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

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

1. Model Description

The encoder is a standard U-Net with 5 convolutional layers for feature learning at different scales and downsampling, followed by 5 layers for upsampling the learned features, and a final convolutional layer to condense the number of channels back to 3 (see Figure 1). Skip-connections between each downsampling layer and the corresponding upsampling layer to avoid gradient-fading issues. The input is a 400×400×3 tensor representing an RGB image, and a 100-bit string as a message. The bit string is converted into an image using a fully-connected layer, followed by an upsampling operation. The result is concatenated into the image tensor, resulting in a 400×400×6 input tensor, fed to the U-Net, resulting in a residual image added to the input cover image.

The decoder is a standard CNN with 7 layers for feature learning, ending in 2 fully-connected layers to combine said features and output the bit string representing the message (see Figure 2). The decoder may be prepended by an STN network to learn and fix spatial distortions introduced by camera motion.

2. Printed Images

Images are printed on common office A4 paper sheets with sizes of 3×3cm, 5×5cm, and 10×10cm. In order to automatically detect the images, each one was augmented by a magenta border 5 pixels wide on all sides. Figure 3 shows samples of printed images in three sizes.

3. Ablation Studies

We assess the effect of our proposed encoder loss function by running an ablation study where we measure the impact of each term in the encoded image quality, see Table 1 for results. Additionally, we compare these results with our proposed loss with 60% and 100% of the residual added to the container images, to measure the impact of the said residual on image quality. Finally, we show the results of RoSteALS [1] and SSL [2] methods for reference. We can see in Table 1 that our complete loss with 60% of the residual (RiemStega60 column) presents the best results in terms of image quality, followed by our loss with 100% of the residual (RiemStega column). Compared to our model with full residual, RoSteALS obtains comparable results for all metrics, while SSL Watermarking comes close regarding PSNR.

Regarding individual loss terms, the LPIPS term naturally presents the best LPIPS score compared to the L2 and $\mathbf{R}_{loss}$ terms. $\mathbf{R}_{loss}$ obtain better PSNR and SSIM compared with L2. Using only the $\mathbf{R}_{loss}$ term presents PSNR results comparable to StegaStamp while lacking in terms of LPIPS and SSIM. Both L2 and LPIPS terms present SSIM scores comparable to StegaStamp and RoSteALS, but lack in terms of PSNR. These results indicate that the $\mathbf{R}_{loss}$ term leads to less noisy images. In contrast, the LPIPS term leads to better perceptual results besides higher structural similarity between the encoded and cover images, justifying the combination of these terms in our loss function.

References

[1] 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. 1, 3

[2] 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. 1, 3

*anianabrito@isr.uc.pt

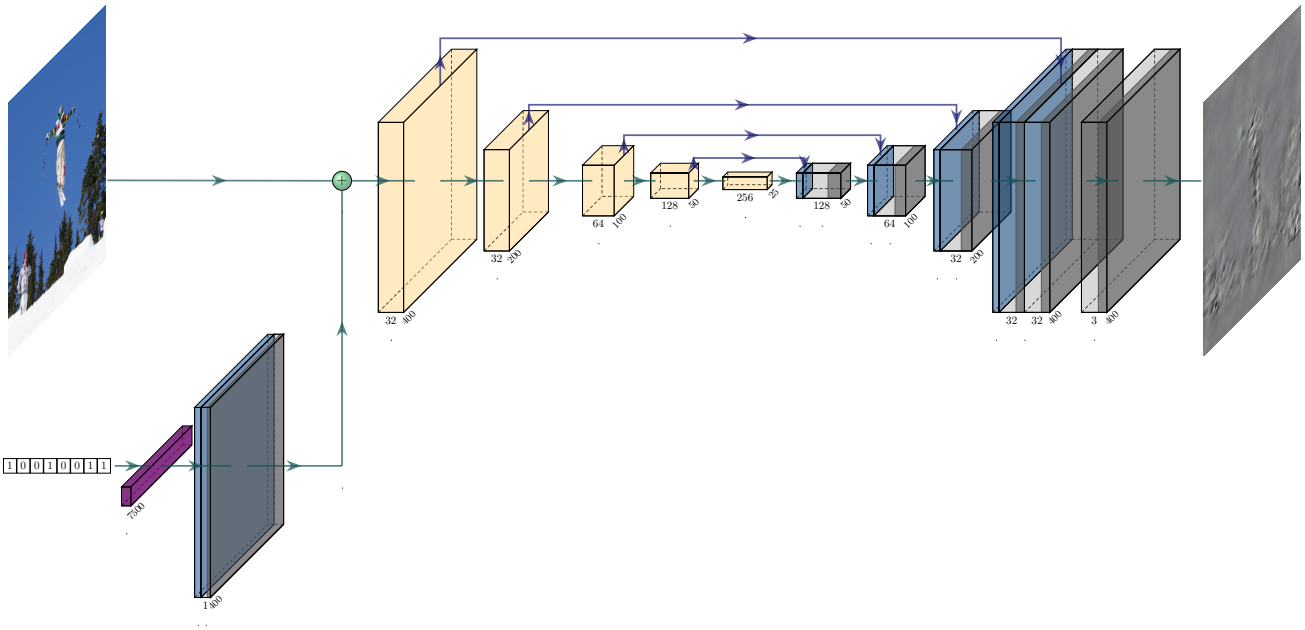

Figure 1. Encoder architecture used in our work. The model receives a 400×400×3 image and a 100-bit message. The message is converted to floating-point, fed to a dense layer with 7500 nodes, reshaped to 50×50×3, and upsampled to match the image size. Finally, this tensor is concatenated to form the input to a U-Net (400×400×6 tensor). The output of this U-Net is a residual image, shown on the right, which is added to the container image.

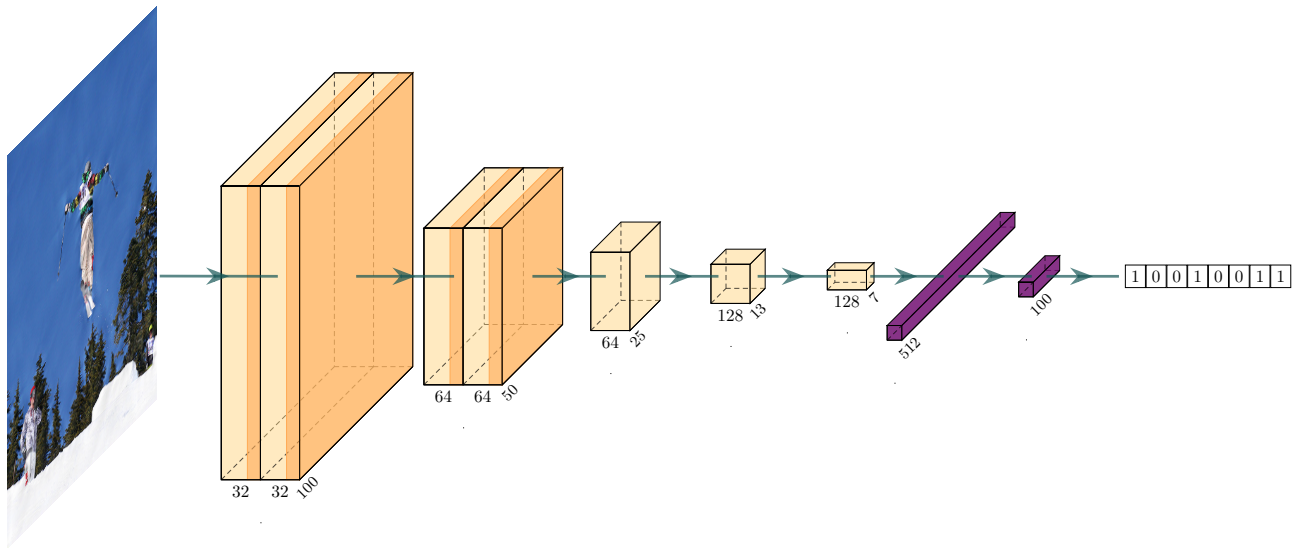

Figure 2. Decoder architecture employed in our work. The model receives a potentially encoded, 400×400×3 image, which is resized to 200×200×3 and, optionally passed to a Spatial-transformer Network (not illustrated here), where spatial transformations are mitigated. Afterwards this transformed image is fed to a series of convolutional layers, and finally, two dense layers, which result in a bit array with the decoded message.

Figure 3. Samples of encoded images using RiemStega printed with size 10×10cm, 5×5cm, and 3×3cm in A4 paper sheets with magenta borders for detection.

Table 1. The SSIM, PSNR, and LPIPS of 500 randomly selected images. The first three columns after Metric ($\mathbf{R}_{loss}$, L2 Loss, and LPIPS Loss) show the metric values for our model using these loss terms individually. The next columns, RiemStega and RiemStega60, show the results of our loss with 100% and 60% of the residual image applied. Finally, the last columns show the results of RoSteALS [1] and SSL [2] methods.

Metric	$\mathbf{R}_{loss}$	L2 Loss	LPIPS Loss	RiemStega	RiemStega60	StegaStamp	RoSteALS	SSL
SSIM ↑	0,873	0,905	0,926	0,949	**0,979**	0,894	0,937	0,895
PSNR ↑	28,503	25,943	25,574	30,031	**34,387**	28,470	30,360	32,620
LPIPS ↓	0,080	0,189	0,030	0,024	**0,013**	0,029	0,027	0,050