicability of DW to GenAI has increased with an emphasis of using AI-based tools for watermarking [5–7]. The focus on AI-based watermarking has give rise to "deep watermarking", a domain that merges deep learning innovations with DW. In such deep watermarking approaches, the focus so far has largely been on watermark extraction methods that are invariant to geometric transformations, revealing a gap in precise geometry recovery and synchronization, in contrast to the aforementioned DW approaches.

In this work, we propose DeepSync, a novel deep learning approach enabling watermark synchronization applicable for both template-based and deep watermarks.

Deep Watermarking Background

HiDDeN [8] discusses an approach to deep watermarking of images by utilizing CNNs for both embedding and extracting payloads. A discriminator/adversary CNN was used for inducing watermark imperceptibility. Robustness was induced by explicitly modeling a limited set of digital distortions at training time. HiDDeN illustrated the potential of deep learning for creating robust and imperceptible watermarks capable of surviving digital image distortions, including compression. Unlike HiDDeN, distortion agnostic deep watermarking [9] discusses a deep learning framework for embedding watermarks in images which combines adversarial training and channel coding to attempt to generalize

better to unknown distortions. ReSWAT [10] focuses on signal provenance across various signal types (e.g., image, video, audio). This method embeds watermarks into media by means of gradient descent optimization over the cover media and utilizes a CNN for detection. SSL [11] explores watermarking in images by utilizing image latent spaces produced by pre-trained self-supervised networks. The embedding of either zero-bit or multi-bit payloads relies on gradient descent optimization in the latent space of the image. Extraction relies on the latent space of the watermarked image. Stable Signature [12] discusses a method for embedding watermarks directly into the generation process of Latent Diffusion Models (LDMs), with the goal that generated images carry an invisible signature for later detection or identification. LECA [13] discusses a deep image watermarking technique which estimates and inverts a selection of geometric transforms before extraction. A light-weight encoder produces a mask-pattern which is tiled and alpha-blended with the cover image. The synchronization network produces a signal pattern which is used to estimate scale and translation by means of an exhaustive pattern-matching search. The decoder network extracts the payload after inversion of scale and translation. RoSteALS [14] discusses a data hiding approach which takes advantage of pre-trained autoencoders for robust payload embedding within images. This approach reduces training and exhibits high performance with respect to payload recovery and image quality. More recently, TrustMark [15] describes a GAN-based watermarking method designed for images of arbitrary resolution. This approach balances watermark imperceptibility and recovery accuracy through architectural designs and loss functions, enhancing robustness against various digital image distortions.

A noticeable pattern emerges across the above deep watermarking methods. All approaches, with the exception of LECA, extract watermarks without inverting geometric transforms, most paying little attention to image rotation, thus highlighting a prevalent trend towards transform-invariant extraction. LECA recovers geometric transforms by means of an exhaustive pattern-matching search where a universal template derived from periodic signals is matched to a template output of a synchronization network. In contrast, our proposed approach does not necessarily rely on an inserted template signal.

Geometric Transform Invariance Cost

Attempting to learn geometric transform invariance, rather than explicitly recovering the geometry, has become the trend in deep watermarking systems as AI model training can incorporate required imaging transforms. However, this trend is not without its trade-offs. In this section, we probe the implicit costs associated with learning transform invariance. First, we define payload capacity as the number of bits a model can convey while recovering payload bits with Bit Error Rate (BER) less than 2%. Then, we consider rotation and/or scale transforms without any additional noise/blur and strive to evaluate payload capacity as a function of transform severity. For our evaluation, we use HiDDeN [8] at a reduced complexity level of 32 channels per convolutional layer to streamline the numerical experiments, allowing for more efficient computation. Rotation severity varies in the range of interest $(\phi_{\min}, \phi_{\max})$ by varying $\phi_{\min} = -\phi_{\max} \in \{0, 30, 60, 90, 180\}^\circ$. The scale, when applied, is chosen at random from range of $(0.5, 2)$. For every combination

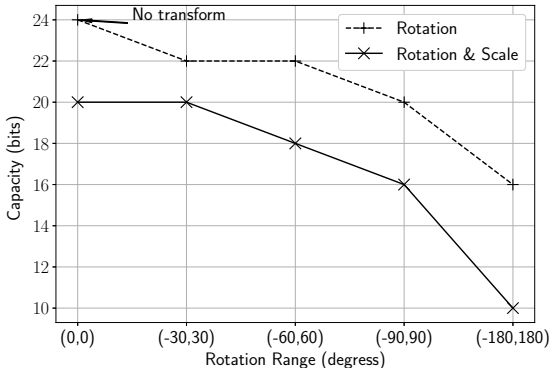

Figure 1. *Reduced complexity HiDDeN payload capacity vs transform severity.*

of rotation and/or scale transforms, we train variants of HiDDeN on COCO2017 [16] for increasing payload lengths until we identify the maximum payload length for which BER $\leq 2\%$ on the validation set. We report our findings in Figure 1. We observe that as severity of transformation increases, payload capacity decreases; the relationship between transform severity and payload capacity is inversely proportional and non-linear.

Problem Statement

Watermarking systems designed for commercial applications need to survive geometric transforms (e.g., rotation, scale, crop) as such transforms form the foundation of existing imaging and print workflows. Fragility of DW would render large-scale systems unreliable. This has been accomplished by explicit estimation and inversion of the transforms–i.e., watermark synchronization. In contrast, deep watermarking systems attempt to learn geometric transform invariance during training. Empirically, learning transform invariance comes at a cost with respect to payload capacity and/or watermark imperceptibility (see Figure 1). The explicit estimation and inversion of geometric transforms remains largely unexplored in deep watermarking literature. In this work, we strive to bridge this gap. We propose DeepSync, a new deep learning approach for watermark synchronization applicable to watermark signaling methods, including content adaptive methods, independent of whether they employ templates or not.

Proposed DeepSync Approach

To lay the groundwork for explaining DeepSync, we commence with an overview of the overall DW workflow within which it functions. We consider availability of generic watermark embedding and extraction processes $\mathscr{W}_{\text{emb}}$ and $\mathscr{W}_{\text{ext}}$, respectively. Process $\mathscr{W}_{\text{emb}}$ embeds a length-L binary sequence of bits (i.e., the payload) $\mathbf{p}$ to an RGB host image $\mathbf{X}$ producing the watermarked image $\mathbf{Y}$. The marked image experiences geometric transforms (e.g., rotation, scale, translation, crop) resulting in the distorted image $\widetilde{\mathbf{Y}}$. By means of supervised learning, DeepSync learns to estimate the transforms experienced by the watermark signal. These estimates are used to invert the geometric transforms (i.e., watermark synchronization) producing the inverse transformed image $\widehat{\mathbf{Y}}$. Finally, the inverse transformed and synchronized image $\widehat{\mathbf{Y}}$ is given as input to the watermark extractor $\mathscr{W}_{\text{ext}}$ which, in

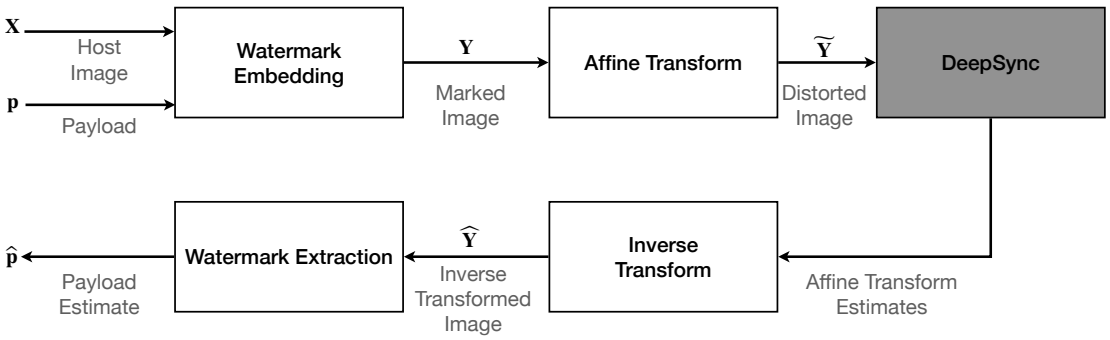

Figure 2. *The overall watermarking workflow in which DeepSync functions.*

turn, outputs the payload estimate $\widehat{\mathbf{p}}$. A visual illustration of this watermarking workflow is offered in Figure 2. A detailed description of DeepSync follows.

Motivation

For simplicity in presentation, we consider that the transform of interest is rotation in the range

$$\Phi = [\phi_{\min},\ \phi_{\max}). \tag{1}$$

At its core, estimating the rotation experienced by the watermark signal is a regression problem. That is, considering that a watermarked image experienced rotation by angle $\phi \in \Phi$, a straightforward approach would be to extract features (e.g., via an AI backbone) based on which an estimate $\widehat{\phi}$ would be computed. We empirically observed that such a straightforward approach exhibits non-satisfactory performance. Consequently, we transform the regression problem into combined classification and regression by following a *range splitting* approach. Without loss of generality, Φ can be expressed as the union of $N \geq 1$ smaller ranges each of width

$$w = \frac{\phi_{\max} - \phi_{\min}}{N}, \tag{2}$$

where N is a user-configurable parameter. These smaller ranges are commonly referred to as "bins" or "intervals". For simplicity, we will henceforth refer to these ranges as bins. Mathematically, the range of interest Φ is split into N bins as

$$\Phi = \bigcup_{n=1}^{N} \Phi_n. \tag{3}$$

Bins are mutually exclusive, that is, for positive integers $n \leq N$ and $k \leq N$ such that $n \neq k$, it holds

$$\Phi_n \cap \Phi_k = \emptyset. \tag{4}$$

For every $n \geq 1$, the bounds of the n-th bin are defined as a function of N (or, bin width) as

$$\Phi_n = \left[\phi_{\min}^{(n)},\ \phi_{\max}^{(n)}\right) = [\phi_{\min} + (n-1)w,\ \phi_{\min} + nw). \tag{5}$$

We are considering one regression head for every bin Φ_n, $1 \leq n \leq N$; i.e., the n-th regression head estimates angle $\widehat{\phi}_n$ such that

$$\phi_{\min}^{(n)} \leq \widehat{\phi}_n < \phi_{\max}^{(n)}. \tag{6}$$

Multiple angle estimate bins introduce ambiguity because there is no way to know which regression head's output to consider as the final estimate. This motivates the need for a classification head which addresses this ambiguity. Interestingly, for $N = 1$, the problem simplifies to standard regression. For $N \geq 2$, the problem transforms to classification followed by a refinement (regression) step.

DeepSync Architecture

Image samples are given as input to generic AI backbone (e.g., EfficientNet, ResNet, MobileNet, other) for feature extraction. In turn, the backbone returns a length-F feature vector $\mathbf{f}$ which is shared by three tasks of interest in a cascaded detector architecture.

First, features in $\mathbf{f}$ are given as input to a fully connected layer which utilizes a sigmoid activation function to output a probability p. This probability indicates presence or absence of the watermark signal

$$p \underset{\text{present}}{\overset{\text{absent}}{\lessgtr}} \tau, \tag{7}$$

where $\tau \in (0,1)$ is a user-defined threshold enabling a trade-off between true- and false-positive rates.

When presence of the watermark signal is detected, a second fully connected layer takes $\mathbf{f}$ as input and outputs N estimates converted to probabilities $p_1, p_2, \ldots, p_N$ by utilizing the softmax activation function. Identifying

$$n^\star = \underset{n \in \{1,2,\ldots,N\}}{\operatorname{argmax}}\ p_n \tag{8}$$

indicates that the output of the $n^\star$-th regression head should be considered as the final prediction.

Finally, $\mathbf{f}$ is given as input to a third fully-connected layer which outputs N estimates $r_1, r_2, \ldots, r_N$ which, in turn, are converted to angle estimates as follows. For every $n \geq 1$, r_n is converted to the angle estimate

$$\widehat{\phi}_n = \phi_{\min}^{(n)} + \sigma(r_n)w, \tag{9}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function. In practice, only $\widehat{\phi}_{n^\star}$ needs to be computed. A schematic illustration of the DeepSync architecture is offered in Figure 3. Notably, DeepSync is easily extensible to include other invertible transforms similar to rotation, such as scaling and translation.

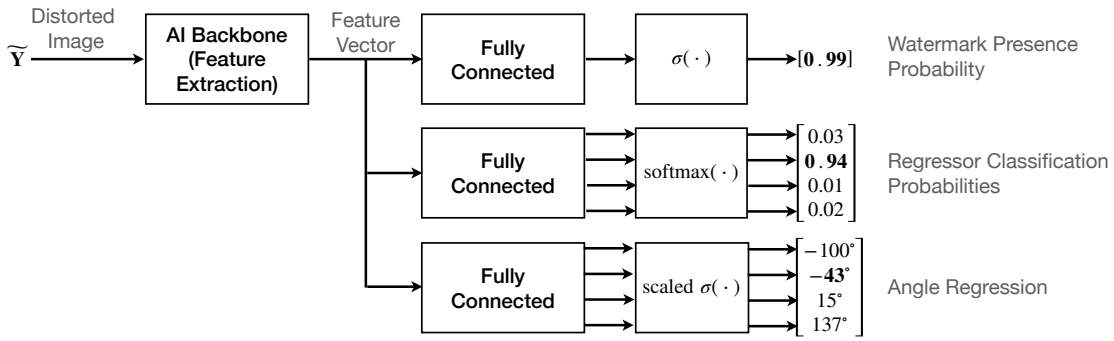

Figure 3. *DeepSync ($N = 4$) architecture for rotation only. Easily extensible to include other invertible transforms (e.g., scale and translation). Watermark presence is detected with* 0.99 *probability followed by selection of the 2nd regression bin leading to final angle estimate of* $-43°$.

Training Methodology

Every image is resized such that its minimum spatial dimension is of length 384. Then, a random crop of size 384-by-384 is extracted and undergoes a series of stochastic transformations that up-sample and rotate the image, in randomized order, resulting in the host image $\mathbf{X}$. This pre-processing step is designed to prevent DeepSync from relying on the natural orientation and scale of the host image [18], thereby encouraging the network to learn features associated with the watermark signal. Next, $\mathbf{X}$ is marked by means of $\mathscr{W}_{\text{enc}}$ with probability $\rho = 0.5$, producing $\mathbf{Y}$. When marked, $\mathbf{Y}$ is assigned label $y = 1$ indicating watermark presence, else $y = 0$. Thereafter, $\mathbf{Y}$ is rotated and resized by random angle $\phi \in \Phi$ and scale coefficient $s \in \mathscr{S} = [0.5, 2]$, respectively. A random crop of size 128-by-128 is extracted from the resulting image producing the distorted image $\widetilde{\mathbf{Y}}$, input to DeepSync. Angle ϕ and scale coefficient s serve as supervision signals for training the regression heads. Moreover, assuming N regression heads for rotation, by definition of $\{\Phi_n\}_{n=1}^{N}$, $\exists k \in \{1, 2, \ldots, N\}$ satisfying

$$\phi_{\min}^{(k)} \leq \phi < \phi_{\max}^{(k)}. \tag{10}$$

$\widetilde{\mathbf{Y}}$ is assigned to the k-th regression head. In addition, a length-N one-hot encoded vector $\mathbf{j}$–i.e., $[\mathbf{j}]_k = 1$ and $[\mathbf{j}]_n = 0$ for every $n \neq k$–serves as the supervision signal for training the classification head for rotation. Like rotation, we create one-hot encoded vector $\mathbf{h}$ for scale. We utilize the Binary Cross Entropy (BCE) loss for training the watermark presence/absence classifier. For rotation/scale regression, we utilize the Mean-Squared-Error (MSE) loss. Finally, we utilize the Cross Entropy (CE) loss for training the classifiers responsible for selecting the correct regression head. MSE and CE losses are only calculated for watermarked images.

Numerical Experiments

We evaluate the performance of DeepSync across an array of tasks of interest, including recovering geometric transforms, both with template-based and deep watermarks without explicit templates. All models in this Section are trained and tested on the COCO2017 [16] train and test sets which comprise $118,000$ and $41,000$ images, respectively.

Template-Based Signaling

The watermark template signal used in this investigation is a zero-mean spread-spectrum watermark as described in [17]. The process of watermark embedding combines an original RGB image with grayscale watermark tile uniformly scaled by a strength factor $\alpha \in (0.05, 0.5)$ along the luminance direction. More elaborate implementations may use visual masking models such as in [19]. We consider both rotation and scale transforms in $\Phi = [-180°, 180°)$ and $\mathscr{S} = [0.5, 2]$, respectively. Following the training methodology above, we train DeepSync with an EfficientNet_B0 as a backbone, N regression heads for rotation, $N \in \{40, 80, 120, 160\}$, and 10 regression heads for scale. As a benchmark, we train a model with one regression head for rotation and one regression head for scaling. This model attempts to solve the synchronization problem via regression directly.

Table 1: Template Watermark presence/absence classification.

Model	AUC
Regression (N=1)	0.998
DeepSync (N=40)	0.995
DeepSync (N=80)	0.996
DeepSync (N=120)	0.996
DeepSync (N=160)	0.995

We commence with a performance evaluation with respect to detecting watermark presence or absence by measuring the Area Under the Curve (AUC) metric. This metric offers a comprehensive assessment of classification performance by integrating the trade-off between true- and false-positive rates. We report AUC performance in Table 1. All models exhibit very high AUC performance with the benchmark model marginally outperforming the DeepSync models.

Next, we evaluate the rotation estimation performance by measuring the empirical Cumulative Distribution Function (CDF) of an estimators' errors. The CDF curves are illustrated in Figure 4. For every $N \geq 40$, DeepSync markedly outperforms the benchmark model which clearly illustrates the merit of range splitting. As N increases, the rotation estimation precision increases. For instance, for $N = 160$ and $N = 80$, about 79% and 70% of the samples, respectively, exhibit an error less than $|2|°$.

Lastly, like rotation, we measure the distribution of absolute errors for scale estimation by utilizing the empirical CDF metric. We plot the CDF curves in Figure 5. All methods exhibit high estimation performance. DeepSync models exhibit marginally different performances due to the influence of the number of regres-

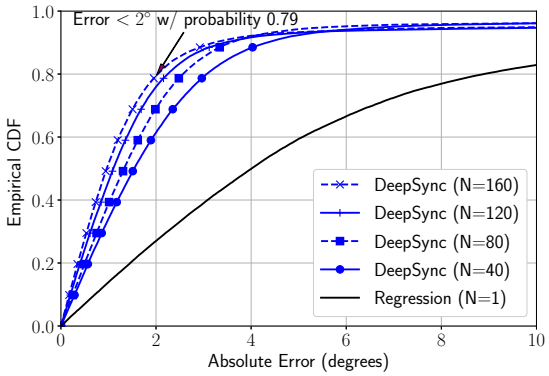

Figure 4. *Cumulative distribution of rotation estimation errors using DeepSync with template-based watermarks for varying number of regression heads N for rotation and 10 regression heads for scale.*

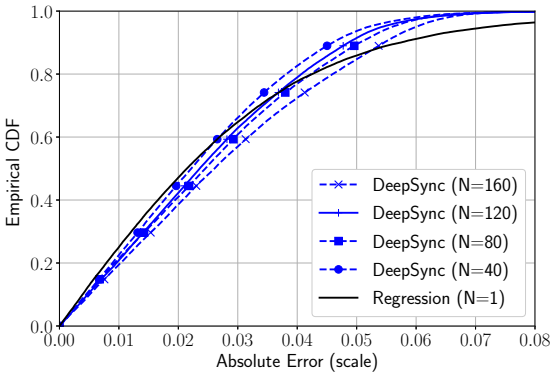

Figure 5. *Cumulative distribution of scale estimation errors using DeepSync with template-based watermarks for varying number of regression heads N for rotation and 10 regression heads for scale.*

sion heads for rotation N, while the number of regression heads for scaling is fixed at 10.

HiDDeN Deep Watermark

HiDDeN deep watermarks are used in this investigation to compare with our proposed method. HiDDeN watermarks are content-adaptive–i.e., there is no predefined pattern or template overlayed on the marked images for watermark synchronization [8]. Briefly, an image and a length-L bit payload are given as input to a CNN which outputs the watermarked image. For simplicity, in this study we consider only rotation transform in $\Phi = [-180^\circ, 180^\circ)$.

We first train a reduced complexity HiDDeN invariant to very small rotations–i.e., we train encoder, decoder, and discriminator with 32 channels per convolutional layer for payload length $L \in \{21, 22\}$. A host image is resized and cropped to size 384-by-384 before encoding. The watermarked image is rotated by random angle $\phi \in \Phi$. The rotated image is then inverse transformed by angle $\tilde{\phi} = \mathscr{P}(-\phi \pm \varepsilon)$, where $\mathscr{P}(\cdot)$ projects its input argument onto Φ and $\varepsilon \sim \mathscr{U}(0, \varepsilon_{\max} = 10^\circ)$.[1] Then, a random crop of size 128-by-128 is extracted from the inverse transformed within $|\varepsilon|$ degrees image and given as input to the decoder for payload extraction. By introducing rotation and its inversion in the training process, we allow the decoder to learn payload extraction despite rotation artifacts. Moreover, by inverting the image within $|\varepsilon|$ degrees, we introduce some synchronization error-tolerance. In fact, a visual inspection of Figure 1 suggests that a precise synchronization may not be necessary.

Then, we freeze HiDDeN and train DeepSync by utilizing the HiDDeN encoder as the embedding process. We utilize EfficientNet_B3 as a backbone and fix $N = 40$ regression heads for rotation. As before, we train a model with one regression head for rotation as a benchmark attempting to solve the synchronization problem via direct regression.

We commence with an evaluation of the watermark presence/absence classification. We illustrate the AUC curves in Table 2 where we observe that, like with template watermarks, all models exhibit high AUC performance with the benchmark model–i.e., regression–exhibiting marginally higher performance. Additionally, the marginally higher best AUC for HiDDeN can be attributed to the use of a larger backbone.

[1] $\mathscr{U}(a,b)$ denotes the uniform distribution with lower bounds a and b, respectively.

Table 2: HiDDeN Watermark presence/absence classification.

Model	L	AUC
Regression (N=1)	22	1.000
DeepSync (N=40)	21	0.998
DeepSync (N=40)	22	0.997

We continue with an assessment of the rotation estimation performance that DeepSync exhibits. We plot the CDF curves in Figure 6. We notice that DeepSync exhibits high estimation performance, estimating rotation within $\varepsilon_{\max}$-accuracy for over 90% of the samples. In contrast, the benchmark model achieves same accuracy levels for only 60% of the samples. Interestingly, when DeepSync makes errors, these align within either a 90-degrees or 180-degrees symmetries indicating informed and systematic inaccuracies rather than random mistakes.

Finally, we measure the per sample Bit Error Rate (BER) that the HiDDeN decoder exhibits after watermarked images are inverse transformed based on DeepSync rotation estimates before payload extraction. We plot the corresponding CDF curves in Figure 7. As expected, we notice exact payload recovery for over 90% of the samples when relying on DeepSync rotation estimates compared to exact payload recovery for less than 80% of the samples when relying on rotation estimates from the benchmark model.

Conclusions

Transform invariant extraction has become the norm in deep watermarking literature. Consequently, existing deep watermarking approaches lack precise recovery of geometry–i.e., synchronization. We empirically observe that absence of precise recovery is associated with reduced payload capacity and/or watermark imperceptibility degradation. Reduced payload capacity, in turn, influences false positive rates and applicability to large scale deployments. In this work, we proposed DeepSync, a new deep

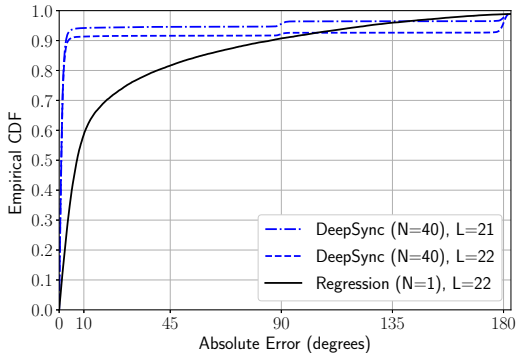

Figure 6. *Cumulative distribution of rotation estimation errors using DeepSync with HiDDeN watermarks for varying number of regression heads N for rotation and varying number of information bits L.*

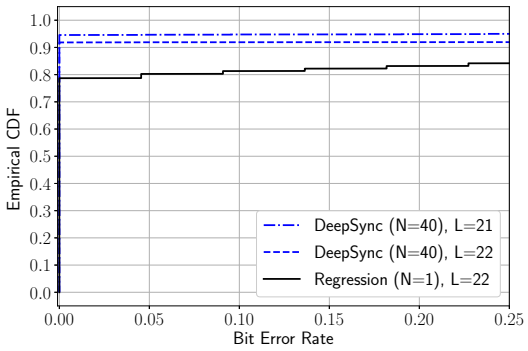

Figure 7. *Cumulative distribution of Bit Error Rate (BER) using DeepSync with HiDDeN watermarks for varying number of regression heads N for rotation and varying number of information bits L.*

learning approach for watermark synchronization applicable to both template-based and deep watermarks which explicitly estimates geometry enabling inversion. DeepSync lays the groundwork for developing deep watermarking systems that have the potential to be more practical at scale, paving the way for further advancements in the field.

Acknowledgments

We extend our sincere gratitude to Ravi Sharma, Tony Rodriguez, Steve Stewart, and Joel Meyer of Digimarc Corporation for their invaluable comments and insights that significantly enhanced the quality of this work.

References

[1] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, Digital Watermarking and Steganography, Morgan Kaufmann, 2007.

[2] BBC, How invisible barcodes could increase plastic recycling, *Online*, Available: `https://www.bbc.com/news/av/business-59552404`, (2021).

[3] L. Rosenthol, C2PA: The World's First Industry Standard for Content Provenance, Proc. SPIE, Applications of Digital Image Processing, (2022).

[4] The White House, Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence, *Online*, Available: https://www.whitehouse.gov, (2023).

[5] T. Rodriguez, O. Alattar, H. Brunk, J. Meyer, W. Conwell and A. Kamath, Learning Systems and Methods, U.S. Published Application No. US20210217128A1 17/152,498, (2021).

[6] G. Rhoads, and R. Sharma, Enhanced Neural Network Systems and Methods, US Published Patent Application No. US20210357690A1, (2017).

[7] A. Kamath, C. Ambiel, U. Deshmukh, Artwork Generated to Convey Digital Messages, and Methods/Apparatuses for Generating such Artwork, US patent 11,704,765, (2017).

[8] F. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, Hidden: Hiding Data with Deep Networks, Proc. European Conf. Computer Vision (ECCV), pp. 657-672, (2018).

[9] X. Luo, R. Zhan, H. Chang, F. Yang and P. Milanfar, Distortion Agnostic Deep Watermarking, Proc. Conf. Computer Vision and Pattern Recogn., Seattle, WA, pp. 13545-13554, (2020).

[10] J. Hayes, K. Dvijotham, Y. Chen, S. Dieleman, P. Kohli, and N. Casagrande, Towards transformation-resilient provenance detection of digital media, arxiv.2011.07355v1, *Online*, Available: `https://arxiv.org/abs/2011.07355`, (2020).

[11] P. Fernandez, A. Sablayrolles, T. Furon, H. Jégou and M. Douze, Watermarking Images in Self-Supervised Latent Spaces, Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process. (ICASSP), Singapore, Singapore, pp. 3054-3058, (2022).

[12] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon, The Stable Signature: Rooting Watermarks in Latent Diffusion Models, Proc. IEEE Int. Conf. Computer Vision, Paris, France, pp. 22409-22420, (2023).

[13] X. Luo, M. Goebel, E. Barshan, and F. Yang, LECA: A Learned Approach for Efficient Cover-Agnostic Watermarking, Electronic Imaging, pp. 1-6, (2023).

[14] T. Bui, S. Agarwal, N. Yu, and J. Collomosse, RoSteALS: Robust Steganography using Autoencoder Latent Space, Proc. Int. Conf. Computer Vision Pattern Recogn., pp. 933-942, (2023).

[15] T. Bui, S. Agarwal, and J. Collomosse, TrustMark: Universal Watermarking for Arbitrary Resolution Images, arXiv:2311.18297, *Online*, Available: `https://arxiv.org/abs/2311.18297`, (2023).

[16] T. Y. Lin, M. Maire, S, Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, Microsoft COCO: Common Objects in Context, Proc. European Conf. Computer Vision, pp. 740-755, (2014).

[17] A. Reed, T. Filler, K. Falkenstern, and Y. Bai, Watermarking Spot Colors in Packaging, Proc. Media Watermarking, Security, and Forensics, pp. 46-58, (2015).

[18] S. Gidaris, P. Singh, and N. Komodakis, Unsupervised Representation Learning by Predicting Image Rotations, arXiv:1803.07728, *Online*, Available: `https://arxiv.org/abs/1803.07728`, (2018).

[19] S. Czolbe, O. Krause, I. Cox, C. Igel, A Loss Function for Generative Neural Networks Based on Watson's Perceptual Model, Advances in Neural Information Processing Systems 33, pp. 2051-2061, (2020).