# Proactive Image Manipulation Detection
# – Supplementary material –

Vishal Asnani[1], Xi Yin[2], Tal Hassner[2], Sijia Liu[1], Xiaoming Liu[1]
[1]Michigan State University, [2]Meta AI
[1]{asnanivi, liusiji5, liuxm}@msu.edu, [2]{yinxi, thassner}@fb.com

## 1. Cross Encoder-Template Set Evaluation

Our framework encrypts a real image using a template from the template set. This encryption would aid in the image manipulation detection if the image is corrupted by any unseen GM. The framework is divided in two stages namely, image encryption and recovery of template where each stage works independently in inference. We therefore provide an ablation to study the performance using different encoder and template set, *i.e.*, we evaluate recovering ability of an encoder using a template set trained with different initialization seeds. The results are shown in Tab. 1. We observe that even though the template set and the encoder are initialized with different seeds, the performance of our framework doesn't vary much. This shows the stability of our framework even though the initialization seeds of both stages during training are different.

**Table 1.** Cross encoder-template set evaluation with different initialization seeds.

| Initialization seed | | Test GM Average precision (%) | | |
|---|---|---|---|---|
| Encoder | Template set | StarGAN | CycleGAN | GauGAN |
| 1 | 1 | 96.12 | 100 | 91.62 |
| | 2 | 94.65 | 100 | 91.15 |
| | 3 | 94.83 | 100 | 91.46 |
| 2 | 1 | 95.48 | 100 | 91.56 |
| | 2 | 95.54 | 100 | 90.85 |
| | 3 | 95.84 | 100 | 91.06 |
| 3 | 1 | 95.56 | 100 | 91.32 |
| | 2 | 95.62 | 100 | 91.42 |
| | 3 | 96.14 | 100 | 90.41 |

## 2. Template Strength

We provide the ablation for hyperparameter $m$ used to control the strength of the added template in Sec. 4.3. We observe that the performance is better if we increase the template strength. However, this comes at a trade-off with PSNR which declines if the template strength increases. This is also justified in Fig. 1 which shows the images with different strength of added template. The images become noisier as the template strength is increased. This is not desirable as there shouldn't be much distortion in the encrypted real image due to our added template. Therefore for our experiments, we select $30\%$ as the strength for the added template.

## 3. Implementation Details

**Image editing techniques** We use various image editing techniques in Sec. 4.2. All the techniques are applied after addition of our template. We provide the implementation details for all these techniques below:

1. Blur: We apply Gaussian blur to the image with 50% probability using $\sigma$ sampled from $[0, 3]$,
2. JPEG: We JPEG-compress the image with 50% probability images using Imaging Library (PIL), with quality sampled from Uniform$\{30, 31, ..., 100\}$.
3. Blur + JPEG (p): The image is possibly blurred and JPEG-compressed, each with probability p.
4. Resizing: We perform the training using $50\%$ of the images with $256 \times 256 \times 3$ resolution and rest with $128 \times 128 \times 3$ resolution images in CelebA-HQ dataset.
5. Crop: We randomly crop the images with 50% probability on each side with pixels sampled from $[0, 30]$. The images are resized to $128 \times 128 \times 3$ resolution.
6. Gaussian noise: We add Gaussian noise with zero mean and unit variance to the images with 50% probability.

**Network architecture** Fig. 2 shows the network architecture used in different experiments for our framework's evaluation. For our framework, our encoder has 2 stem convolution layers and 10 convolution blocks to recover the added template from encrypted real images. Each block comprises of convolution, batch normalization and ReLU activation.

In ablation experiments for Table 8, we use a classification network with the similar number of layers as our encoder. This is done to show the importance of recovering templates using encoder. This classification networks has 8 convolution blocks followed by three fully connected layers

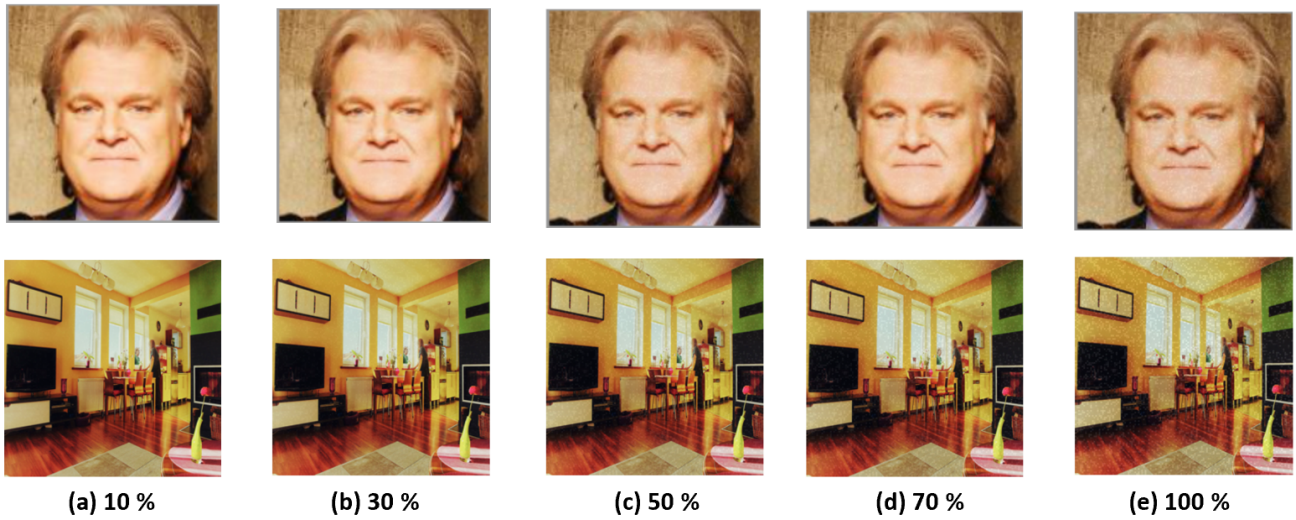


**Figure 1.** Visualization of input images with different template strength. As the template strength is increased, the images become noisier.

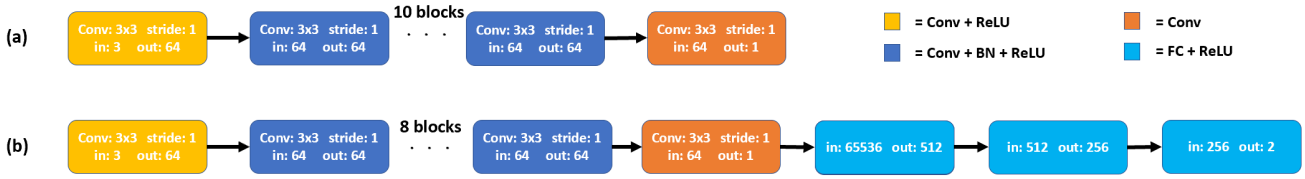


**Figure 2.** Network architecture for our (a) encoder (b) classifier network for image manipulation detection.

**Table 2.** List of GMs with their datasets and input image resolution used for evaluating our framework's generalization ability.

| GM | STGAN [7] | StarGAN [2] | CycleGAN [22] | GauGAN [11] | UNIT [8] | MUNIT [4] | StarGAN2 [3] | BicycleGAN [23] |
|---|---|---|---|---|---|---|---|---|
| Dataset | CelebA-HQ [6] | CelebA-HQ [6] | Facades [16] | COCO [1] | GTA2City [15] | Edges2Shoes [20, 21] | CelebA-HQ [6] | Facades [16] |
| Resolution | $128 \times 128 \times 3$ | $256 \times 256 \times 3$ | $256 \times 256 \times 3$ | $256 \times 256 \times 3$ | $512 \times 931 \times 3$ | $256 \times 512 \times 3$ | $256 \times 256 \times 3$ | $256 \times 256 \times 3$ |

| GM | CONT_Encoder [12] | SEAN [24] | ALAE [13] | Pix2Pix [5] | DualGAN [19] | CouncilGAN [10] | ESRGAN [18] | GANimation [14] |
|---|---|---|---|---|---|---|---|---|
| Dataset | Paris Street-View [10] | CelebA-HQ [6] | CelebA-HQ [6] | Facades [16] | Sketch-Photo [17] | CelebA [9] | CelebA [9] | CelebA [9] |
| Resolution | $64 \times 64 \times 3$ | $256 \times 256 \times 3$ | $256 \times 256 \times 3$ | $256 \times 256 \times 3$ | $256 \times 256 \times 3$ | $256 \times 256 \times 3$ | $128 \times 128 \times 3$ | $128 \times 128 \times 3$ |

with ReLU activation in between the layers. The network outputs 2 dimension logits used for image manipulation detection.

## 4. List of GMs

We use a variety of GMs to test the generalization ability of our framework. These GMs have varied network architectures and many of them are trained on different datasets. We summarize all the GMs in Tab. 2. We also provide visualization for different real image samples used in evaluating the performance for all these GMs in Fig. 3 - 18. We show the added template and the recovered templates in "gist_rainbow" cmap for better visualization and indicate the cosine similarity of the recovered template with the added template. As shown in Fig. 3 for training with STGAN, the encrypted real images have higher cosine similarity compared to their manipulated counterparts. However, during testing, the difference between the two cosine similarities decreases as shown in Fig. 4 - 18 for different GMs.

## 5. Dataset License Information

We use diverse datasets for our experiments which include face and non-face datasets. For face datasets, we use existing datasets including CelebA [9] and CelebA-HQ [6]. The CelebA dataset contains images entirely from the internet and has no associated IRB approval. The authors mention that the dataset is available for non-commercial research purposes only, which we strictly adhere to. We only use the database internally for our work and primarily for evaluation. CelebA-HQ consists images collected from the internet. Although there is no associated IRB approval, the authors assert in the dataset agreement that the dataset is only to be used for non-commercial research purposes, which we strictly adhere to.

We use some non-face datasets too for our experiments.

The Facades [16] dataset was collected at the Center for Machine Perception and is provided under Attribution-ShareAlike license. Edges2Shoes [20, 21] is a large shoe dataset consisting of images collected from https://www.zappos.com. The authors mention that this dataset is for academic, non-commercial use only. GTA2City [15] dataset consists of a large number of densely labelled frames extracted from computer games. The authors mention that the data is for research and educational use only. The sketch-photo [17] datset refers to the CUHK face sketch FERET database. The authors assert in the dataset agreement that the dataset is only to be used for noncommercial research purposes, which we strictly adhere to. Paris street-view [10] dataset contains images collected using google street view and is to be used for noncommercial research purposes.

## References


[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *CVPR*, 2018. 2

[2] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 2

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. StarGAN v2: Diverse image synthesis for multiple domains. In *CVPR*, 2020. 2

[4] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[6] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 2

[7] Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. STGAN: A unified selective transfer network for arbitrary image attribute editing. In *CVPR*, 2019. 2

[8] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NeurIPS*, 2017. 2

[9] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 2

[10] Ori Nizan and Ayellet Tal. Breaking the cycle - colleagues are all you need. In *CVPR*, 2020. 2, 3

[11] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. GauGAN: semantic image synthesis with spatially adaptive normalization. In *ACM*, 2019. 2

[12] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[13] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *CVPR*, 2020. 2

[14] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. GANimation: One-shot anatomically consistent facial animation. *International Journal of Computer Vision*, 128:698–713, 2020. 2

[15] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 2, 3

[16] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *GCPR*, 2013. 2, 3

[17] Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31:1955–1967, 2008. 2, 3

[18] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *CVPR*, 2021. 2

[19] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. DualGAN: Unsupervised dual learning for image-to-image translation. In *CVPR*, 2017. 2

[20] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 2, 3

[21] A. Yu and K. Grauman. Semantic jitter: Dense supervision for visual comparisons via synthetic images. In *ICCV*, 2017. 2, 3

[22] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2

[23] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017. 2

[24] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. SEAN: Image synthesis with semantic region-adaptive normalization. In *CVPR*, 2020. 2

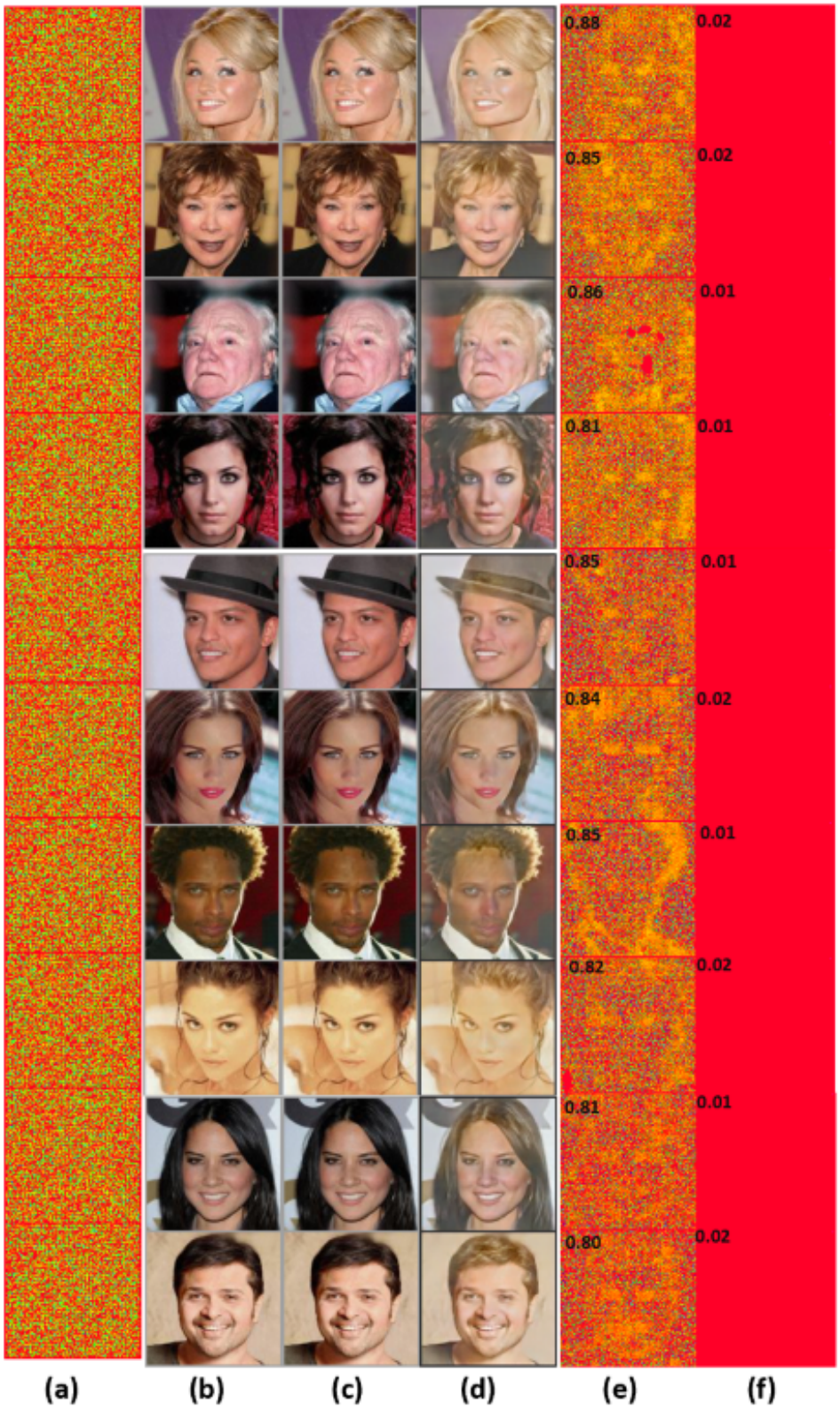


**Figure 3.** Visualization of samples used for GM STGAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

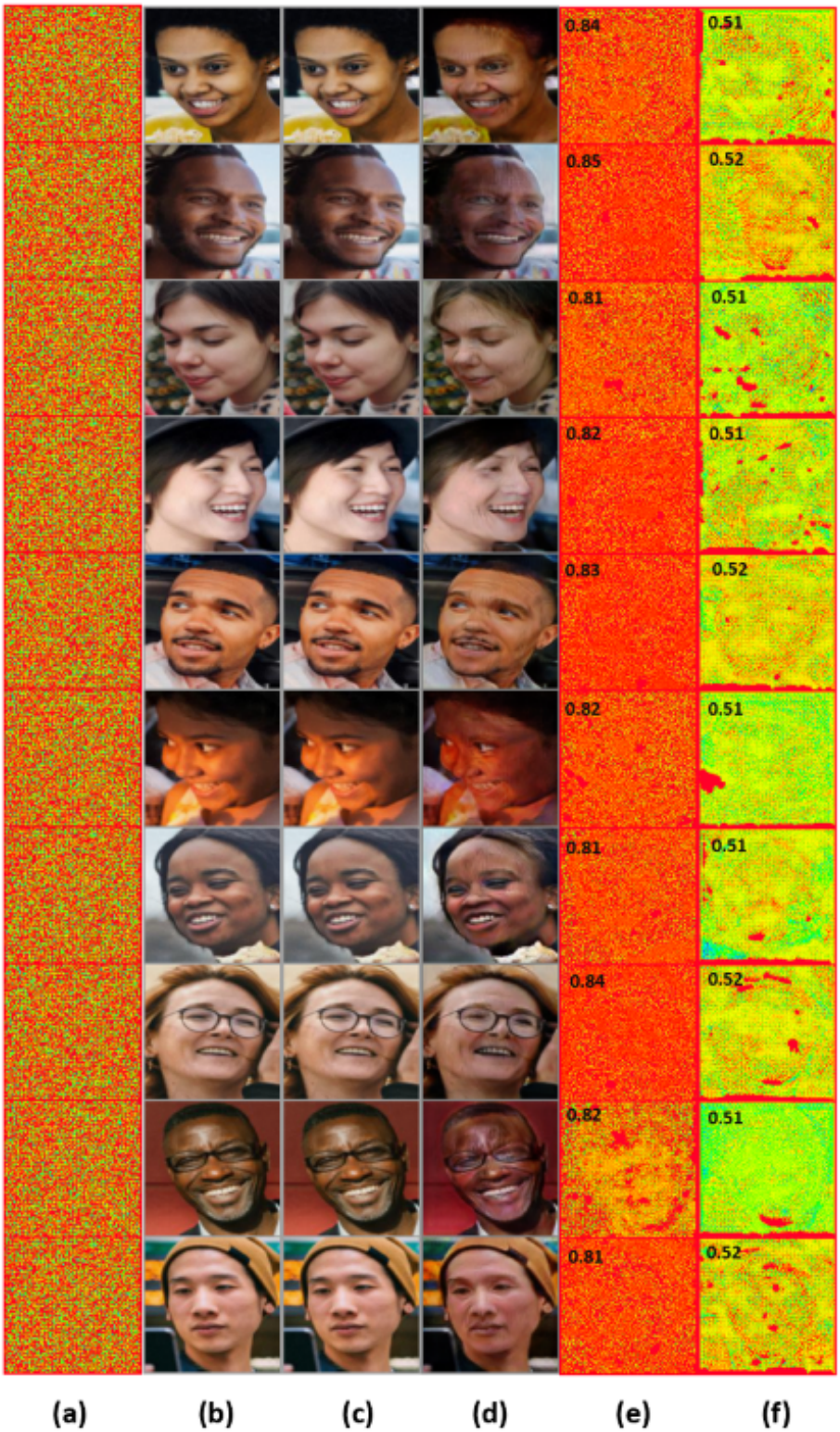


**Figure 4.** Visualization of samples used for GM StarGAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

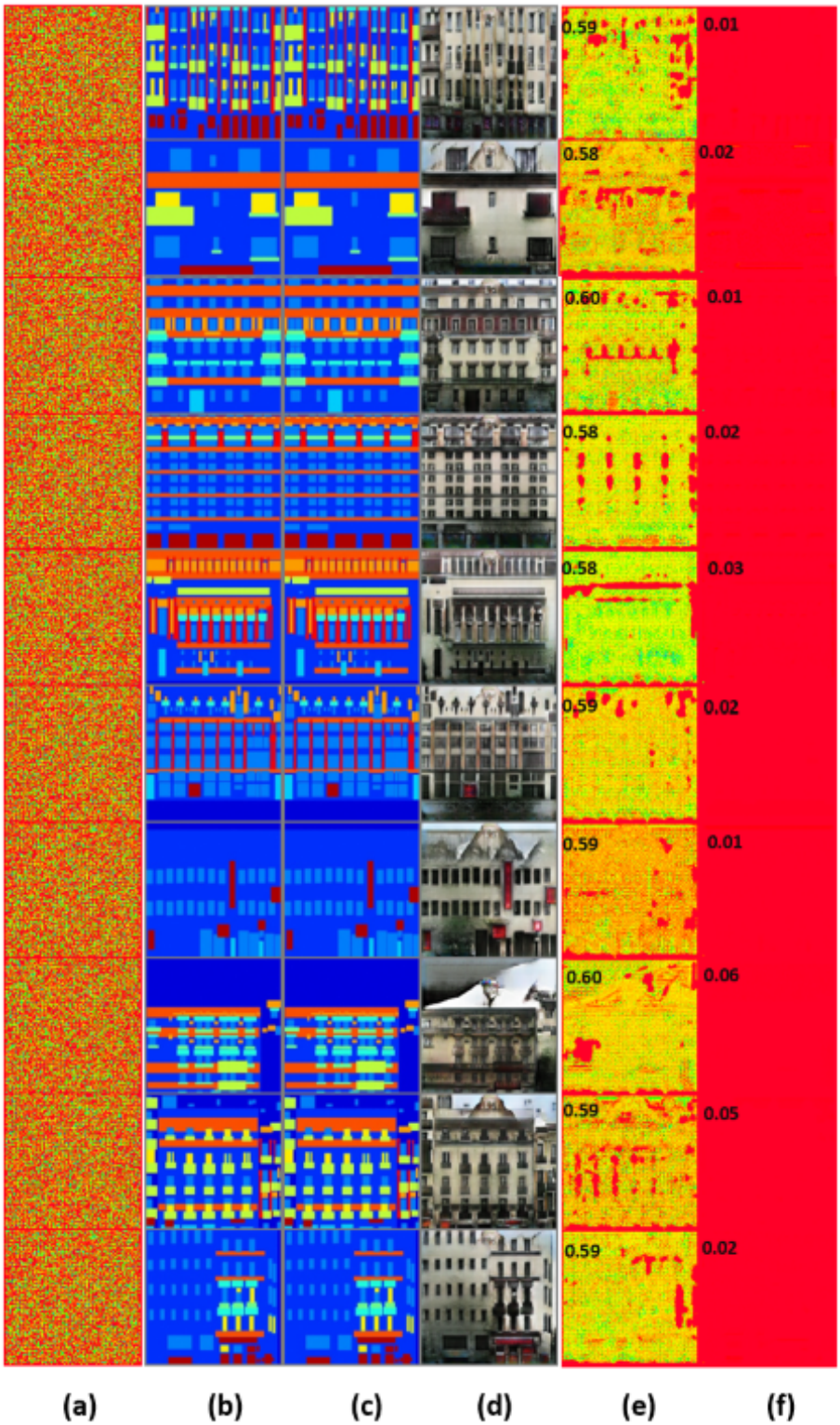


**Figure 5.** Visualization of samples used for GM CycleGAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

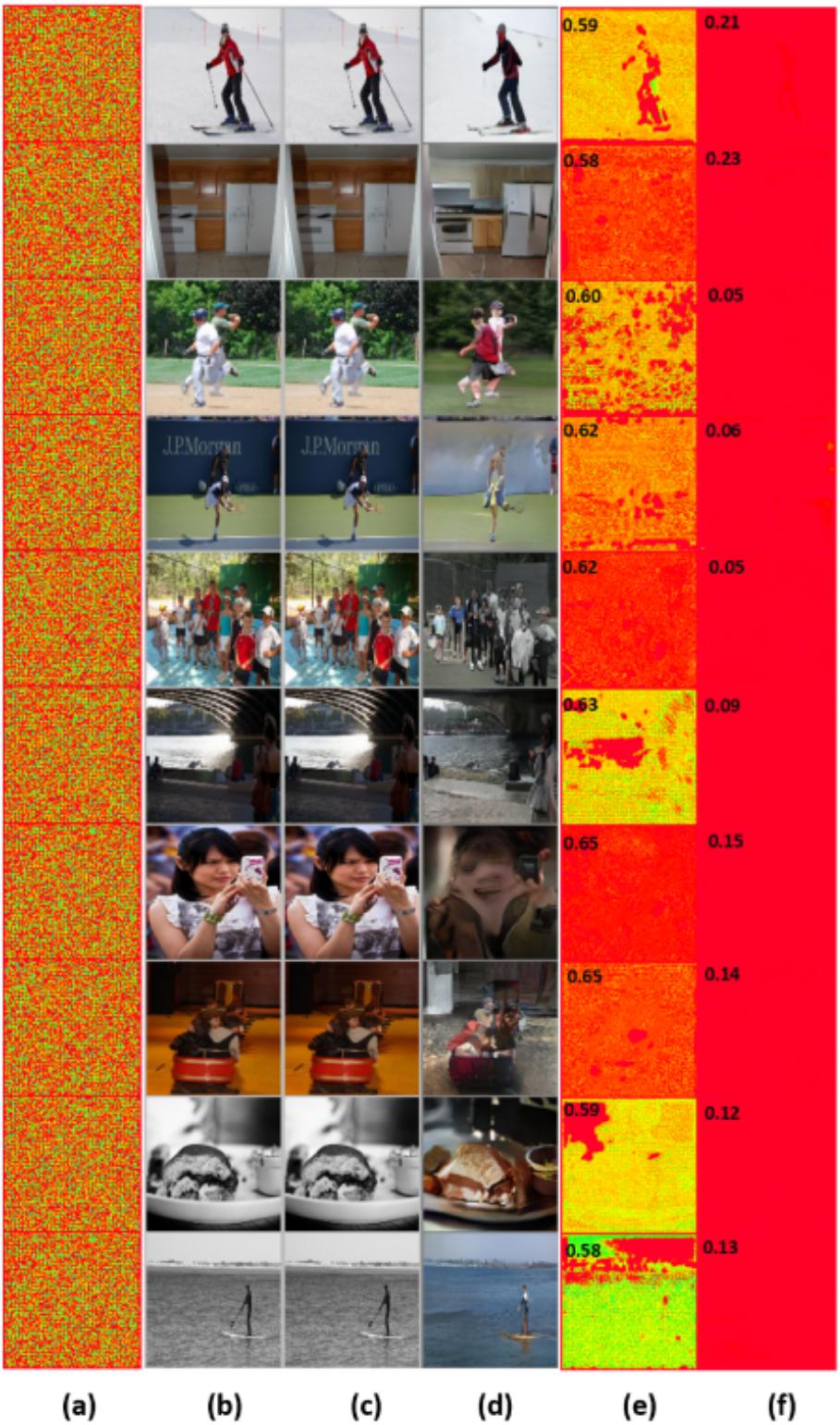


**Figure 6.** Visualization of samples used for GM GauGAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

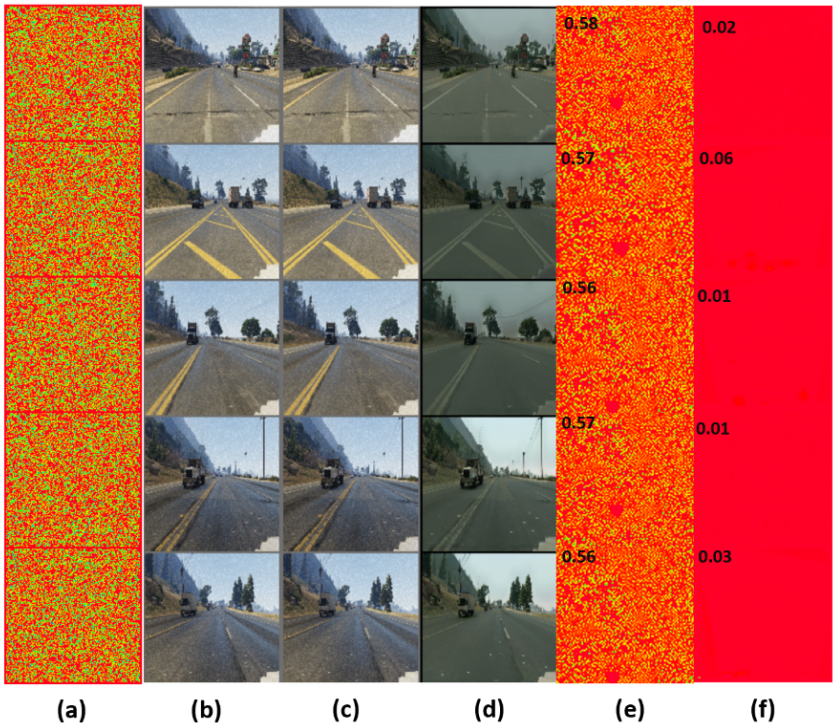


**Figure 7.** Visualization of samples used for GM UNIT; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

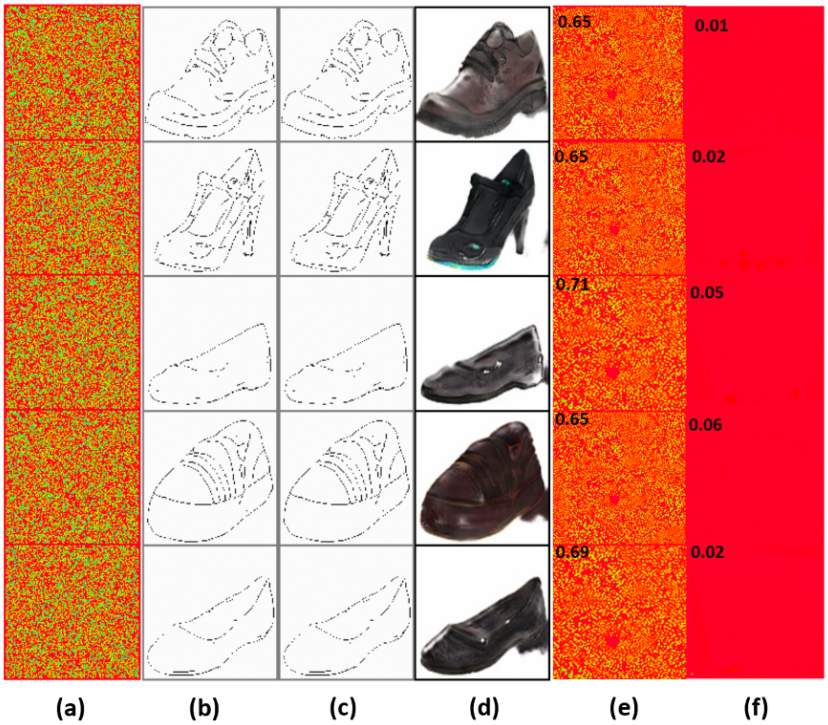


**Figure 8.** Visualization of samples used for GM MUNIT; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

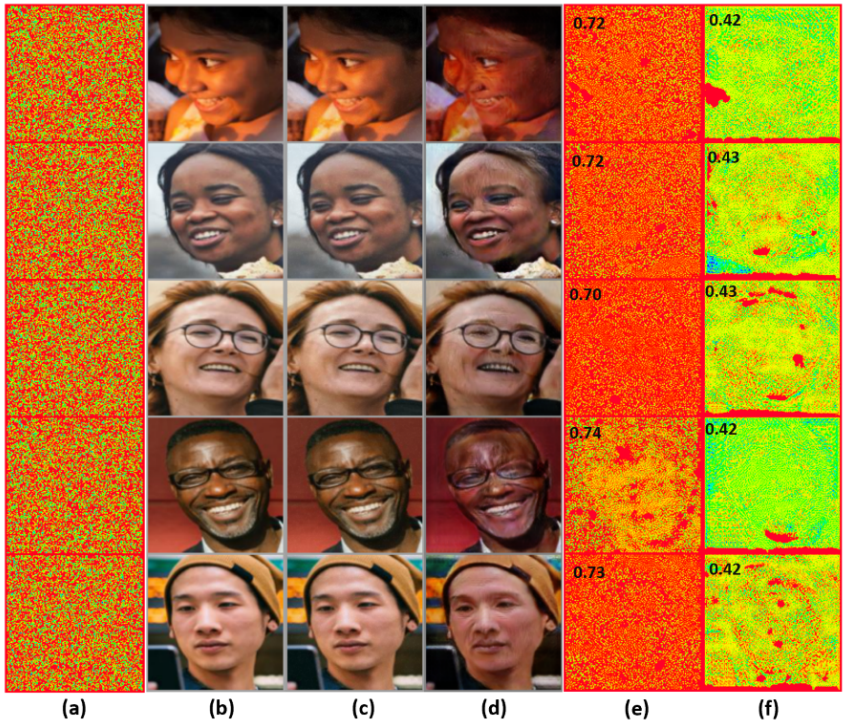


**Figure 9.** Visualization of samples used for GM StarGANv2; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

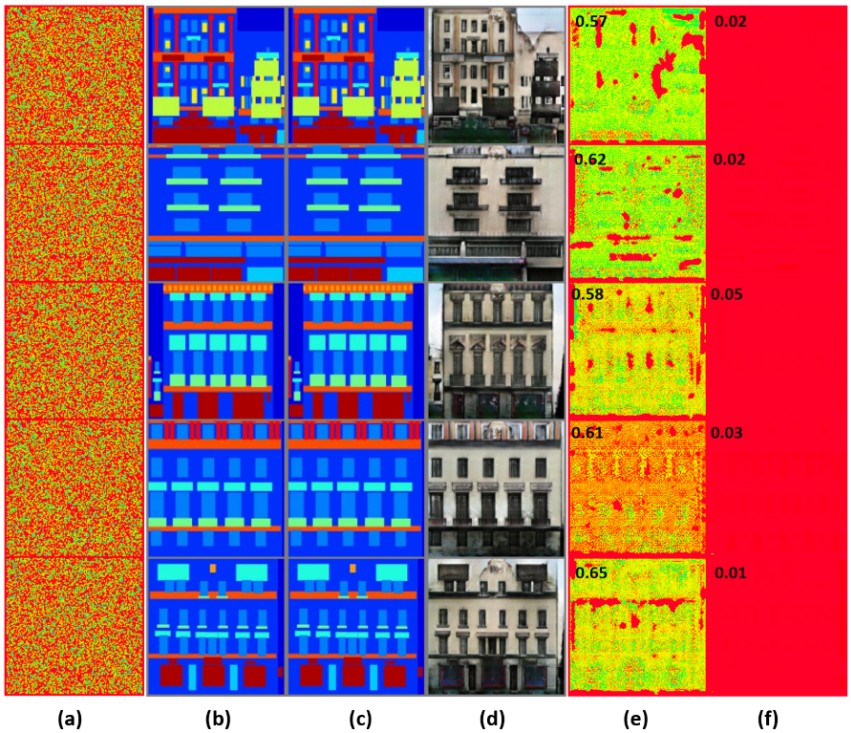


**Figure 10.** Visualization of samples used for GM BicycleGAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

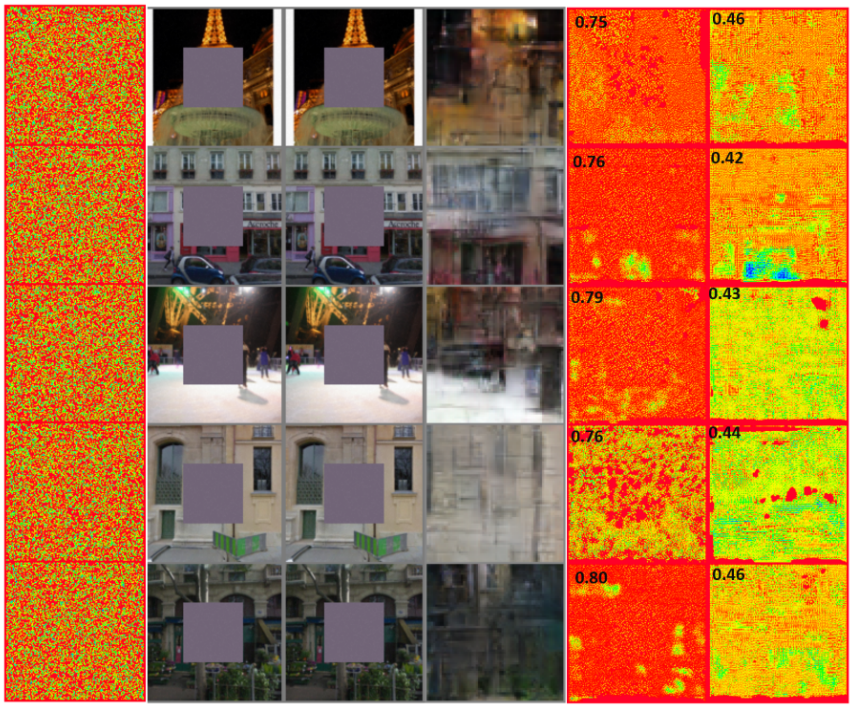


**Figure 11.** Visualization of samples used for GM CONT_Encoder; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

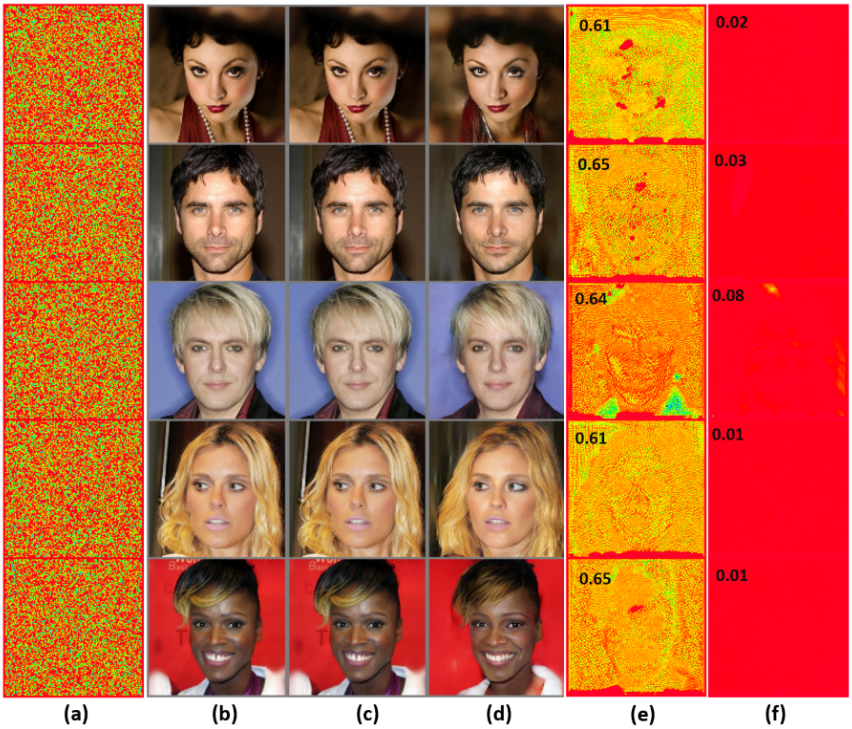


**Figure 12.** Visualization of samples used for GM SEAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

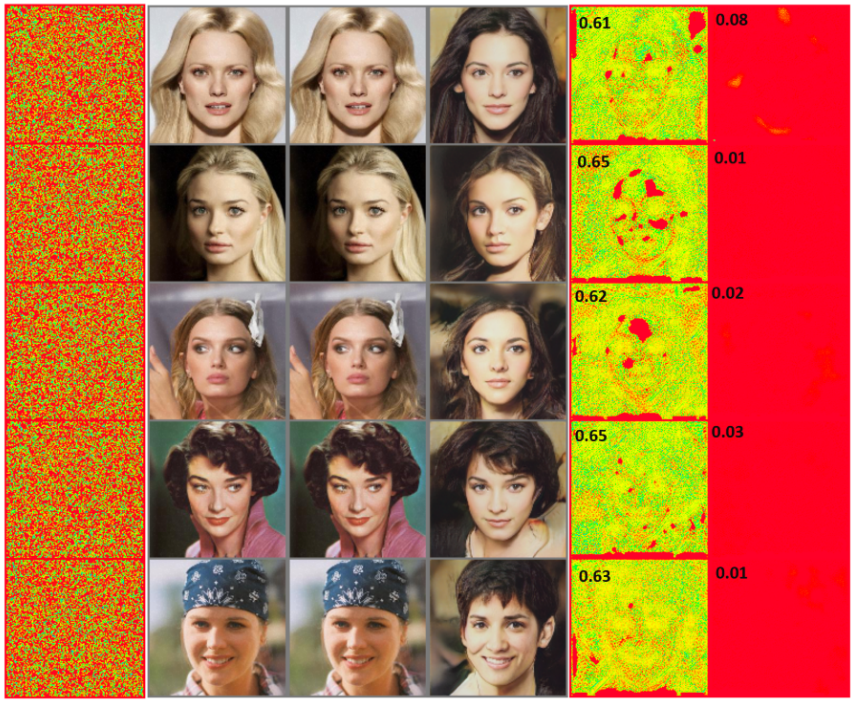


**Figure 13.** Visualization of samples used for GM ALAE; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

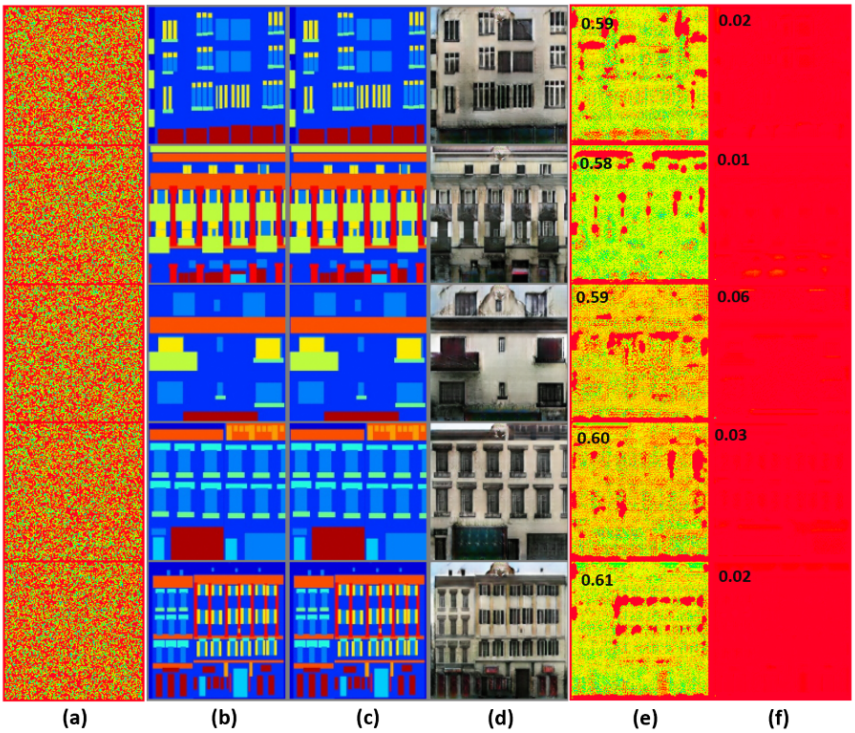


**Figure 14.** Visualization of samples used for GM Pix2Pix; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

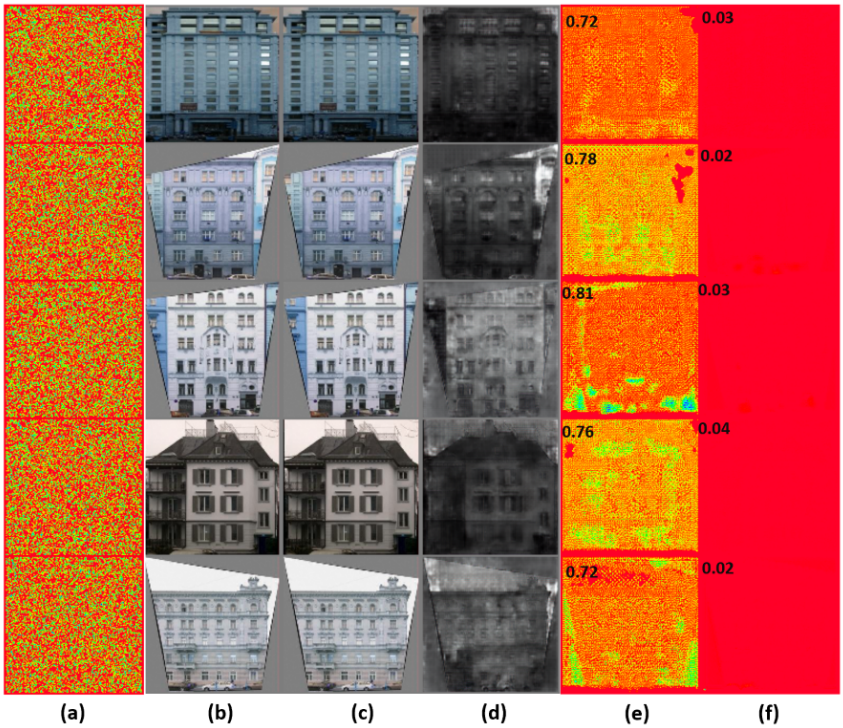


**Figure 15.** Visualization of samples used for GM DualGAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

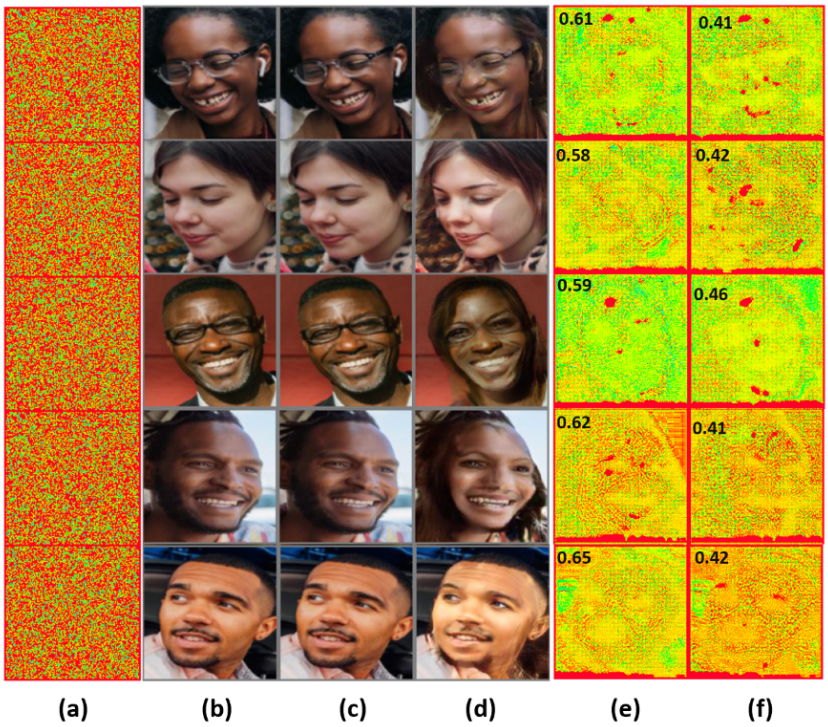


**Figure 16.** Visualization of samples used for GM CouncilGAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

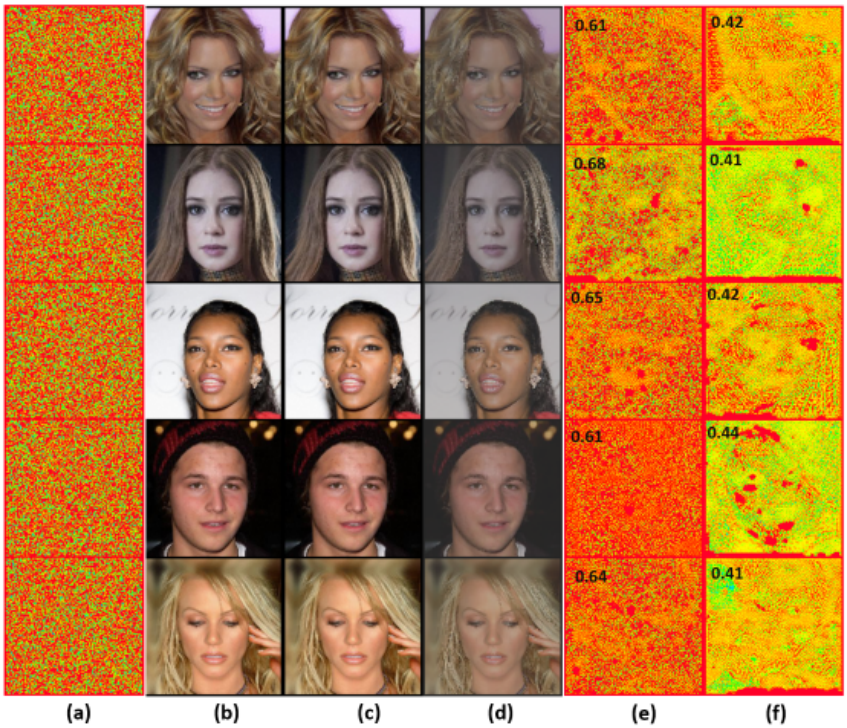


**Figure 17.** Visualization of samples used for GM ESRGAN; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.

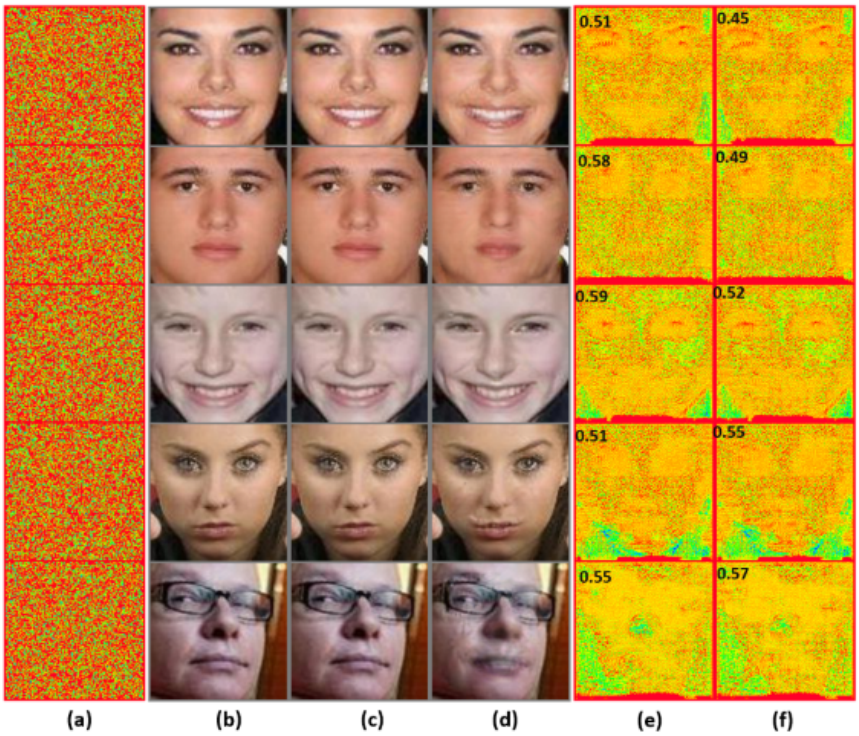


**Figure 18.** Visualization of samples used for GM GANimation; (a) added template, (b) real images, (c) encrypted real images after adding a template, (d) manipulated images output by a GM, (e) recovered template from (c), and (f) recovered template from (d). Top left corner in last two columns shows the cosine similarity of the recovered template with the added template.