RiemStega: Covariance-based loss for print-proof transmission of data in images

Aniana Cruz[1*] Guilherme Schardong[1] Luiz Schirmer[2] João Marcos[1] Farhad Shadmand[1] Nuno Gonçalves[1,3]

[1] Institute of Systems and Robotics, University of Coimbra
[2] University of the Sinos River Valley
[3] Portuguese Mint and Official Printing Office

Abstract

Covariance matrices outperform first-order features in many tasks, attracting considerable attention from the computer vision research community. Covariance matrices encode second-order statistics between features, at the same time it is robust to noise. Based on this, we propose representing images by covariance matrices and defining a loss function that measures the distance between them through the Riemannian distance. Motivated by the robustness and invariance properties of the affine invariant Riemannian metric the proposed method was validated in printer-proof data transmission, which is a challenging task due to the trade-off between image quality and message recovery capabilities after printing and digitization procedures. The effectiveness of this approach was systematically assessed using MS COCO and IMM Face datasets. The results demonstrated that the proposed approach outperforms conventional methods that use Euclidean distance, generating encoded images with better quality and achieving higher recovery accuracy in printed images. Additionally, a broader application of the proposed loss was successfully tested in image generation tasks, using generative adversarial networks (GANs).

1. Introduction

The security of information and documents is of considerable interest to both academia and industry. Steganography and watermarking are the most common methods for concealing a secret message which can be text, image, or video inside a digital medium (e.g. cover image). The main goal of steganography is to provide secure and covert communication, where only the sender and the intended receiver should know the existence of the secret message, without raising suspicion of unauthorized parties. On the other hand, watermarking is often used to assign ownership, for example, to identify the author of content, or to verify the authenticity and integrity of data. The message must be robust to noise so that it is possible to recover the information even after distortion, therefore watermarking prioritizes robustness over secrecy. Although the techniques have different purposes, they share a similar process, encoding and decoding secret information. The encoding algorithm needs to generate encoded images (cover image with the secret message) similar to the cover image with minimal distortion, and the decoding algorithm must reliably recover the secret message [3, 32].

The covariance matrix has recently emerged as a crucial tool for data representation due to its ability to provide a compact representation of data, encode linear correlations between features, and be robust to noise. First introduced as a region descriptor in [38], the covariance matrix has been widely used in various applications, including classification, detection, and object tracking. The covariance matrix does not lie in Euclidean space, instead, it lies in a Riemannian manifold of the symmetric and positive definite (SPD) matrix. Consequently, a natural approach to analyzing the covariance matrix involves using metrics that capture the geometric structure of the SPD manifold. Common methods for comparing SPD matrices include Riemannian metrics such as the affine invariant Riemannian metric (AIRM) [29] and the bi-invariant log-Euclidean metric (for a zero-curvature manifold) [2]. Besides Riemannian metrics, Bregman divergences, such as the Jeffrey and Stein divergences, can also be employed [8].

This study proposes representing the images using covariance matrices and defining a loss function based on the distance between these matrices. Affine-invariant Riemannian distance is used to calculate the similarity between covariance matrices. Riemannian distance is selected because it is an accurate distance measure that considers the geometry of the manifold, meaning that the distance between

*anianabrito@isr.uc.pt
This work has been supported by Fundação para a Ciência e a Tecnologia (FCT) under the project UIDB/00048/2020.

two matrices depends both on the values of their elements and on their relative positions in the manifold. Therefore, it provides useful information for making the hidden message less perceptible and generating encoded images more similar to cover images. Furthermore, Riemannian distance has important invariance properties such as invariance under affine transformations (e.g. rotation, and resizing) and inversion and it is robust to noise and transformations that can occur in the data [9,27]. Consequently, this new loss can make encoded images more robust to changes introduced by the printing and digitization processes.

The main use case of the proposed loss is data transmission in printed images. The security of both electronic and printed documents is important, however, the majority of studies focus on digital images [6, 16, 35]. Printer-proof data transmission is more challenging because the decoder should be robust to the unpredictable transformation that occurs during the printing and digitization process, such as variations in contrast, the perspective of the acquired image, color, and thermal noise [10]. Few works have addressed the issues of printed images. StegaStamp [36] was the first watermarking/steganography method that obtained encouraging results in printed images, and it stands out as the current state-of-the-art (SoTA). StegaStamp uses L2 distance as a loss function to compare stego and cover images, thus it ignores image structures and it is vulnerable to noise. To address these limitations, we introduce a new approach that represents the images as covariance matrices and generalizes the loss function to the Riemannian manifold. To the best of our knowledge, this is the first work that investigated the effects of the Riemannian distance in printer-proof encoding and decoding of data in images.

Nevertheless, this proposed loss has broad applications that can be used in many computer vision applications, which require accurate computation of image similarity such as image retrieval and face/object recognition. The approach was validated in two applications, printer-proof watermarking, and GAN for image-generative tasks. The efficiency of the proposed approach was assessed in digital images, images displayed on the computer screen, and printed images regarding both image quality and performance when extracting the hidden message. The results demonstrated that it generated encoded images with better quality. The recovery capability was evaluated in printed images of two publicly available datasets, using three different image sizes and two printers. The proposed method outperformed StegaStamp, the current SoTA, by achieving superior-quality of encoded images and higher decoding accuracy in printer-proof scenarios.

2. Related Works

Use of covariance matrices: Covariance matrices have been successfully applied in diverse tasks, including emotion recognition, object detection, texture classification, medical image, and brain-computer interface [7, 9, 38]. These matrices have been employed for representing images, videos, and 3D point clouds, mainly in classification and recognition tasks [14, 17]. Recently, in the realm of deep learning, the covariance matrices have been utilized as a representation of convolutional features and also as a part of network architectures [7,20,40]. In [20], the author introduced an end-to-end manifold deep network to non-linearly learn SPD matrices on Riemannian manifolds. This architecture is designed to receive SPD matrices as inputs and preserve their structure across layers.

Watermarking and Steganography work: Deep learning has presented remarkable outcomes in many fields of computer vision, thus there is an increasing interest in using it for watermarking and steganography as well. Here, we focus only on watermarking and steganography methods based on deep learning. For an overall review on this subject see [5, 35]. Several studies have been successful in hiding the secret message while maintaining image quality and large information capacity [33, 42]. The size of secret messages directly affects the appearance of encoded images. Many works have used convolutional neural networks to encode and recover the secret message [12,30,37]. Other approaches based on GANs were also proposed [34, 44]. The first end-to-end neural network method to embed a watermark in a cover image was introduced in [44], which proposed the HiDDeN (Hiding Data With Deep Networks) algorithm. HiDDeN is composed of four main components: an encoder, a noise layer, a decoder, and an adversarial discriminator. To improve the robustness, the authors added the noise layer between the encoder and decoder, which distorts the encoded image by applying six different types of transformations. The approach was evaluated on digital images, achieving high quantitative and qualitative performance. The study in [36] extended the validation for printed images and developed a new method called StegaStamp. Based on the idea of previous works that integrate noise simulation to increase the robustness of image transformation, the authors propose a new noise simulation module that adds many different pixel-wise and spatial image corruptions. The results showed good decoding performance in a controlled illumination condition for the acquired printed encoded images and good image quality. However, some encoded images had perceptible artifacts. To overcome this limitation, and generate more natural encoded images we use Riemannian distance instead of L2 distance to compare the images.

3. Riemannian Geometry principles

This section describes some basic concepts of the Riemannian manifold of symmetric positive definite matrices. Riemannian manifolds are smooth manifolds equipped with

a Riemannian metric that determines an inner product on tangent spaces [24]. The Riemannian manifold of SPD matrices is a powerful mathematical tool that provides metrics that can be applied directly in the space of covariance matrices. It has applications in many different areas like mathematics, physics, and engineering. In computer vision, the Riemannian manifold of SPD matrices has been used to develop efficient algorithms for image generation [15], face recognition [45], pedestrian detection [39], and image classification [40], among others.

3.1. Mathematical notation

The following definitions and notation will be used:

- $M(d) \in \mathbb{R}^{d\times d}$ denotes the space of $d \times d$ square matrices;
- $S(d) \in M(d)$ is the set of all $d \times d$ symmetric matrices in the space of $M(d)$;
- $P(d) \in S(d)$ represents the space of all SPD matrices;
- $\mathcal{S}^d_{++}$ is the SPD manifold, and
- $GL(d)$ is set of real invertible $d \times d$ matrices.

3.2. Symmetric and positive definite manifold

Let $v \in \mathbb{R}^d$ be a nonzero vector, the matrix $P \in \mathbb{R}^{d\times d}$ is said to be SPD matrices if $v^T Pv > 0$. The SPD manifold $\mathcal{S}^d_{++}$ consists of a commutative Lie group formed by the space of $d \times d$ SPD matrices [41]

$\mathcal{S}^d_{++} = \{P \in \mathbb{R}^{d\times d} : P = P^T, v^T Pv > 0, \forall v \in \mathbb{R}^d \setminus \{0_d\}\}$, where $\mathbb{R}^d \setminus \{0_d\}$ denotes the $\mathbb{R}^d$ space without the zero vector.

3.3. Riemannian Distance

The Riemannian distance between any two SPD matrices P_1 and $P_2 \in \mathcal{S}^d_{++}$ is the shortest length of all admissible curves (geodesic) connecting them as defined by [27]

$$\delta_R(P_1, P_2) = \|log(P_1^{-1}P_2)\|_F = \left[\sum_i log^2\lambda_i\right]^{\frac{1}{2}} \quad (1)$$

where $log(.)$ denotes the matrix logarithm, $\|.\|_F$ is the Frobenius norm of a matrix, and λ_i are the real eigenvalues of $P_1^{-1}P_2$.

The Riemannian distance $\delta_R(.,.)$ has several invariance properties. Here we explore its invariance under affine transformations (e.g. translations, rotations, and scaling) by any invertible matrix $A \in GL(d)$. The affine invariance property of the Riemannian distance guarantees that:

$$\delta_R(A^T P_1 A, A^T P_2 A) = \delta_R(P_1, P_2). \quad (2)$$

Affine invariance is an especially appealing property for printer-proof watermarking tasks since it ensures that the distance remains unchanged under different transformations that can occur in printed images, such as rotations, and scale.

4. Materials and Methods

4.1. Covariance Loss

In this study, each RGB image is represented as covariance matrices, which describe the relationship between the color channels of each pixel in the image. The covariance matrix can provide useful information about the structure of the image, with the diagonal values representing the variance of each color channel, and the nondiagonal values representing the correlations. The covariance matrix decreases the impact of noisy samples due to the averaging step during its computation [38]. Let us consider an image $I \in \mathbb{R}^{W\times H\times C}$ with W width, H height, and C number of channels. The covariance matrix ($\mathbf{C}ov \in \mathbb{R}^{C\times C}$) is computed as

$$\mathbf{C}ov = \frac{1}{n}\sum_i^n (\widetilde{I}_i - \mu)^T(\widetilde{I}_i - \mu) \quad (3)$$

where $\widetilde{I} \in \mathbb{R}^{n\times C}$ is the reshape of I, $n = W \times H$, and μ is the mean of $\widetilde{I}$. To avoid singularity, a very small value is added to each element of the matrix $\mathbf{C}ov = |\mathbf{C}ov| + 1 \times 10^{-12}$. Given a set of k original images $I^{'} = \{I^{'}_1, ..., I^{'}_k\}$, and a set of k generated images $I^{''} = \{I^{''}_1, ..., I^{''}_k\}$, the SPD matrices are obtained through their covariance matrix $\mathbf{C}ov^{'}$ and $\mathbf{C}ov^{''}$ respectively. The loss function is defined as

$$\mathbf{R}_{loss} = \frac{1}{k}\sum_i^k \delta^2_R(\mathbf{C}ov^{'}_i, \mathbf{C}ov^{''}_i) \quad (4)$$

where $\delta_R(.,.)$ is the Riemannian distance (Equation (1)). In this work, we propose to use the Riemannian distance to measure the similarity between images due to its invariance properties and robustness to noise. Therefore, we expect that the proposed method, hereinafter referred to as RiemStega, will lead to more similar generated and original images, while being more robust to noise that occurs during printing and digitalization such as color noise, misalignment of image channels [10], thereby leading to images with higher quality and message recovery capability.

The complete loss function is defined as follows:

$$L = r \times \mathbf{R}_{loss} + p \times L_P + w \times L_W + m \times L_M \quad (5)$$

where L_P is the LPIPS perceptual loss function [43], L_W is the Wasserstein loss [1], L_M is the cross-entropy loss, and r, p, w, and m are the weights for each loss function component.

4.2. Training Dataset

Similar to the StegaStamp [36], we use the MIRFLICKR dataset [21] for training. The MIRFLICKR dataset consists of 25000 images including 10 topics (e.g. sky, water, people, and animals), and many subtopics. During training the images were rescaled to a 400×400 resolution and embedded with randomly generated binary messages.

4.3. Testing Datasets

The effectiveness of the proposed approach is evaluated using MS COCO [25] and IMM Face [28] datasets. The MS COCO dataset comprises 328,000 images with 91 common object categories such as person, train, airplane, etc. The IMM Face dataset contains 240 images of 40 different human faces (7 females and 33 males) with neutral and happy expressions.

4.4. Optimization

For the training process, the Riemannian ADAM optimizer [4] is used, which is a generalization of ADAM optimizer to Riemannian manifolds. It has faster convergence and a lower training loss value. We used a product manifold with a learning rate of 1×10^{-4}.

The model being optimized follows an encoder-decoder architecture, detailed and illustrated in Supp. Mat.

Hardware configuration We used an NVIDIA GeForce RTX 3090 GPU, with 24 GB of memory, with an AMD Ryzen Threadripper PRO 5965WX CPU and 256 GB of DDR4 memory.

4.5. Evaluation Metrics

Image quality is usually determined based on a set of factors, such as contrast, resolution, noise, level of resulting artifacts, and distortion degree. Herein, the quality of encoded images was assessed using the following image quality metrics: peak signal noise ratio (PSNR), learned perceptual similarity metric (LPIPS), and structural similarity index (SSIM). GAN results are also assessed by Fréchet Inception Distance (FID) [18]. The higher value of PSNR and SSIM means better encoded image quality, and for LPIPS and FID, the opposite is true (lower is better). The techniques were also evaluated regarding the capacity to recover hidden messages. The performance was assessed by the accuracy computed as a ratio between the number of decoded images and the number of total images.

5. Experimental results for watermarking task

The feasibility of the proposed method and its impact on both image quality and accuracy of extracting the hidden messages is evaluated in two datasets and validated in a set of experiments including digital images, images captured on the computer screen, and printed images.

5.1. Image quality measures

The quality of the encoded images is assessed through human visual perception and image quality assessment metrics.

5.1.1 Factors influencing image quality

The encoded image is generated by adding a residual component (the gray images in Figure 1) to the cover image. The quality of encoded images is directly affected by both the length of the hidden message and the amount of the residual component added to the cover image. We compared the quality of encoded images using message lengths of 50, 100, 150, and 200 bits. The results presented at the top of Figure 1 and Table 1 reveal that the quality of the images is degraded with the increase in message length. In this study, we chose a message length of 100 bits since it has been demonstrated that it offers a good trade-off between image quality and information transfer.

The assessment of how the amount of added residual influences the quality of the encoded image was performed by varying the percentages of residual added to the cover image in 100%, 80%, 60%, and 40%. Similarly, the results illustrated at the bottom of Figure 1 showed a decrease in image quality with an increase in the fraction of residual. Having encoded images with very high quality is pointless if the messages cannot be recovered. Therefore, we analyzed what is the minimum percentage of residual required to achieve 100% decoding accuracy in three scenarios: 1) digital images, randomly selecting 500 images from the MS COCO dataset, and then encoding and decoding the message always in the digital realm; 2) images captured from a screen (monitor HP Inc. 27" with a resolution of 1440×900), 20 images were randomly selected and encoded, then each image was displayed on a computer screen and 10 photos were captured from each image (totaling 200 samples for each residual level) and decoded; 3) printed images, encoding the same 20 images, and printed on A4 paper sheets with size 10cm $\times$ 10cm. Then 10 photos were captured from each printed image and decoded. Table 2 shows that the message recovery performance is affected by the transmission means. To achieve 100% accuracy in digital, screen, and printed images we need 60%, 80%, and 100% of the residual respectively. When using the 60% of the residual, there is a substantial drop in performance for printed images, showing that a direct comparison between the results obtained using digital images and printed images is not straightforward.

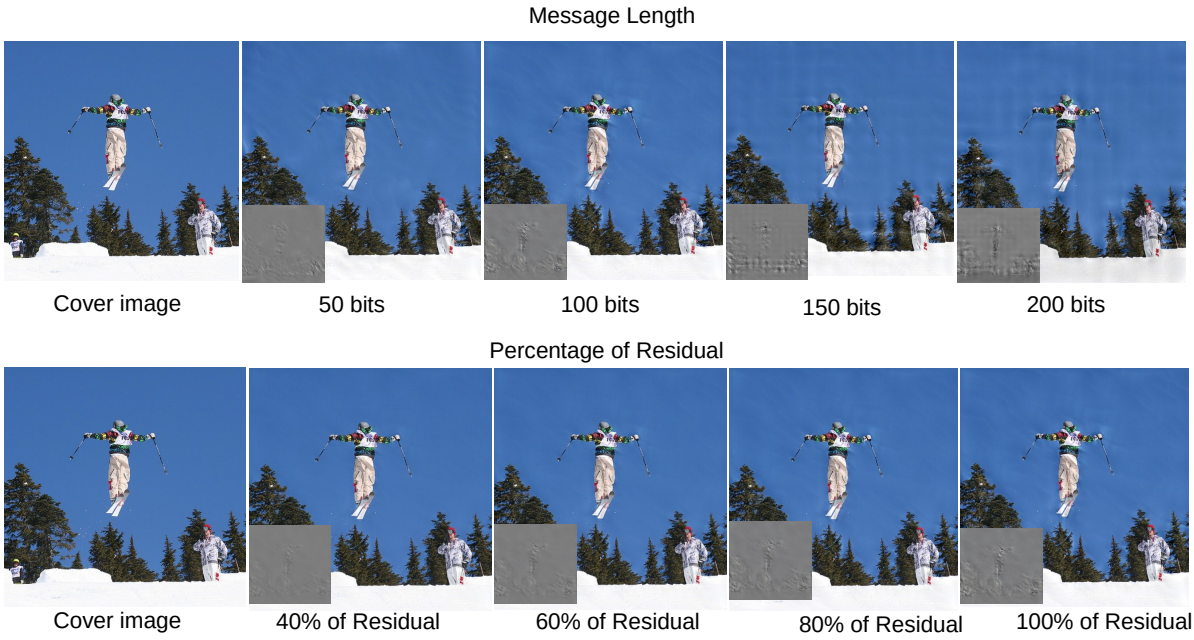

Figure 1. Samples of encoded images obtained using: Top) different message lengths using full residual (100%) and, Bottom) different percentage of residual added to the cover image using a message length of 100 bits. Residual is the gray image in the bottom left corner.

Table 1. Image quality of 500 images randomly selected from MS COCO dataset using models trained with different message lengths.

Message length (bits)	50	100	150	200
SSIM ↑	0.965	0.949	0.917	0.901
PNSR ↑	32.413	30.031	27.058	24.732
LPIPS ↓	0.018	0.024	0.034	0.041

Table 2. Decoding performance for digital images, images captured from a computer screen and printed image using RiemStega method (Ours).

Percentage of residual	40%	60%	80%	100%
Digital images	64.2	100.0	100.0	100.0
Screen Images	36.5	97.5	100.0	100.0
Printed images	0.0	67.0	99.0	100.0

5.1.2 Image quality assessment metrics

We compare the proposed algorithm (RiemStega) with the following state-of-the-art methods: StegaStamp [36], SSL [16], and RoSteALS [6]. We also tested the ARWGAN [19], CIN [26], and PIMoG [13] methods, however, with images resized bigger than 128 × 128, the result is unfocused and not comparable. For this reason, the results are not included here. Considering that SSL and RoSteALS were validated in digital images, we also evaluated our method with 60% residual (RiemStega60) which obtained 100% recovery accuracy in digital images. Figure 2 shows illustrative examples of cover images and encoded images generated by each method using the MS COCO dataset and IMM dataset. All methods performed better in heterogeneous images characterized by diverse colors and rich in details (first column) than in homogeneous images with few details. Images created by RiemStega60 and RoSteALS presented similar quality metrics, being superior to the other methods. However, 81% of RoSteALS digital images were decoded successfully, as opposed to 100% of RiemStega60 images. Visual perception is intrinsically subjective, thus we proceed with a quantitative assessment based on SSIM, PSNR, and LPIPS of 500 images.

Table 3 compares the qualities of the encoded images based on the quality assessment metrics described previously. The results showed the superiority of RiemStega60 in all metrics, except for PSNR for the MS COCO dataset. However, it is arguable whether these metrics match our visual perception. In [31] the author presented the weakness of PSNR. The RiemStega method achieved better average values for the three metrics compared to StegaStamp, suggesting an improvement in encoded image quality. These results revealed that our approach generates images with higher quality than StegaStamp regardless of the dataset. The RiemStega uses a message length of 100 bits and adds 100% of residual to the cover image because our focus is

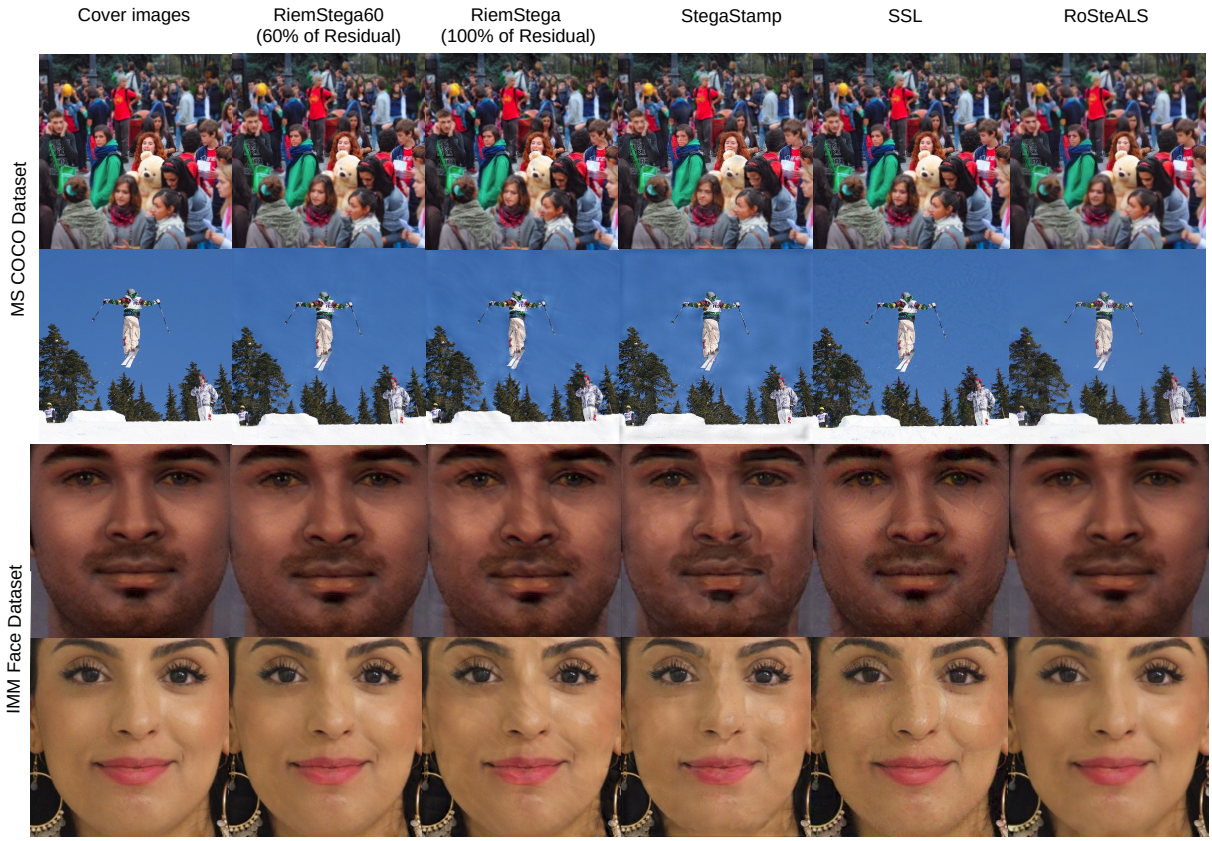

Figure 2. Samples of cover images and encoded images obtained by employing different techniques using MS COCO and IMM Face datasets. It is important to note that only 81% of encoded images generated with RoSteALS were decoded digitally.

printed images. However, other approaches can be used, for example, using the message length of 30 bits frequently employed in some studies, and adding 60% of residual we achieved values of 0.99, 38.00, and 0.01, for SSIM, PSNR, and LPIPS respectively, which significantly improves the visual image quality.

5.2. Analysis of methods for similarity measures

There are several metrics for measuring similarity between SPD matrices, and the most commonly used are AIRM, Log-Euclidean, and Bregman divergences. In our RiemStega approach, AIRM was chosen to measure the similarity between the covariance matrices of the encoded and cover images, due to its robustness to noise and its strong invariance properties. We trained additional models using Log-Euclidean and Jeffrey divergence metrics to support our choice. Then, these models were evaluated on the MS COCO dataset to assess their impact on image quality. The Log-Euclidean method achieved mean SSIM, PSNR, and LPIPS values of 0.93, 27.0, and 0.03, respectively, while Jeffrey divergence obtained values of 0.94, 28.7, and 0.03. Notably, the AIRM method generated images of superior quality.

5.3. Impact of proposed loss function

To assess the effect of the proposed loss function on image quality, we trained a new model excluding the $\mathbf{R}_{loss}$. In this configuration, the loss function (Equation (5)) included only three components: LPIPS, Wasserstein loss, and cross-entropy loss. Using the MS COCO dataset, this model achieved mean values of 0.93 for SSIM, 25.57 for PSNR, and 0.03 for LPIPS. With the addition of the proposed loss, there was an improvement in all metrics. Additionally, we measured the time needed to calculate δ_R. We generated two random matrices $A, B \in \mathbb{R}^{1024\times1024\times3}$ and calculated $\delta_R(A, B)$ 1000 times for statistical measurements, with 10 steps for warm-up. The results show an average running time of 0.052 s per call, indicating a measurable, but not significant impact on training performance.

5.4. Recover capability results

In order to evaluate the robustness of the proposed approach regarding the capacity to extract the hidden message, a set of experiments was performed using digital and printed images.

Table 3. The SSIM, PSNR, and LPIPS between stego and cover images using MS COCO and IMM Face Dataset. The best values are in bold.

	MS COCO Dataset			IMM Face Dataset		
Models	SSIM ↑	PSNR ↑	LPIPS ↓	SSIM ↑	PSNR ↑	LPIPS ↓
RiemStega60 (Ours)	**0.98**	34.39	**0.01**	**0.98**	**36.98**	**0.01**
RiemStega (Ours)	0.95	30.03	0.02	0.96	32.61	0.03
StegaStamp	0.89	28.47	0.03	0.92	30.75	0.04
RoSteALS	0.94	30.36	0.03	0.96	35.62	0.02
SSL	0.95	**36.41**	0.02	0.93	36.42	0.06

5.4.1 Robustness evaluated in digital images

RiemStega integrates Spatial Transformer Networks (STN) [23], which provides some invariance properties to scale, rotation, and more generic warping. In order to evaluate the invariance of the Riemannian approach we trained the same model excluding STN (RiemStega without STN). Therefore, we assessed the robustness of the Riemannian approach with and without STN. The robustness of the methods was evaluated using different transformations, such as rotation, resize, blur, JPEG, contrast, and brightness. The results presented in Figure 3, showed that the Riemannian methods outperformed the other three methods in almost all transformations. Although they have lower performance than SSL in rotations bigger than 15 degrees, they have stronger robustness for small values of rotations. The Riemannian model without STN was as effective as the Riemannian model with STN, showing the effectiveness of the Riemannian distance under these transformations.

5.4.2 Robustness evaluated in printed images

In the experiments, we first randomly selected 30 images from each dataset, and then embedded 100-bit messages, which comprise hash codes with eight characters. Afterward, the hashes are converted into bit strings and applied the error-correcting codes Bose-Chaud huri-Hocquenghem (BCH) [11]. The decoding capabilities were tested in printed images with three sizes: 3cm $\times$ 3cm, 5cm $\times$ 5cm, and 10cm $\times$ 10cm. These images were printed with two different printers: printer 1 (Konica Minolta C360i), and printer 2 (brother-HLL3270CDW-series). In order to keep the same distance to the camera and the same surrounding illumination sources, the printed images are uniformly positioned, and then photographed using a Samsung Galaxy S22 Ultra. These pictures were cropped and rectified with classic image processing methods. A magenta boundary with a width of 5 pixels on each side is added to each printed image (see supplementary material). Then, the images are converted to grayscale and binarized using a threshold value of 160. This threshold was set empirically to detect only the content inside of the magenta border. Afterwards, a binary hole-filling technique was applied to detect the bounding boxes encompassing all regions bigger than $20,000$ pixels. All regions with an aspect ratio bigger than 0.82 are then selected for decoding. All these thresholds were set empirically to minimize false detections.

For each printed image, 10 photos were captured, having 300 samples for each image size. In total, considering the three sizes we have 900 samples for each printer. SSL and RoSteALS methods were tested only in the MS COCO dataset and one printer because none of the printed images were decoded. The decoding results are reported in Table 4. For RiemStega, all captured images with sizes of 5cm, and 10cm were decoded when using the IMM Face dataset (100% accuracy) and almost all with the MS COCO dataset (accuracy higher than 94%). For images with a size of 3cm, there was a slight decrease in accuracy with the IMM Face dataset and a higher decrease with the MS COCO dataset (accuracy of 61.3%). The StegaStamp method performed well for image sizes of 10cm and 5cm with accuracy higher than 86.7% and 70% respectively. For images with a size of 3cm, there was a strong accuracy decrease, achieving in one case a very low value (accuracy of 27%). This performance degradation indicates that StegaStamp is more dependent on image size. These results show evidence that RiemStega is a more robust solution. Furthermore, RiemStega appears to have good generalization across datasets and printers.

6. Results for image generation task

To show a wider application of the proposed approach, preliminary experiments were conducted involving the generation of images using GAN. To evaluate the proposed loss, we train a pix2pix network [22] with $256 \times 256 \times 3$ resolution on the Edges2Shoes dataset [22]. We define the generator loss (G_{Loss}) as the combination of LPIPS and proposed loss, that is, $G_{Loss} = LPIPS + \mathbf{R}_{loss}$. The training was performed using the Edges2Shoes dataset (49.8k images for training and 200 images for testing). A learning rate of 1×10^{-5} and a batch size of 8 was employed during the training process. The other setups are the same as the original code published by the authors. The results reported in Figure 4 showed that the proposed method generated images closer to the ground truth than the conventional

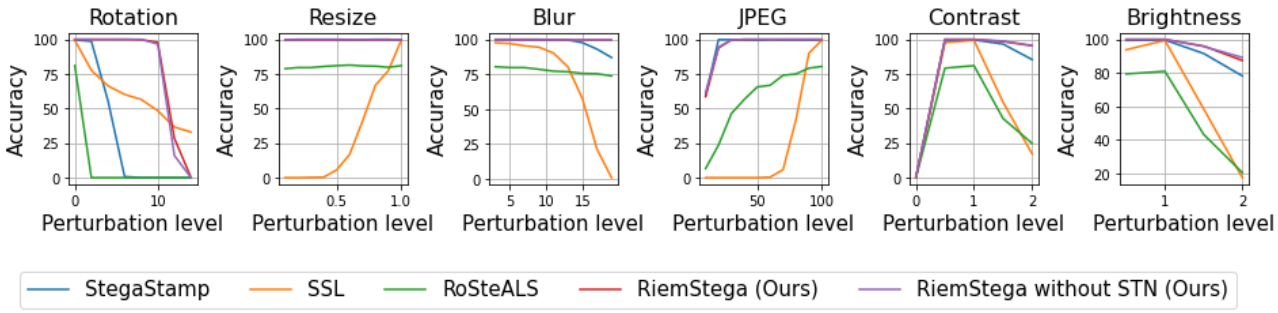

Figure 3. Decoding accuracy after different transformations, namely, rotation angle, resize, blur, JPEG, contrast, and brightness.

Table 4. Decoding accuracy for printed images using MS COCO Dataset and IMM Face Dataset.

		MS COCO Dataset						IMM Face Dataset			
		Printer 1		Printer 2				Printer 1		Printer 2	
	Size (cm)	Riem Stega (Ours)	Stega Stamp	Riem Stega (Ours)	Stega Stamp	RoSteALS	SSL	Riem Stega (Ours)	Stega Stamp	Riem Stega (Ours)	Stega Stamp
Accuracy	10x10	**99.7**	99.3	**97.0**	86.7	0	0	**100.0**	98.7	**100.0**	95.7
	5x5	97.7	**99.0**	**94.0**	70.0	0	0	**100.0**	81.0	**100.0**	94.0
	3x3	**76.7**	58.3	**61.3**	27.0	0	0	**99.3**	74.0	**93.7**	56.0

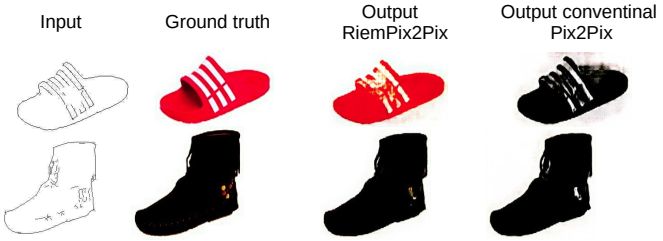

Figure 4. Images generated using RiemPix2pix and conventional Pix2pix based on the input image to produce the corresponding ground truth.

Table 5. The SSIM, PSNR, LPIPS, and FID between original and generated images using conventional Pix2pix and Pix2pix with the proposed loss (RiemPix2pix) using 200 images.

Metrics	SSIM ↑	PNSR ↑	LPIPS ↓	FID ↓
RiemPix2pix	**0.629**	**10.7**	**0.249**	**82.5**
Pix2pix	0.610	9.9	0.254	100.5

algorithm, and Table 5 demonstrated that they have better quality.

7. Limitations

Although we achieved a good trade-off between image quality and decoding capability, there are still many challenges ahead. Some encoded images still have visible artifacts, mainly homogeneous images with few details. As future work, it is intended to exploit: 1) alternative ways to merge the cover image and the residual, such as alpha blending, and image gradients, and 2) new neural network architectures. In some cases the quantitative metrics do not match human perception, so in future work we will develop metrics more aligned with our visual perception.

8. Conclusions

This study proposes an approach that represents images through covariance matrices and defines a new loss function based on Riemannian distance, which aims to generate encoded images that are more similar to the original ones. It also uses invariance properties of Riemannian distance to tackle the transformation that usually happens in printed images. The robustness of the proposed approach was evaluated in two different tasks, namely, printer-proof watermarking, and image-generative task using GANs. This approach proves to be effective in generating encoded images with higher quality and higher decoding accuracy. The results show promising evidence that the Riemannian distance is an alternative to Euclidean distance for measuring image similarity since the RiemStega method obtained better results than the StegaStamp that uses L2 distance.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In International conference on machine learning, pages 214–223. PMLR, 2017. 3

[2] Vincent Arsigny, Pierre Fillard, Xavier Pennec, and Nicholas Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, 56(2):411–421, 2006. 1

[3] Shumeet Baluja. Hiding images in plain sight: Deep steganography. Advances in neural information processing systems, 30, 2017. 1

[4] Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. arXiv preprint arXiv:1810.00760, 2018. 4

[5] Mahbuba Begum and Mohammad Shorif Uddin. Digital image watermarking techniques: a review. Information, 11(2):110, 2020. 2

[6] Tu Bui, Shruti Agarwal, Ning Yu, and John Collomosse. Rosteals: Robust steganography using autoencoder latent space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 933–942, 2023. 2, 5

[7] Rudrasis Chakraborty, Jose Bouza, Jonathan H Manton, and Baba C Vemuri. Manifoldnet: A deep neural network for manifold-valued data with applications. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(2):799–810, 2020. 2

[8] Anoop Cherian, Suvrit Sra, Arindam Banerjee, and Nikolaos Papanikolopoulos. Jensen-bregman logdet divergence with application to efficient similarity search for covariance matrices. IEEE transactions on pattern analysis and machine intelligence, 35(9):2161–2174, 2012. 1

[9] Aniana Cruz, Gabriel Pires, and Urbano J Nunes. Spatial filtering based on riemannian distance to improve the generalization of errp classification. Neurocomputing, 470:236–246, 2022. 2

[10] Telmo Cunha., Luiz Schirmer., João Marcos., and Nuno Gonçalves. Noise simulation for the improvement of training deep neural network for printer-proof steganography. In Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods - ICPRAM, pages 179–186. INSTICC, SciTePress, 2024. 2, 3

[11] Harm Derksen. Error-correcting codes and b/sub h/-sequences. IEEE Transactions on Information Theory, 50(3):476–485, 2004. 7

[12] Xintao Duan, Nao Liu, Mengxiao Gou, Wenxin Wang, and Chuan Qin. Steganocnn: Image steganography with generalization ability based on convolutional neural network. Entropy, 22(10):1140, 2020. 2

[13] Han Fang, Zhaoyang Jia, Zehua Ma, Ee-Chien Chang, and Weiming Zhang. Pimog: An effective screen-shooting noise-layer simulation for deep-learning-based watermarking network. In Proceedings of the 30th ACM international conference on multimedia, pages 2267–2275, 2022. 5

[14] Duc Fehr, Anoop Cherian, Ravishankar Sivalingam, Sam Nickolay, Vassilios Morellas, and Nikolaos Papanikolopoulos. Compact covariance descriptors in 3d point clouds for object recognition. In 2012 IEEE international conference on robotics and automation, pages 1793–1798. IEEE, 2012. 2

[15] Ruili Feng, Deli Zhao, and Zheng-Jun Zha. Understanding noise injection in gans. In International Conference on Machine Learning, pages 3284–3293. PMLR, 2021. 3

[16] Pierre Fernandez, Alexandre Sablayrolles, Teddy Furon, Hervé Jégou, and Matthijs Douze. Watermarking images in self-supervised latent spaces. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3054–3058. IEEE, 2022. 2, 5

[17] Zilin Gao, Qilong Wang, Bingbing Zhang, Qinghua Hu, and Peihua Li. Temporal-attentive covariance pooling networks for video recognition. Advances in Neural Information Processing Systems, 34:13587–13598, 2021. 2

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30, 2017. 4

[19] Jiangtao Huang, Ting Luo, Li Li, Gaobo Yang, Haiyong Xu, and Chin-Chen Chang. Arwgan: Attention-guided robust image watermarking model based on gan. IEEE Transactions on Instrumentation and Measurement, 72:1–17, 2023. 5

[20] Zhiwu Huang and Luc Van Gool. A riemannian network for spd matrix learning. In Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017. 2

[21] Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In Proceedings of the 1st ACM international conference on Multimedia information retrieval, pages 39–43, 2008. 4

[22] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017. 7

[23] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. Advances in neural information processing systems, 28, 2015. 7

[24] John M Lee. Riemannian manifolds: an introduction to curvature, volume 176. Springer Science & Business Media, 2006. 3

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pages 740–755. Springer, 2014. 4

[26] Rui Ma, Mengxi Guo, Yi Hou, Fan Yang, Yuan Li, Huizhu Jia, and Xiaodong Xie. Towards blind watermarking: Combining invertible and non-invertible mechanisms. In Proceedings of the 30th ACM International Conference on Multimedia, pages 1532–1542, 2022. 5

[27] Maher Moakher. A differential geometric approach to the geometric mean of symmetric positive-definite matrices. SIAM

Journal on Matrix Analysis and Applications, 26(3):735–747, 2005. 2, 3

[28] Michael M Nordstrøm, Mads Larsen, Janusz Sierakowski, and Mikkel Bille Stegmann. The imm face database-an annotated dataset of 240 face images. 2004. 4

[29] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. International Journal of computer vision, 66:41–66, 2006. 1

[30] Rafia Rahim, Shahroz Nadeem, et al. End-to-end trained cnn encoder-decoder networks for image steganography. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pages 0–0, 2018. 2

[31] De Rosal Igantius Moses Setiadi. Psnr vs ssim: imperceptibility quality assessment for image steganography. Multimedia Tools and Applications, 80(6):8423–8444, 2021. 5

[32] Farhad Shadmand, Iurii Medvedev, and Nuno Gonçalves. Codeface: A deep learning printer-proof steganography for face portraits. IEEE Access, 9:167282–167291, 2021. 1

[33] Farhad Shadmand, Iurii Medvedev, Luiz Schirmer, João Marcos, and Nuno Gonçalves. Stampone: Addressing frequency balance in printer-proof steganography. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4367–4376, 2024. 2

[34] Yueyun Shang, Shunzhi Jiang, Dengpan Ye, and Jiaqing Huang. Enhancing the security of deep learning steganography via adversarial examples. Mathematics, 8(9):1446, 2020. 2

[35] Nandhini Subramanian, Omar Elharrouss, Somaya Al-Maadeed, and Ahmed Bouridane. Image steganography: A review of the recent advances. IEEE access, 9:23409–23423, 2021. 2

[36] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2117–2126, 2020. 2, 4, 5

[37] Weixuan Tang, Bin Li, Shunquan Tan, Mauro Barni, and Jiwu Huang. Cnn-based adversarial embedding for image steganography. IEEE Transactions on Information Forensics and Security, 14(8):2074–2087, 2019. 2

[38] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9, pages 589–600. Springer, 2006. 1, 2, 3

[39] Oncel Tuzel, Fatih Porikli, and Peter Meer. Pedestrian detection via classification on riemannian manifolds. IEEE transactions on pattern analysis and machine intelligence, 30(10):1713–1727, 2008. 3

[40] Rui Wang, Xiao-Jun Wu, and Josef Kittler. Symnet: A simple symmetric positive definite manifold deep learning method for image set classification. IEEE Transactions on Neural Networks and Learning Systems, 33(5):2208–2222, 2021. 2, 3

[41] Rui Wang, Xiao-Jun Wu, Tianyang Xu, Cong Hu, and Josef Kittler. U-spdnet: An spd manifold learning-based neural network for visual classification. Neural Networks, 161:382–396, 2023. 3

[42] Chaoning Zhang, Philipp Benz, Adil Karjauv, Geng Sun, and In So Kweon. Udh: Universal deep hiding for steganography, watermarking, and light field messaging. Advances in Neural Information Processing Systems, 33:10223–10234, 2020. 2

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 586–595, 2018. 3

[44] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In Proceedings of the European conference on computer vision (ECCV), pages 657–672, 2018. 2

[45] Jian Zou, Yue Zhang, Hongjian Liu, and Lifeng Ma. Monogenic features based single sample face recognition by kernel sparse representation on multiple riemannian manifolds. Neurocomputing, 504:82–98, 2022. 3