StampOne: Addressing Frequency Balance in Printer-proof Steganography (Supplementary Material)

Farhad Shadmand
ISR-UC[1]
farhad.shadmand@isr.uc.pt

Iurii Medvedev
ISR-UC[1]
iurii.medvedev@isr.uc.pt

Luiz Schirmer
ISR-UC[1], Unisinos[2]
luizschirmer@unisinos.br

João Marcos
ISR-UC[1]
joao.marcos@isr.uc.pt

Nuno Gonçalves
ISR-UC[1]
nunogon@deec.uc.pt

[1] Institute of Systems and Robotics, University of Coimbra, Portugal
[2] University of the Sinos River Valley Rio de Janeiro, Brazil

In the Supplementary Material, we start our goals of introducing StampOne as Steganography and watermark models. Then we compare the U-Shape networks mentioned in the paper. Subsequently, we delve into the intricacies of the Message Preparation Network, detailing the step-by-step process that led to the final network architecture. We then expand upon our experimentation, testing the decoder's performance with additional camera sensors and datasets. Additionally, we discuss the computation of spectral density and its division into low and high frequencies for each image. Code is available at https://github.com/farhadsh1992/StampOne.git

1. Application of StampOne

Our primary objective in introducing this model is twofold. Firstly, we aim to implement a stamp as a security layer to safeguard the originality of ID documents and enhance brand protection. This ensures the encoded image (stamp) remains indistinguishable from the original image. Secondly, we seek to address methods aimed at fabricating fake documents, rather than compromising the authenticity of a document. Additionally, we ensure that the stamp remains resistant to changes that occur over time, such as surface scratches on the encoded image. To achieve the first objective, we leverage watermarking techniques, while the second objective aligns with the goals of steganography. Our approach is motivated by previous researches [4, 12, 13], which were adopted to advance the field of steganography, Therefore, we consider StampOne as a steganography technique.

Table 1. Performance of StampOnes with various U−Shape structures. M1 incorporates all details of StegaStamp except for our highlighted dimensional preprocessing model.

Methods	**(A) Encoded images quality** SSIM (⇑)	LPIPS (⇓)	ColorHisto (⇓)
AttentionVNet	**0.98 ± 0.00002**	**1.25 ± 0.4**	**5.38 ± 4.9**
VNet	0.97 ± 0.00007	2.2 ± 0.3	5.8 ± 4.6
LeViTUnet	0.97 ± 0.00008	4.8 ± 2.8	7.8 ± 5.2
ResNetUnet	0.97 ± 0.00009	2.1 ± 1.6	6.5 ± 4.0
UNetPlus	0.96 ± 0.00007	2.74 ± 2.38	6.30 ± 4.07
AttentionUnet	0.95 ± 0.0001	2.8 ± 1.3	6.4 ± 4.4
SwinUnet	0.95 ± 0.0004	3.8 ± 1.01	7.8 ± 6.4
EfficientB0Unet	0.93 ± 0.0004	4.1 ± 2.5	8.7 ± 5.6
Non-robust (M1)	0.92 ± 0.001623	1.04± 1.69	2.80 ± 60.8

2. StampOne with different U−Shape networks

As discussed in the paper, StampOne was implemented with various networks, including UNet [10], VNet [11], Eff-UNet [2], LeViT-UNet [16], ResUNet [15], Swin-UNet [5], Attention-UNet [9], Attention-VNet, and UNet++ [17]. The perceptual quality of the encoded images was compared using LPIPS and Color Histogram [1], as presented in Table 1. The resulting encoded images from various structural configurations are depicted in Figures 8, 9, and 10. Optimal performance in both decoder and encoder functions was observed when employing the Attention-VNet and UNetPlus architectures.

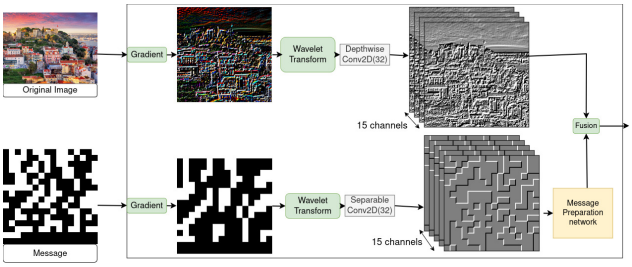

Figure 1. In our preprocessing stage, Original image and message are passed from gradient operation and wavelet transform. This Figure shows the output results of these two operations.

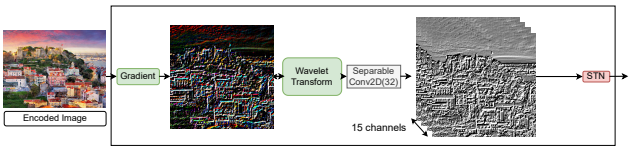

Figure 2. After applying the gradient operation and wavelet transform in our preprocessing stage, the encoded image is then directed to the decoder. This figure illustrates the output results of these two operations on the encoded image.

Table 2. StampOne's Encoders Performance Metrics. M1 denotes a non-robust model constructed using two instances of AttentionVNet.

Methods	Encoded images quality SSIM (⇑)	LPIPS (⇓)	ColorHisto (⇓)
GeLU	**0.98 ± 0.00002**	**1.89 ± 0.08**	**5.8 ± 4.9**
Snake	0.97 ± 0.0001	3.16 ± 0.2	8.7 ± 7.52
Tanh	0.97 ± 0.00007	2.24 ± 0.3	**5.8 ± 4.6**
ReLU	0.96 ± 0.0001	4.06 ± 0.5	7.4 ± 5.1

3. Message Preparation Network (MPN)

During preprocessing, the inputs of the encoder (2D binary message and original image) and decoder (encoded image) are converted to the Fast Frequency domain by a gradient operator and wavelet transformations. The outputs of these two transformations are depicted in Figures 1 and 2. Then the transformed message is passed through the Message Preparation Network (MPN) and finally concatenated with the transformed image (the highlighted high frequency domains of the original image).

We developed and assessed four distinct Message Preparation Networks (MPN), in Figure 3. A visual representation of the output results from the four different MPN is presented in Figure 4. MPN (D) has the best performance among others. It significantly improves the decoder and the encoder performances. It includes two branches, 1D Convolution, and a Dense layer. These two branches are connected to each other by a self-attention block, that is borrowed from

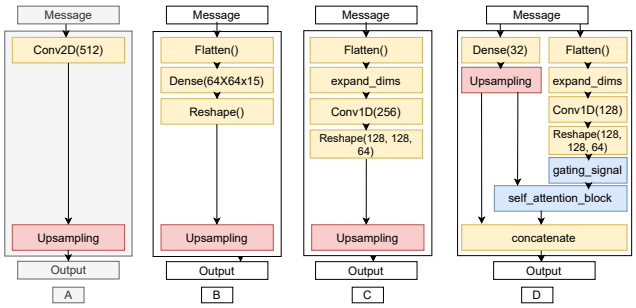

Figure 3. We compared four different Message Preparation Networks (MPNs) for incorporating messages into images: (A): This is the simplest approach, but it's only effective for digital steganography. With (B) implementation, we reached printed-proof, but it is not stable for every U-shape structure. (C) and (D): These two models provide stable results for both digital and printed steganography. However, model (D) offers better encoded image quality while maintaining stability.

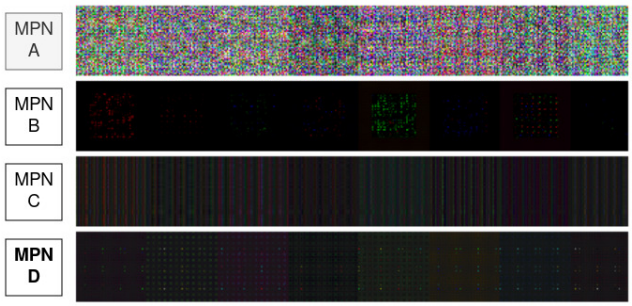

Figure 4. The output results of four different MPNs.

the AttentionUNet model [9], and a concatenate layer. We depict the encoded images generated by applying these four MPNs and VNets in Figure 5.

Model (A) is memory-efficient and suitable for digital transfers (requiring approximately 4 GB of memory). However, it lacks robustness in decoding messages from printed encoded images. Model (B) requires substantial GPU VRAM (approximately 12 GB) but can be effectively trained using VNet and AttentionVNet architectures, accommodating various noises instead of noise wrapping. For printed stamps, model (C) is recommended due to its lightweight nature and stability with both printed and digital images. Various U-shape structures were utilized, enabling efficient decoding of printed images, albeit with a decrease in encoded image quality. MPN (D) exhibits robustness against various noises while maintaining acceptable perceptual quality in its encoded images.

Another critical factor influencing network performance is the choice of activation function for MPN. To investigate this, we conducted tests and comparisons using several activation functions including "Snake," "GeLU," "ReLU," and

Table 3. Bit accuracy (%) during decoding is evaluated under various types and levels of noise. For comparative analysis, we employed StampOne with the VNet network architecture. MPN, incorporating various activation functions, was utilized in this evaluation.

	JPEG (%)			Gaussian (Std 0 to 1)			Resolution (Pixel)		
Methods	70	60	50	0.08	0.06	0.04	(60 × 60)	(80 × 80)	(100 × 100)
GeLU	86	96	92	84	94	96	67	**99**	**100**
Snake	**100**	**100**	**100**	**88**	**96**	**99**	**72**	94	99
Tanh	70	76	85	80	92	97	45	88	98
ReLU	60	84	94	80	92	98	45	94	98

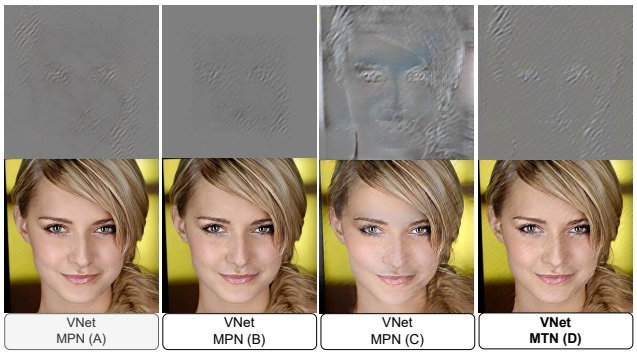

Figure 5. The result of encoded images with each MPNs. the four MPN's structures are showen and described precisely. The encoded images corresponding to each MPN are depicted in the Figure 3. The first row illustrates the residual image, while the second row displays the encoded images.

"Tanh." Snake [18] is a periodic activation function in training's forward and backward (gradient) directions, as shown below,

$$Snake_a(x) = x + \frac{1}{a} sin^2(ax) \tag{1}$$

where a is experimentally initialized to 0.5 according to the Snake publication [18]. At the end of the training, a is between 0.2 to 0.56, depending on the message size. The "GeLU" and "ReLU" activation functions are interpolated linear transformations. The "Tanh" activation extrapolates points as a constant function.

The comparative experiments regarding encoder performance are presented in Table 2. Additionally, decoder performance is outlined in Table 3. Based on our experimental findings, the activation functions "GeLU" and "Snake" exhibited the most effectiveness. In discussions concerning perceptual quality, "GeLU" emerges as a suitable option. However, in the context of robustness, "Snake" outperforms other activation functions, which we prioritize due to its paramount importance. Therefore, all StampOne models developed and discussed herein utilize the "Snake" activation function in the Message Preparation Network (MPN).

4. Decoder performance

Benchmark: In the experiment with printed images, we prepared two benchmark datasets. The first benchmark comprised a total of 40 randomly selected encoded images, sourced from both the BSDS500 dataset [8] and the Urban dataset [7]. The second benchmark included 40 face images extracted from the VGGFace2 dataset [6].

Camera: In our experiment to acquire images from printed images, we utilized two smartphones, Samsung S22 and HUAWEI P2, by capturing the video stream. The recorded videos were then transferred to a desktop workstation where the encoded images were extracted and subsequently decoded. In our work, to simulate realistic conditions of users holding in hands a smartphone, we define that successful decoding is achieved when the decoder correctly decodes at least 90% of the information in 20 frames from a video stream of encoded images.

Printouts: We printed encoded images on regular A4 paper, as illustrated in Figure 6. The sizes (width and height) of the encoded images range from 6×6 cm to 2×2 cm.

The results obtained from the Samsung S22 for the VGGFace2 face images dataset are reported in the manuscript. However, here, we provide comprehensive results for the decoder performance of StampOne utilizing the best structures (AttentionVNet and UNetPlus) for both benchmarks. These results are presented in Tables 4 and 5. StampOne demonstrates superior performance in extracting messages from encoded printed photos, except in cases where a weaker smartphone camera, such as the HUAWEI P20, is utilized and the photos are in a 2 configuration. In such scenarios, StegaStamp exhibits better performance (Table 5).

5. Gradient images in the frequency domain

Figure 7 (A) displays the spectral density of the original and encoded gradient images using StampOne and a non-robust steganography model. The difference in spectral density represents the degree of frequency bias in steganography GAN models, highlighting the effectiveness of our approach in mitigating this bias. In Figure 7 (B), we illustrate the transformation of the original and recovered

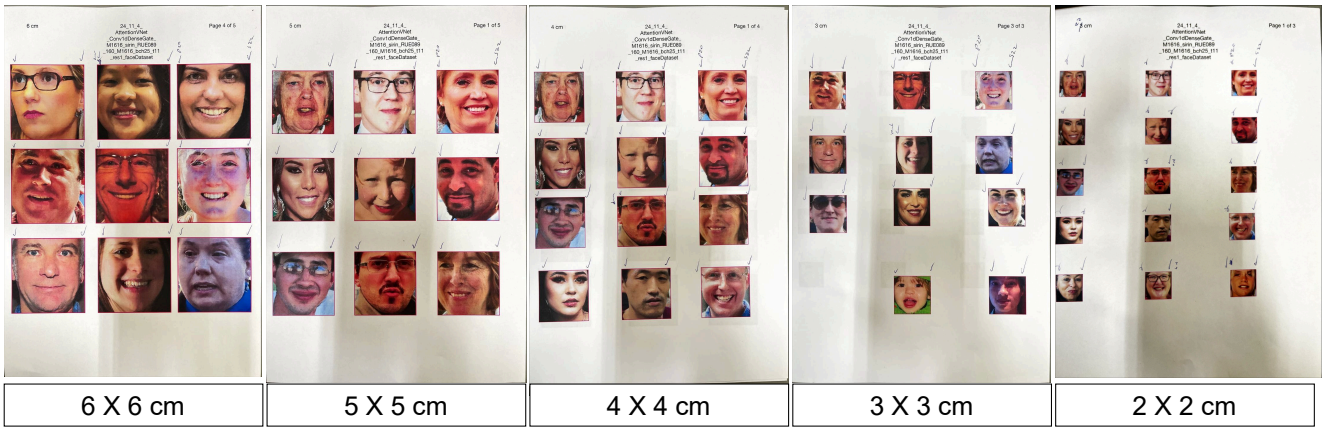

Figure 6. The samples of encoded images that are printed on paper for printer testing. The size of encoded images are from 6×6 cm to 2×2 cm on the A4 paper.

Table 4. The Bit accuracy of StampOne's decoder captured by the Samsung S22. M1 and M2 refer to StampOne models employing AttentionVNet and UNetPlus architectures, respectively. M3 denotes a non-robust model constructed using two instances of AttentionVNet.

Bit acc (%)-images captured with Samsung S22 Ultra smartphones.										
	VGGFace2 [6]					(B)**BSDS500 [8] and Urban [7]**				
Methods	6×6 cm	5×5 cm	4×4 cm	3×3 cm	2×2cm	6×6 cm	5×5 cm	4×4 cm	3×3 cm	2×2cm
StegaStamp [13]	78	72	70	65	48	82	82	82	82	72
Code Face [12]	55	55	50	38	15	0	0	0	0	0
StampOne (M1)	**100**	**100**	**100**	**95**	**62**	**90**	**90**	**90**	**80**	**80**
StampOne (M2)	88	85	72	63	43	72	72	52	45	12
Non-robust (M3)	0	0	0	0	0	0	0	0	0	0
RoSteALS [4]	0	0	0	0	0	0	0	0	0	0

messages into the frequency domain. This transformation aims to minimize the variance of spectral densities, allowing for effective message concealment in the specific frequency components of the images. Plot (C) in Figure 7 displays the spectral density of both the original and printed gradient images. The higher degree of similarity between the printed images from the Brother $L3270CDW$ printer and the original images indicates that this printer is more suitable than Epson $ET8500$ printer for robust steganography models.

6. Discrete cosine transform (DCT)

In order to enhance the high-frequency components of the decoder's output, we utilized the Discrete Cosine Transform (DCT) to convert both the original and recovered messages into the frequency domain. These transformed messages were then fed into the spectral discriminator for further processing. The specific calculations and details of the DCT are provided in this section.

An image represented by the three channels (red, green, and blue) combine to form a complete image. Each channel is represented by a 2D matrix with values ranging from 0 (the darkest) to 255 (the brightest). In addition to the sequential definition of a digital photo (RGB), the image can also be conceptualized as a composition of numerous sinusoidal gratings. These gratings are created by combining 2D sine and cosine wave series [14]. These sinusoidal gratings vary in terms of frequencies, phases, and orientations, contributing to the overall visual characteristics of the image. To serialize an image, it can be represented in terms of its sinusoidal gratings, allowing for its recreation from these components. A sinusoidal grating refers to a diffraction screen that possesses a sinusoidal groove profile. The grooves of the grating are symmetrical and lack a specific blazing direction. By decomposing an image into these sinusoidal gratings, we can capture its essential characteristics and enable its reconstruction using these components [14]. To represent an image's sinusoidal gratings, we utilize the Discrete Cosine Transform (DCT) to compute the frequency, phase, and amplitude specifications of the series. This information is stored in the k-space matrix of the image. The k-space matrix contains the DCT coefficients that represent the frequency, phase, and amplitude characteris-

Table 5. The Bit accuracy of StampOne's decoder captured by the HUAWEI P20. M1 and M2 refer to StampOne models employing AttentionVNet and UNetPlus architectures, respectively. M3 denotes a non-robust model constructed using two instances of AttentionVNet. In the printer test, encoded images are printed on A4 paper ranging from 6×6 cm to 2×2 cm (width×height).

Bit acc (%)-by using HUAWEI p20.										
	VGGFace2 [6]					**BSDS500 [8] and Urban [7]**				
Methods	6×6 cm	5×5 cm	4×4 cm	3×3 cm	2×2cm	6×6 cm	5×5 cm	4×4 cm	3×3 cm	2×2cm
StegaStamp [13]	62	57	52	52	**45**	78	78	75	75	**60**
Code Face [12]	35	32	25	12	0	0	0	0	0	0
StampOne (M1)	**100**	**100**	**98**	**92**	0	**90**	**90**	**88**	**88**	0
StampOne (M2)	75	75	70	57	0	65	65	20	5	0
Non-robust (M3)	0	0	0	0	0	0	0	0	0	0
RoSteALS [4]	0	0	0	0	0	0	0	0	0	0

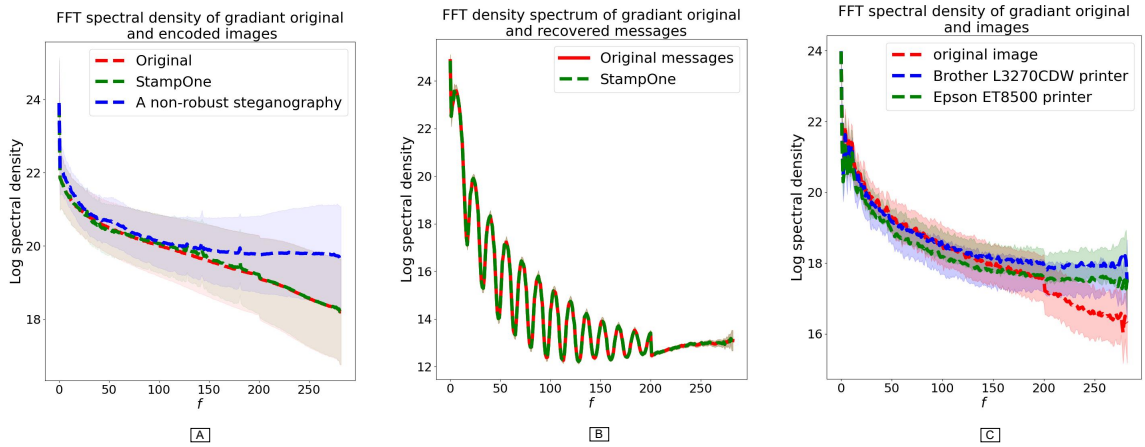

Figure 7. (A) presents the log spectral density of the original and encoded gradient images (with StampOne and a non-robust steganography algorithm). For non-robust steganography we utilize two VNet networks [11] for the encoder and decoder with StegaStamp loss functions and a discriminator without noise simulation. In contrast to the non-robust steganography GANs, the high frequency of StampOne's encode images is correlated to the original images. (B) demonstrates the log spectral density of the original and recovered gradient messages of StampOne decoder. The spectral density of 2D binary messages shows a sharper decline in high frequency; thus, the decoder's struggle with high frequency is more extreme. (C) displays the log spectral density of both the original and printed Gradient images. We utilized two separate office printers ($Brother\ L3270CDW$ and $Epson\ ET8500$) to print and scan 30 sample images. Printers add noise to the high frequency of images and raise the spectral density of high frequency. These plots for the gradient of the images are similar to the ones presented in the paper.

tics of the image.

By performing the two-dimensional DCT on an image, denoted as I($W \times H$), we can partition its spatial area into two categories: low frequency and high frequency components.

$$DCT(I_{ij}) = \alpha_p \alpha_q \sum_{j}^{H} \sum_{i}^{w} A_{ij} \times \cos\left(\frac{\pi(2i+1)p}{2H}\right) \times \cos\left(\frac{\pi(2j+1)q}{2W}\right) \tag{2}$$

where A_{ij} are the DCT coefficients. H and W are the height and width of the image, respectively. p and q are the vertical and horizontal frequencies of the image, respectively. α_q and α_p are, respectively,

$$\alpha_q = \begin{cases} 1/\sqrt{W} & \text{if } q = 0 \\ \sqrt{2/W} & \text{if } 1 \le q \le W - q \end{cases} \tag{3}$$

$$\alpha_p = \begin{cases} 1/\sqrt{H} & \text{if } p = 0 \\ \sqrt{2/H} & \text{if } 1 \le p \le H - q \end{cases} \tag{4}$$

The power spectral density is estimated by the squared magnitudes of the Fourier components, as follows,

$$SD(I_{ij}) = |DCT(I_{ij})^2| \tag{5}$$

Finally, $SD(I_{ij})$ of original and recovered messages are

passed through the spectral discriminator and the Euclidean distance of the outputs is minimized.

References

[1] Mahmoud Afifi, Marcus A Brubaker, and Michael S Brown. Histogan: Controlling colors of gan-generated and real images via color histograms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7941–7950, 2021. 1

[2] Bhakti Baheti, Shubham Innani, Suhas Gajre, and Sanjay Talbar. Eff-unet: A novel architecture for semantic segmentation in unstructured environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 358–359, 2020. 1

[3] Shumeet Baluja. Hiding images in plain sight: Deep steganography. *Advances in neural information processing systems*, 30, 2017. 7

[4] 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, 4, 5

[5] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021. 1

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018. 3, 4, 5

[7] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2015. 3, 4, 5

[8] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, pages 416–423, 2001. 3, 4, 5

[9] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018. 1, 2

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[11] Pegah Salehi and Abdolah Chalechale. Pix2pix-based stain-to-stain translation: A solution for robust stain normalization in histopathology images analysis. In *2020 International Conference on Machine Vision and Image Processing (MVIP)*, pages 1–7. IEEE, 2020. 1, 5

[12] Farhad Shadmand, Iurii Medvedev, and Nuno Gonçalves. Code face: A deep learning printer-proof steganography for face portraits. *IEEE Access*, 9:167282–167291, 2021. 1, 4, 5

[13] 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. 1, 4, 5

[14] Matthieu Urvoy, Dalila Goudia, and Florent Autrusseau. Perceptual dft watermarking with improved detection and robustness to geometrical distortions. *IEEE Transactions on Information Forensics and Security*, 9(7):1108–1119, 2014. 4

[15] Xiao Xiao, Shen Lian, Zhiming Luo, and Shaozi Li. Weighted res-unet for high-quality retina vessel segmentation. In *2018 9th international conference on information technology in medicine and education (ITME)*, pages 327–331. IEEE, 2018. 1

[16] Guoping Xu, Xuan Zhang, Yin Fang, Xinyu Cao, Wentao Liao, Xinwei He, and Xinglong Wu. Levit-unet: Make faster encoders with transformer for biomedical image segmentation. 1

[17] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018. 1

[18] Liu Ziyin, Tilman Hartwig, and Masahito Ueda. Neural networks fail to learn periodic functions and how to fix it. *Advances in Neural Information Processing Systems*, 33:1583–1594, 2020. 3

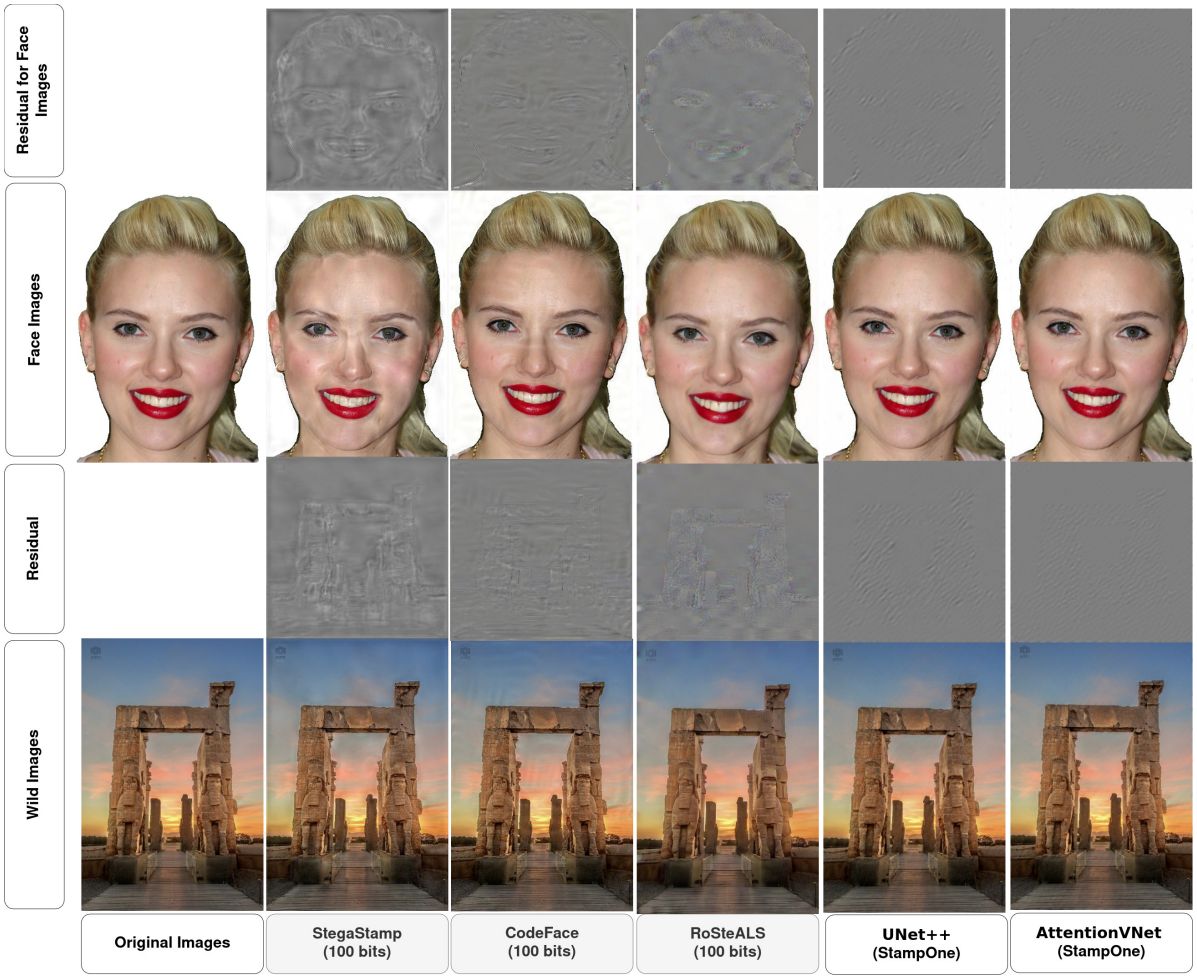

Figure 8. The results of applying our preprocessing model for various U-shape architectures are compared to those of the StegaStamp, Code Face, and the non-robust steganography models (such as [3]).The residual image, highlighting the difference between the original and encoded image, is displayed above each encoded image. While StegaStamp and Code Face use image inputs of $400 \times 400 \times 3$ resolution, other networks utilize a resolution of $256 \times 256 \times 3$. The message size for StegaStamp and Code Face is 100 bits, and for our structures is 256 bits. In the non-robust steganography model that is made from two VNet without our preprocessing and spectral discriminator,The residual image for the non-robust model shows significant color artifacts (indicating low-frequency modifications) compared to the other models whose residuals show minimal variations.

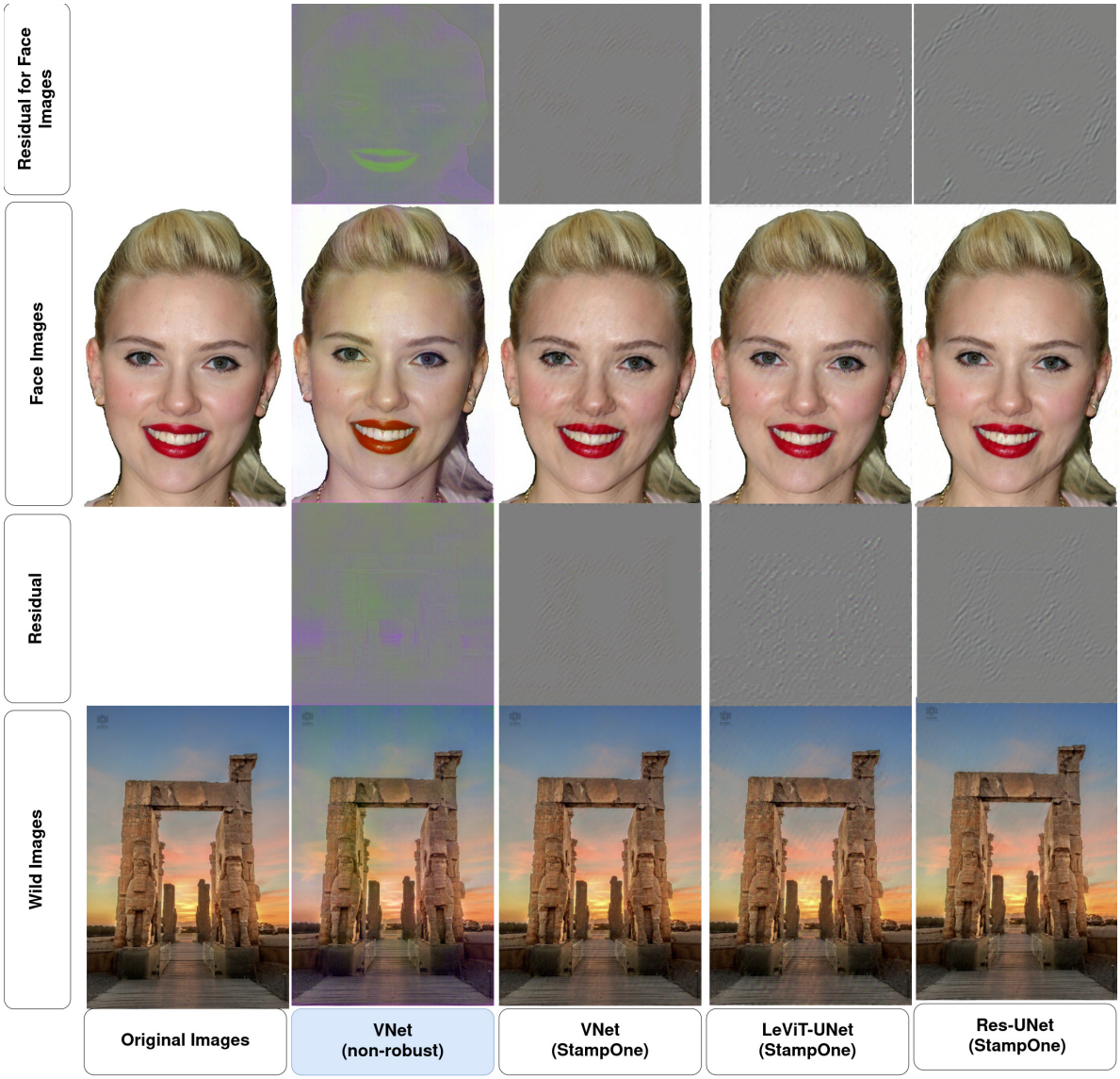

Figure 9. The results of applying StampOne preprocessing model for various U-shape architectures.

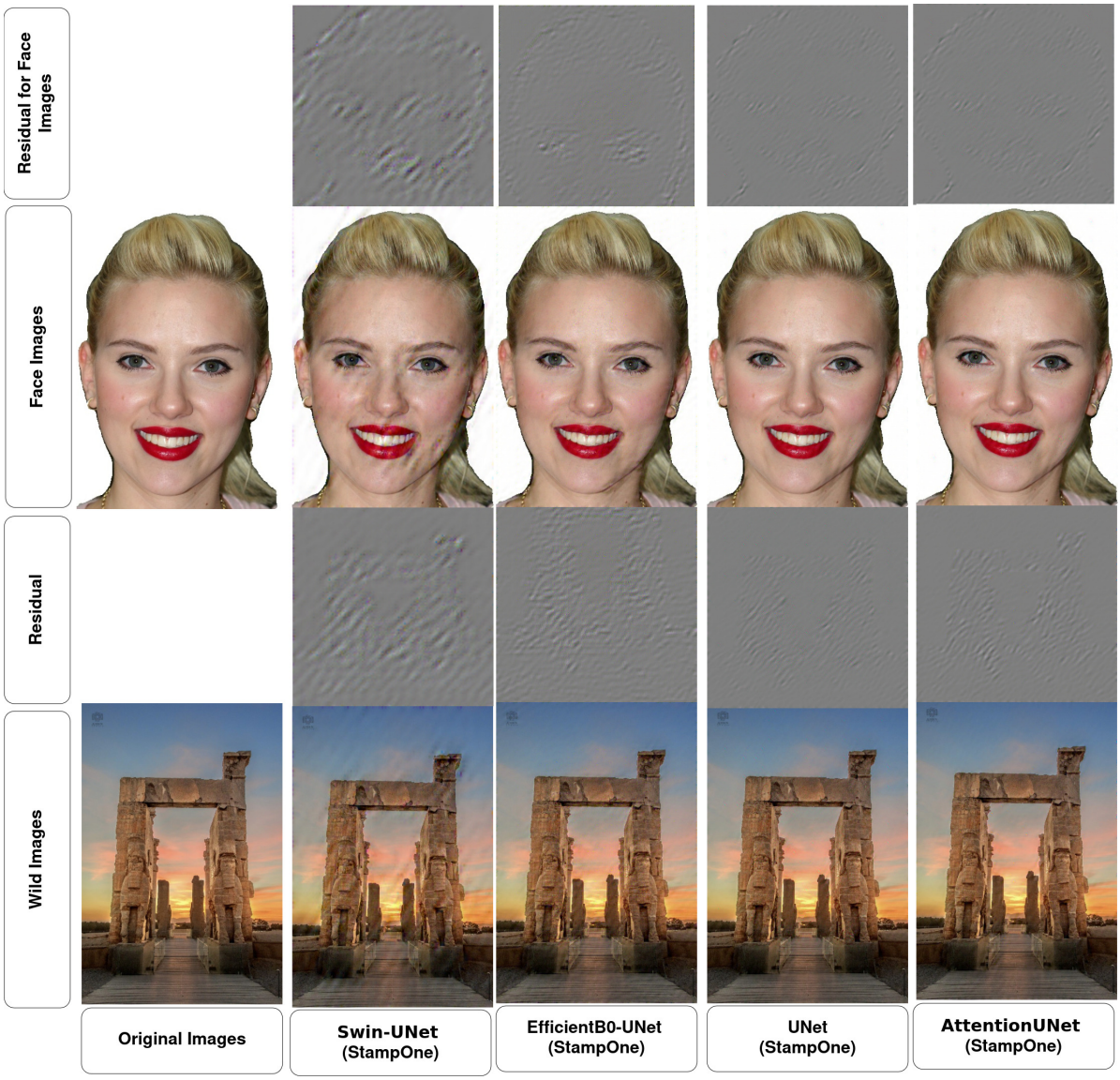

Figure 10. The results of applying StampOne preprocessing model for various U-shape architectures.