Article

Not peer-reviewed version

Mutual Effects of Face-Swap Deepfakes and Digital Watermarking - A Region-Aware Study

Tomasz Walczyna * and Zbigniew Piotrowski

Posted Date: 3 September 2025

doi: 10.20944/preprints202509.0333.v1

Keywords: digital image watermarking; multimedia security; deepfake face swap; generative adversarial networks; invisible watermark; visible watermark; region-aware evaluation; non-local image edits; digital image forensics

Article

Mutual Effects of Face-Swap Deepfakes and Digital Watermarking - A Region-Aware Study

Tomasz Walczyna * and Zbigniew Piotrowski

Electronics and Telecommunications Faculty, Military University of Technology, 00-908 Warsaw, Poland

* Correspondence: tomasz.walczyna@wat.edu.pl

Highlights

What are the main findings?

- Region-aware evaluation across visible and invisible watermarks with tunable strength and six face-swap families shows that edits are non-local and non-monotonic - background changes introduced by generators degrade watermarks even far from the face, and retention does not vary linearly with embedding strength.
- A locality-preserving baseline bounds the minimal impact - architectures that better confine edits to the facial region, typically those with segmentation-weighted objectives, preserve background watermark signal more reliably than globally trained GAN pipelines.

What is the implication of the main finding?

- Classical robustness tests for watermarking are not sufficient on their own - evaluation should include generator-induced transformations from face swap and report region-wise metrics for face and background.
- Watermark strength and placement should be selected in an architecture-aware manner - in our sweeps, appropriately tuned invisible marks achieved higher background correlation under manipulation than visible overlays at comparable perceptual distortion.

Abstract

Face-swap is commonly assumed to act locally on the face region, which motivates placing watermarks away from the face to preserve the integrity of the face. We demonstrate that this assumption is violated in practice. Using a region-aware protocol with tunable-strength visible and invisible watermarks and six face-swap families, we quantify both identity transfer and watermark retention on the VGGFace2 dataset. First, edits are non-local - generators alter background statistics and degrade watermarks even far from the face, as measured by background-only PSNR and Pearson correlation relative to a locality-preserving baseline. Second, dependencies between watermark strength, identity transfer, and retention are non-monotonic and architecture-dependent. Methods that better confine edits to the face—typically those employing segmentation-weighted objectives—preserve background signal more reliably than globally trained GAN pipelines. At comparable perceptual distortion, invisible marks tuned to the background retain higher correlation with the background than visible overlays. These findings indicate that classical robustness tests are insufficient alone - watermark evaluation should report region-wise metrics and be strength- and architecture-aware.

Keywords: digital image watermarking; multimedia security; deepfake face swap; generative adversarial networks; invisible watermark; visible watermark; region-aware evaluation; non-local image edits; digital image forensics;

1. Introduction

Digital image processing and computer vision are increasingly underpinning sensing pipelines, in which images are captured, processed, and authenticated at scale. In this context, multimedia security - including watermarking - intersects with CV tasks such as detection and generative manipulation. Unlike classical perturbations used in robustness testing, face swap generators can induce structured, content-aware edits that propagate beyond the nominal region of interest. The potential misuse of such edits for disinformation or unauthorized redistribution poses a challenge to watermark-based intellectual property protection [1–6].

We study, in a controlled and region-aware manner, both directions of interaction: the resistance of watermarking to local face swap and the impact of watermarking on face swap effectiveness. The study compares two watermarking strategies (a visible overlay with variable opacity and a neural invisible watermark with tunable strength) across six representative face swap families. It includes a locality-preserving baseline that edits only the face region.

Beyond achieving this objective, our work provides four specific contributions for the multimedia security community:

- a two-sided, region-aware evaluation protocol that quantifies both identity transfer and watermark retention using ArcFace distance, Pearson correlation, and background-only PSNR, enabling comparisons across models and strengths;
- empirical evidence that generator edits are non-local and that the relationship between watermark strength, identity transfer, and retention is non-monotonic, challenging the common assumption that placing a mark away from the face suffices;
- an architecture-aware analysis showing that methods which better confine edits to the facial region - typically those leveraging segmentation-weighted objectives - preserve background watermark signal more reliably than globally trained GAN pipelines;
- practical guidance for robustness evaluation in sensing workflows, indicating when tuned invisible marks retain more background correlation than visible overlays at comparable perceptual impact.

Classical watermark robustness studies primarily evaluate compression, resampling, and noise [7–10], whereas face-swap research focuses on optimizing identity transfer and realism [11–16]. Prior works seldom couple both perspectives under a region-aware analysis that separates face and background and contrasts visible versus invisible watermarks with tunable strength [17]. Our study fills this gap by conducting controlled sweeps on VGGFace2 and establishing a locality baseline that bounds the minimal off-face impact. We focus on still images and face swap as the manipulation type, which aligns with common image-centric sensing scenarios.

To enable controlled comparisons, we implement parameterizable variants of the watermarking and face swap methods used in our experiments. Although other manipulation families (reenactment, lip animation, full-face synthesis) may also affect watermarking, we select face swap as the primary case study to establish a precise reference and a dedicated baseline for localized edits.

The following section details the watermarking and face swap methods, including our parameterizable implementations and the locality-preserving baseline, as well as the region-aware experimental protocol and metrics. We then report results for visible and invisible watermarking and discuss implications for robustness evaluation in sensing workflows.

2. Materials and Methods

2.1. Watermark

A digital watermark is identification information intentionally incorporated into an image, sound, video, or document that remains associated with the file during processing. Its basic functions include confirming the source or owner of the content, tracking its distribution, and, in some systems, user authorization. A watermark can be visible, such as a semi-transparent logo that discourages unauthorized use, or hidden, embedded in the spatial or frequency domain in a way that is invisible to the viewer but can be read after typical editing operations such as compression, scaling, or cropping [10,18].

The key features of a well-designed watermark include: robustness, imperceptibility to the end user, capacity to store additional bits, and security against counterfeiting. In practice, this requires a compromise: the more robust the mark, the greater the interference with the data and the potential deterioration in quality; the more discreet it is, the more difficult it is to ensure its readability after aggressive processing. In the case of publicly published content, a hybrid approach is often employed – a visible logo serves as a deterrent against simple copying. At the same time, a hidden identifier facilitates the enforcement of rights in the event of a dispute [19].

This analysis considers two extreme variants of watermarking – visible and invisible – as they represent two basic content protection strategies, directly noticeable or completely invisible to the end user. The research does not focus on preserving the semantic content encoded in the watermark, but on assessing its resistance to DeepFake-type modifications.

2.1.1. Visible Watermark

The analysis employed an explicit watermark in the form of a QR code spanning the entire frame. This solution ensures the uniform distribution of the mark's pixels in the image and eliminates the risk of omitting any area when assessing the marking's impact.

For clarity, a single example – a synthetic face image – is presented in three variants: (a) reference image without a watermark, (b) the difference between the image with a watermark and the reference image, and (c) a composition of both images with a selected level of transparency (Figure 1a–c). This presentation enables us to evaluate the degree to which an explicit watermark affects the image's details, even before DeepFake methods are applied.

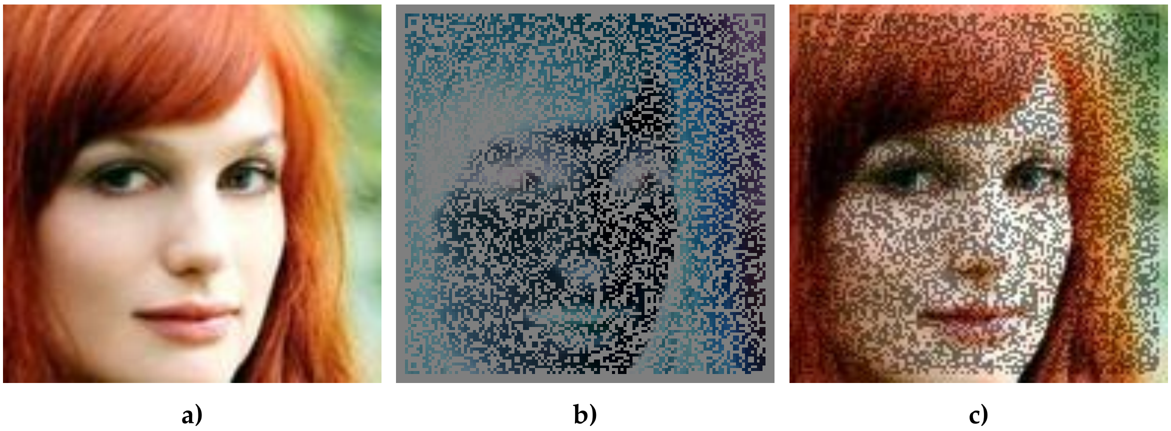
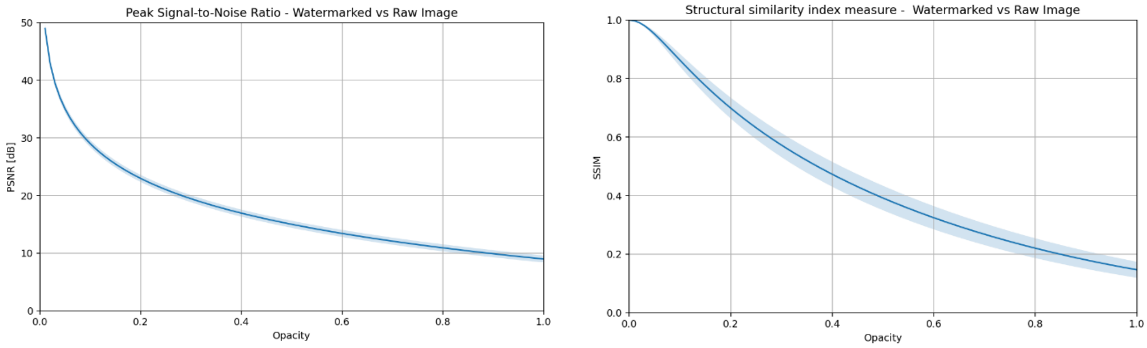


Figure 1. A single example of implemented explicit watermarking: a) image without a watermark, b) difference map (image with watermark minus image without watermark), c) image with watermark (50% opacity).

The impact of transparency on image distortion was determined using two commonly used quality metrics: PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) [20]. Figure 2 illustrates the dependence of these metrics on the opacity parameter, which ranges from 0 to 1, corresponding to the level of watermark visibility. The observed curves illustrate a decrease in image quality as the visibility of the mark increases. These graphs serve as a reference point in both subsequent chapters and the experimental part, where the impact of different transparency levels on the effectiveness of DeepFake methods will be analyzed.



a) **b)**

Figure 2. Impact of the transparency level of an explicit watermark on image distortion – comparison of an image with a watermark and a reference image: a) PSNR, b) SSIM.

2.1.2. Invisible Watermark

The second option analyzed is a hidden watermark embedded using a neural network. Its purpose is to remain completely invisible to the recipient while remaining resistant to typical image processing operations. Many methods of this type have been described in the literature; however, in most cases, it is not possible to precisely control the strength of the interference [7,8]. In studies focused solely on assessing the effectiveness of watermark reading or its impact on a selected task (e.g., classification) [9], such a limitation may be acceptable. However, in a broader analysis—especially when the goal is to generalize the results to an entire group of algorithms (in our case, local face replacement)—it can significantly complicate interpretation.

For this reason, a proprietary model has been developed that allows for smooth adjustment of the watermark signal amplitude – from virtually undetectable to deliberately visible. This allows for a precise examination of the relationship between the strength of the mark and its susceptibility to local modifications, such as DeepFakes.

The designed architecture is based on the classic encoder–decoder approach. The encoder receives an image, to which it matches a watermark, and a message to be embedded. The generated watermark, controlled by the "watermark strength" parameter, is then added to the original image. The resulting image can be manipulated in any way (e.g., face swap), and its degraded version is sent to the decoder, whose task is to recover the encoded message. The invisible watermark used here is purposefully a controllable test instrument rather than a proposed state-of-the-art algorithm. Its novelty for this paper lies in its practicality: the strength parameter is continuously tunable, which enables calibrated sweeps that isolate how watermark energy interacts with generator-induced transforms. This controllability is required to compare visible vs. invisible marks under identical experimental conditions and is not intended as a claim of algorithmic novelty in watermarking.

The encoder was built based on a modified U-Net architecture [21], equipped with FiLM (Feature-wise Linear Modulation) layers [22], which enable the entry of message information at different resolution levels. Residual connections have also been added [23] between successive U-Net levels, which improves gradient flow—a crucial aspect in architectures where part of the cost function is calculated only after the decoder. The encoder output is transformed by a tanh function (with a range of -1 to 1) and then scaled by the "watermark power" parameter (default: 0.1). The default value of the "watermark strength" parameter = 0.1 was adopted experimentally as the midpoint of the range [-1, 1] used in the training process. It ensured a clear yet moderate level of interference with the image, allowing for tests of both greater subtlety and higher visibility of the marker.

To increase the generality of the model and avoid situations where the network hides the watermark only in selected locations, a set of random perturbations was used during training: Gaussian noise, motion blur, Gaussian blur, brightness and contrast changes, resized crop, and random erasing. These were not intended to teach resistance to specific attacks (e.g., face swap), but to force the even distribution of the watermark throughout the image.

The decoder is based on the ResNet architecture [24], whose task is to reduce a tensor containing a degraded image to a vector representing the encoded message.

The learning process involved two primary components of the cost function. The first concerned the correctness of message reading by the decoder.:

$$L_{msg} = BCEWithLogitsLoss(z, m) \quad (1)$$

where z – output tensor from the decoder, m – coded message.

The second component was responsible for minimizing the visibility of the watermark. For this purpose, a combination of mean square error (MSE) and LPIPS metrics was used [25], better reflecting the difference between images as perceived by humans:

$$L_{vis} = MSE(\,\hat{x}, x\,) + 0.2 \cdot LPIPS(\,\hat{x}, x\,) \quad (2)$$

where $\hat{x}$ - image with watermark, x – original image.

All images were scaled to a resolution of 128 × 128 pixels. The adopted resolution of 128×128 pixels is lower than typical in practical applications. This limitation was due to the computational requirements associated with training multiple deepfake networks and a watermarking model in real time, given the available hardware resources. For the comparative analysis, maintaining consistent experimental conditions was more important than achieving absolute image resolution. The message length was set to 64 bits, generated randomly during training. VGGFace2 [26] was used as the dataset, containing photos of different people, which allowed for consistency with the rest of the experiments.

In the case of this model, the key indicator in the context of comparative analysis is not the fact that the message was read correctly (full effectiveness was achieved during training), but the impact of the "strength" parameter of the watermark on its actual visibility and level of interference with the image.

Figure 3 illustrates an example of an image in the default configuration (watermark strength = 0.1) along with its corresponding difference map. Figure 4 illustrates the impact of the "strength" parameter on PSNR and SSIM metrics compared to the original image [8].

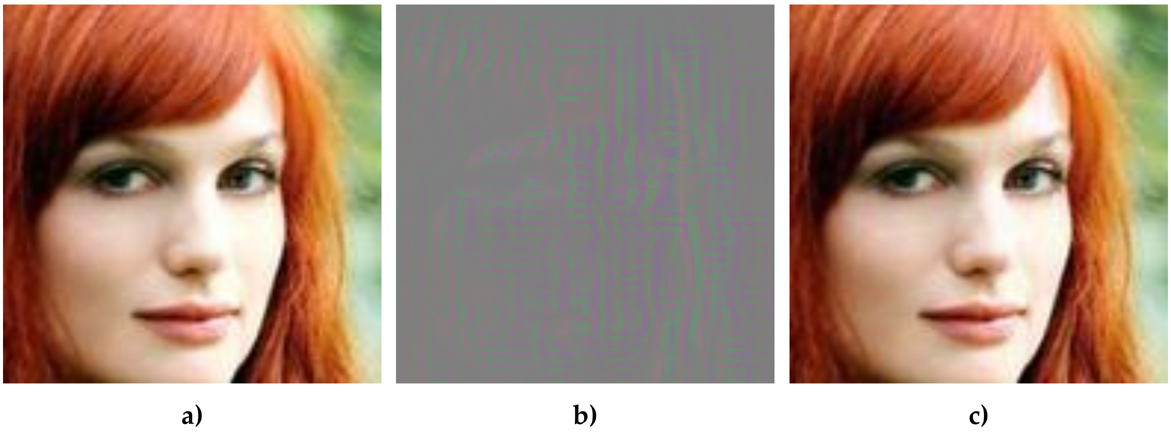
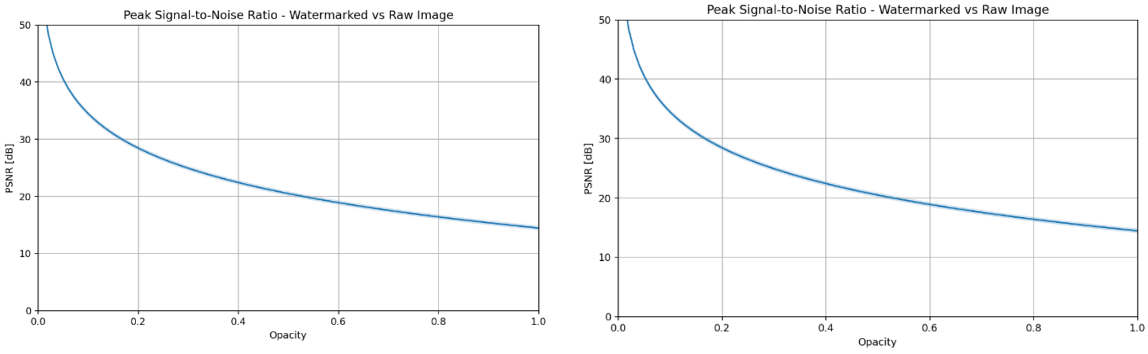


Figure 3. An example of implemented hidden watermarking: a) image without watermark, b) difference map, c) image with watermark (strength 0.1).



a) **b)**

Figure 4. The influence of the "strength" parameter of a hidden watermark on image distortion – comparison of an image with a watermark with the original image: a) PSNR, b) SSIM.

2.2. Face Swap

Face swap is a class of generative algorithms whose goal is to insert a source face into a target image or video in a way that is believable to a human observer, with no visible traces of modification [11,27]. The typical process includes: face detection and alignment, extraction of its semantic representation (embedding), reconstruction or conditional generation of a new texture, and applying it with a blending mask to the output frame [11].

Although theoretically the modification should be limited to the face area, in practice many models – especially those based on generative adversarial networks (GANs) – also affect the background, lighting, and global color statistics. There are many reasons for this behavior, ranging from the nature of the cost functions used to the specifics and complexity of the model architecture.

In the context of watermarking, this leads to two significant consequences. First, even local substitution can unintentionally distort the signal hidden throughout the image, reducing the effectiveness of invisible watermarking techniques. Second, suppose the visible watermark is located near a face or its pattern resembles an image artifact. In that case, the model may attempt to "correct" it, resulting in reduced legibility of the mark.

For further analysis, popular end-to-end networks such as SimSwap [12] and FaceShifter [13] were selected, as well as newer designs incorporating additional segmentation models, key point generation (Ghost [14], FastFake [15]), or closed solutions – InsightFace [16]. Each of these methods controls the scope of editing differently and achieves a different compromise between photorealism and precise control of the modification region.

In some cases, it was necessary to reimplement or adapt the models to meet the established experiment criteria, which may result in slight differences from the results presented by the authors of the original algorithms. Where possible, the same architectures and cost functions as in the original implementations were retained.

To establish a reference point, a proprietary reference method was also developed, based on classic inpainting in the face segmentation mask area. This method edits only the face region, preserving the background pixels, which allows for estimating the minimal impact of a perfectly localized face swap on the watermark. A comparison of these approaches will enable us to determine the extent to which modern, intensely trained models interfere with image content outside the target modification area and to present the theoretical reasons for this behavior.

2.2.1. SimSwap

SimSwap [12] is one of the first publicly available architectures that enable identity swapping for arbitrary pairs of faces without requiring retraining of the network. It combines the simplicity of a single encoder–decoder–GAN setup with the ability to work in many-to-many mode.

The key element of SimSwap is the ID Injection module. After encoding the target frame, the identity vector from the source—obtained from a pre-trained ArcFace model [28]—is injected into the deep layers of the generator using Adaptive Instance Normalization (AdaIN) blocks [29].

To preserve the facial expressions, pose, and lighting of the target image, the authors introduced Weak Feature Matching Loss, which compares the deep output representations of the discriminator between the target image and the reference image. This function promotes the consistency of visual attributes by treating the discriminator as a measure of realism rather than identity consistency.

Identity is enforced through a cost function based on the cosine distance between ArcFace embedding vectors. The realism of the generated images is improved by classic hinge-GAN loss and gradient penalty [30]. Additional Reconstruction Loss is activated when the source and target images depict the same person – in this case, the network learns to minimize changes in the image.

In practice, this combination of cost functions means that modifications are concentrated mainly in the face area, while the background and clothing elements remain largely unaffected. However, the lack of an explicit segmentation mask means that subtle color corrections may occur throughout the frame when there are substantial exposure changes or low contrast.

2.2.2. FaceShifter

FaceShifter [13] is a two-stage face swap network designed to preserve the identity of the source without requiring training for each pair. In the first phase (AEI-Net), the following components are combined: an identity embedding obtained from ArcFace [28] and multi-level attribute maps generated by a U-Net encoder.

Integration is achieved using the Adaptive Attentional Denormalization (AAD) mechanism, which dynamically determines whether a given feature fragment should originate from the embedding ID or the attribute maps. In addition to identity loss, adversarial loss, and reconstruction loss, the cost function also uses attribute loss, which enforces attribute consistency between the target image and the replaced image.

The lack of an explicit segmentation mask means that, in cases of significant differences in lighting or color, AAD can also modify the background, which, from a watermarking perspective, increases the risk of distorting the invisible watermark. At the same time, precise attention masks within AAD keep the primary energy of changes within the face.

2.2.3. Ghost

Authors of GHOST [14] presented a comprehensive, single-shot pipeline that covers all stages – from face detection to generation and super-resolution. However, only the GAN core is relevant in the context of this analysis. The basic architecture is a variation of AEI-Net known from FaceShifter, but with several significant modifications.

Similar to FaceShifter, the identity vector obtained from the ArcFace model is injected into the generator using Adaptive Attentional Denormalization (AAD) layers. A new feature is the use of an additional network targeting the eye region, along with a redesigned cost function – specifically, eye loss – which enables the stable reproduction of gaze direction in the generated image.

The second improvement is the adaptive blending mechanism, which dynamically expands or narrows the face mask based on the differences between the landmarks of the source and target images. This solution enhances the fit of the face shape and edges, thereby increasing the realism of the generated image.

However, in this work, the adaptive blending and super-resolution stages were omitted to focus solely on the analysis of pixel destruction introduced by the generator itself. Furthermore, some of the elements introduced in GHOST, such as adaptive blending, are not differentiable, which could disrupt the training process if a labeling model is to be used, treating face swaps as noise in the learning process.

2.2.4. FastFake

Fast Fake [15] is one of the newer examples of a lightweight GAN-based face swap, where the priority is fast and stable learning on small datasets, rather than achieving photographic perfection in each frame. The core of the model is a generator with Adaptive Attentional Denormalization (AAD) blocks, borrowed from FaceShifter [13]. Still, the entire architecture was designed in the spirit of FastGAN [31], featuring fewer channels, a skip-layer excitation mechanism [32], and a discriminator capable of reconstructing images, which helps limit the phenomenon of mode collapse.

The key difference from the previously discussed models lies in the way segmentation is utilized. The authors include a mask from the BiSeNet network [33] only at the loss calculation stage – pixels outside the face area are sent to the reconstructive MSE, and features obtained from the parser are additionally blurred and compared with analogous maps of the generated image. As a result, the generator learns to ignore the background, because any unjustified change in color increases the loss value. During inference, the mask is no longer used, keeping the computation flow clean and fast.

From the perspective of analyzing the impact of DeepFake on watermarking, this approach has significant implications. The scope of FastFake interference is even narrower than in SimSwap or FaceShifter – global color statistics change minimally, which potentially favors the protection of watermarks placed outside the face area. In theory, the GAN cost function should interfere to some extent with the component that enforces background preservation. However, it cannot be ruled out that the generator will still harm unusual elements of the image, such as watermarks.

Thanks to its low data requirements and fast learning process, FastFake is a representative example of the “economical” branch of face swap methods, which will be compared with other models in terms of their impact on the durability and legibility of watermarks later in this article.

2.2.5. InsightFace

The InsightFace Team [16] has not published a formal article describing the Inswapper module; however, this model is widely used in open-source tools, including Deep-Live-Cam [34], and functions as an informal “market standard” in the field of face swapping.

Similar to the previously discussed methods, Inswapper uses a pre-trained ArcFace model to determine the target’s identity. Although the implementation details are not fully known, a significant difference is the surrounding pipeline: InsightFace provides a complete SDK with its own face detection module and predefined cropping, which also includes arms and a portion of the background. If the detector does not detect a face or rates its quality below a certain threshold, the frame remains unchanged. In the context of watermarking, this means that elements outside the detected face mask can remain completely intact. This feature is also valuable for experiments – it allows you to assess whether the degradation of the watermark is significant enough to prevent the image from being used by popular face swap algorithms.

For this paper, the analysis is limited to the generator block and the mandatory face detector, omitting subsequent stages of the pipeline, such as skin smoothing and super-resolution. In this context, Inswapper serves as a realistic but minimal attack: any violation of the watermark is solely the result of identity substitution—provided that the face is detected and passed on for processing—which reflects a typical use case in popular consumer tools.

2.2.6. Baseline

The last face swap algorithm analyzed is a proprietary reference method explicitly developed for this comparison. Although it does not achieve SOTA results on its own, it stands out with its local face replacement range and an interesting approach to separating information from sources of the same type. Unlike the previously discussed GAN models, the training process uses elements characteristic of currently popular diffusion models [35].

The algorithm consists of three main components:

- U-Net – typical for diffusion models, responsible for removing the noise.

- Identity encoder – compresses input data into a one-dimensional hidden space; receives a photo of the same person, but in a different shot, pose, or lighting.
- Attribute encoder – also compresses data into a hidden space, but accepts the target image in its original form.

At the input, U-Net receives an image with a noisy face area (following the diffusion model approach) and a set of conditions: noise level, attribute vector, and identity vector.

The goal of the model is to recreate the input image based on additional information provided through conditioning. During inference, when a face of another person is fed to the identity encoder, the U-Net generates an image with the identity swapped, while preserving the pose, facial expressions, and lighting resulting from the attribute vector.

The key challenge is to motivate the model to utilize information from the identity encoder, rather than solely from the attribute encoder, which, in the absence of constraints, could contain all the data necessary for reconstruction. To prevent this, three modifications to the attribute vector were applied:

- Masking – randomly zeroing fragments of a vector, which forces the model to draw information from the identity encoder, as they may be insufficient on their own.
- Dropout – increases the dispersion of information in the vector, preventing data concentration in rarely masked fragments.
- Normal distribution constraint (KL divergence loss) – inspired by the VAE approach [36]; forces the elements of the attribute vector to carry a limited amount of information about the details of a specific image.

Thanks to these modifications, the model distributes information more evenly in the hidden space and obtains most of the identification features from the identity encoder.

During inference, the user can control both the noise step (typical for diffusion models) and the masking level of the attribute vector. Only one diffusion step was used in the study – subsequent steps would only improve the visual quality of the face swap without significantly affecting the hidden watermark.

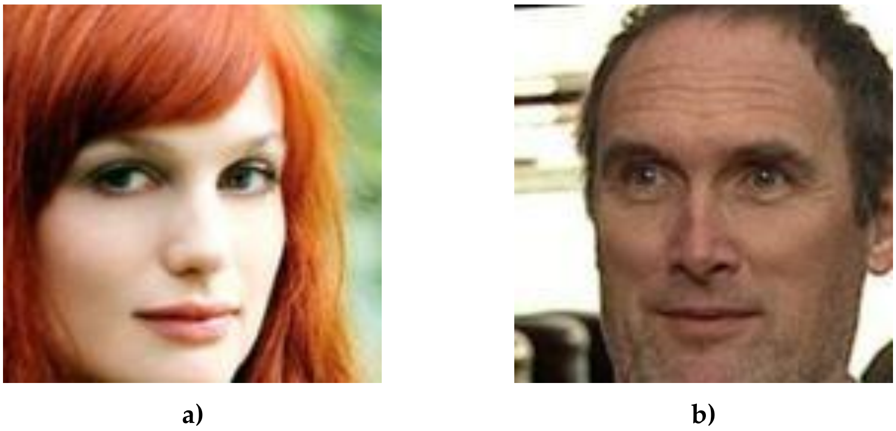
The selection of masking levels within the Baseline method was based on a series of preliminary tests with a trained network. The parameters were selected to ensure that the effect of masking on the extent of image modification was subjectively visible, while maintaining a comparable quality of the generated face. This differentiation enabled the analysis of how varying degrees of attribute isolation impact the degradation of the watermark.

The ability to control the noise level enables the generation of multiple test samples for various initial settings. In addition to analyzing the impact on the watermark, this approach can be used to augment training data for face-swap-resistant tagging systems.

2.2.7. Examples of Implemented Deepfakes

Images from the VGGFace2 [26] dataset were used for unit testing, shown in Figure 5:

a) image of the target face – the one that will be replaced,
b) source identity for face swap algorithms.

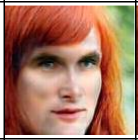
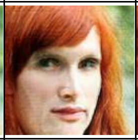
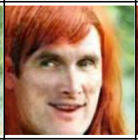
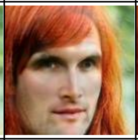
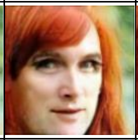
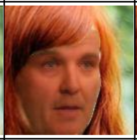
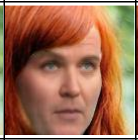
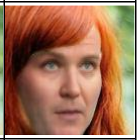
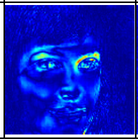
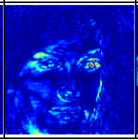
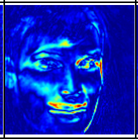
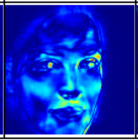
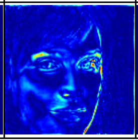
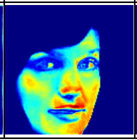
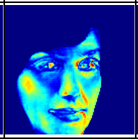
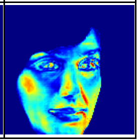


a) b)

Figure 5. Reference images used in unit and comparative tests: a) target image, b) source image.

Table 1 presents the sample face replacement results obtained for all models discussed (SimSwap, FaceShifter, GHOST, FastFake, InsightFace, and Baseline), along with heat maps illustrating the changes made. For the baseline model, three variants are presented, corresponding to different levels of attribute vector masking (high, medium, low).

Table 1. Sample face swap results and corresponding heat maps for the tested models.

	SimSwap	FaceShifter	Ghost	FastFake	InsightFace	Baseline High Mask	Baseline Medium Mask	Baseline Low Mask
Face Swapped								
Heatmap								

The table shows that some deepfake algorithms also introduce changes in the background of the image. This is particularly evident in the SimSwap and FaceShifter methods, where the right side of the heat maps indicates significant modifications outside the face area.

2.3. Experiments

Several research scenarios were conducted as part of the analysis. Due to the intertwining nature of the individual experiments, they were divided into two main blocks: visible tagging and hidden tagging. Each block broadly covers analogous types of analysis, specifically examining the impact of tagging on face swapping on a global scale, broken down into face and background areas.

First, the results for visible tagging will be presented, followed by those for hidden tagging. In both cases, individual examples of the impact of tagging with different parameter configurations are presented, followed by a statistical overview based on a test set.

The primary metrics used in the analyses are:

- ArcFace distance – the cosine distance between feature vectors extracted by the ArcFace model, allowing to assess whether the persons depicted in the compared images are recognized as the same.
- Pearson correlation – Pearson correlation coefficient between the image with the watermark and the image after applying face swap to the material containing the watermark.
- PSNR (Peak Signal-to-Noise Ratio) – used additionally in local analyses for a selected area (background).

The study analyzed two fundamental aspects:

The impact of watermarks on face swaps:

- Heatmaps of differences between the image after face swap performed on material with a watermark and the image after face swap performed on material without a watermark were compared.
- The ArcFace distance between these two variants was calculated to assess the impact of marking on identity recognition.
- Watermark retention:
- The Pearson correlation coefficient was calculated between the original image with the watermark and the image after face swapping was performed on the same image.
- Correlation maps and heat maps showing the distribution of changes in the image were generated.
- In local analyses, the background area was examined separately by calculating the PSNR for this region to estimate the impact of face swap outside the face area.

All experiments were performed on the VGGFace2 dataset. The VGGFace2 dataset includes photographs with varying lighting conditions, quality, compression, and cropping, which allowed us to naturally incorporate the diversity typical of images obtained from the web into our experiments. For each combination of parameters, calculations were performed on 1000 examples, allowing for statistically significant comparisons. We compare the results with a local baseline, which allows us to quantitatively demonstrate the non-locality of editing outside the face area.

3. Results

3.1. Visible Watermark

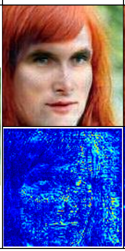
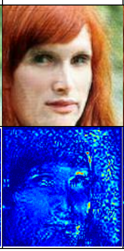
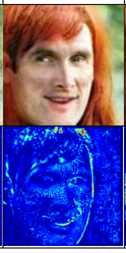
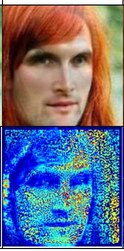
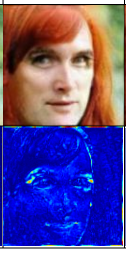
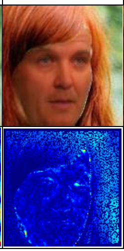
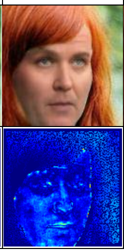
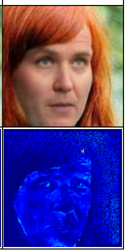
This section presents the results for a visible watermark, unrelated to any neural network training process, and its interaction with deepfake algorithms.

3.1.1. The Impact of Watermarks on the Face Swap Algorithm

The first study analyzed the impact of introducing a visible watermark on the performance of face swap algorithms, both in terms of generation quality and differences compared to the unmarked variant. Since this is a visible mark, a natural effect is a decrease in metrics such as PSNR as the opacity parameter increases.

Table 2 presents examples illustrating the differences between images after face swapping is performed on material with and without a watermark, along with corresponding heatmaps that show areas of change depending on the opacity value. Image grids illustrate spatial patterns, while statistical summaries quantify non-locality and non-monotonicity.

Table 2. Heatmaps showing differences between images after face swap with a watermark and images after face swap without a watermark – visible watermark case.

	Image with Watermark	SimSwap	Face Shifter	Ghost	Fast Fake	Insight Face	Baseline High Mask	Baseline Medium Mask	Baseline Low Mask
5%									

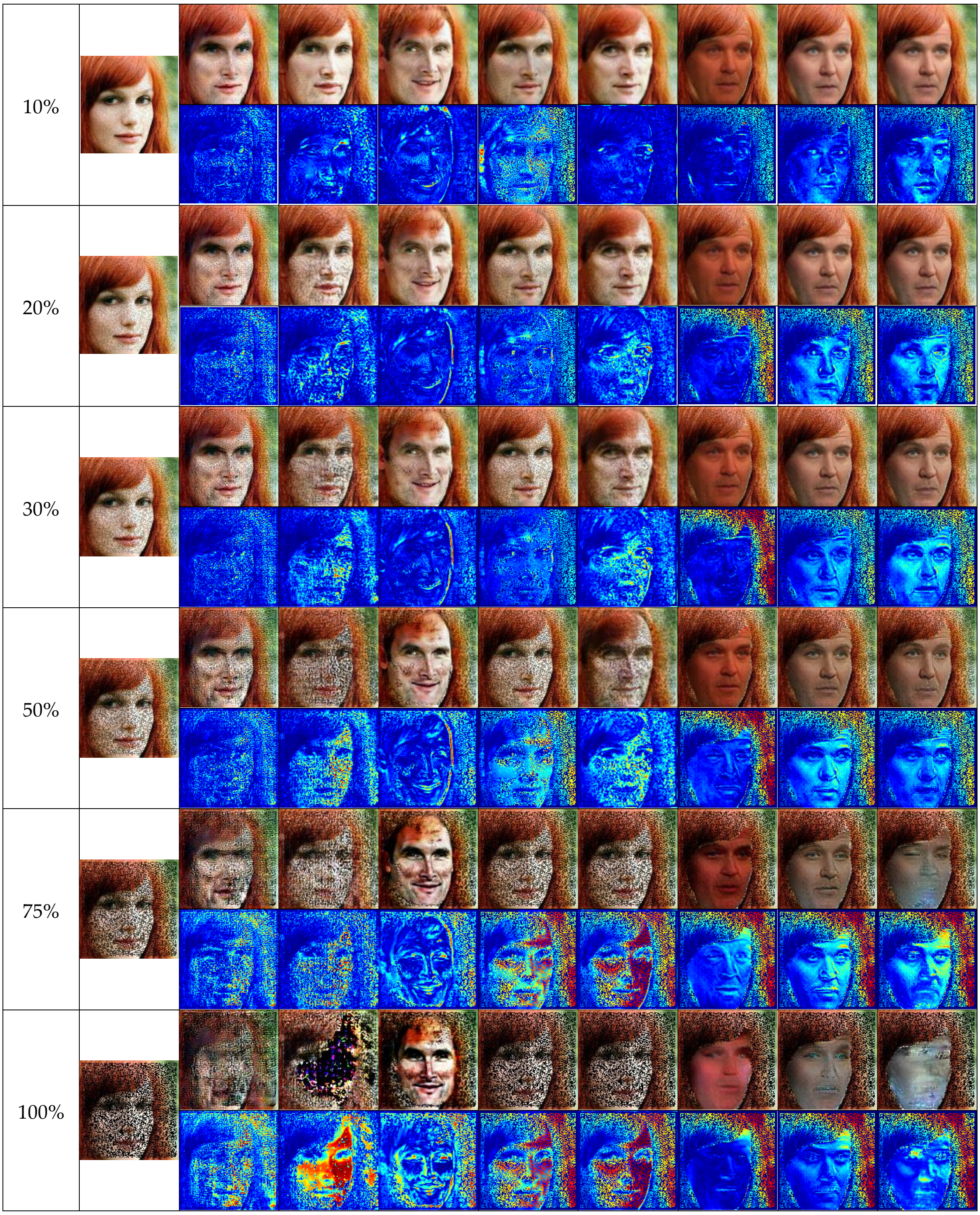


Figure 6 presents the results in the form of a numerical metric – the cosine distance between feature vectors obtained from the ArcFace model. A higher value of this metric means greater identity similarity between images.

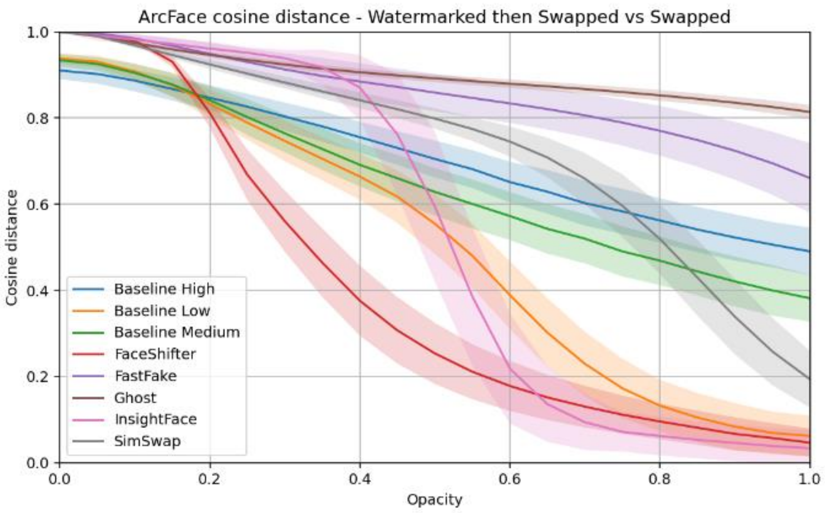


Figure 6. ArcFace cosine distance between the face-swapped image with a watermark and the face-swapped image without a watermark.

The analysis indicates that, regardless of the face swap method used, even when the target identity remains unchanged, degradation occurs. In the case of some algorithms, such as SimSwap, FaceShifter, GHOST, and the reference method (Baseline), mode collapse and the generation of results that do not correspond to the actual data were observed. InsightFace, on the other hand, stopped performing the swap with an opacity value above 40% because the detector rejected images due to low face quality.

The GHOST algorithm achieved the highest identity similarity metric, but qualitative analysis shows that it “forces” identity at the expense of background matching. FastFake behaved most consistently – although it missed a substitution (to a lesser extent than InsightFace), it did not generate unrealistic samples. The low ArcFace metric value for the Baseline method is due to the lack of training based on this metric. Additionally, the addition of a watermark that was not present in the training data resulted in visible errors in the face area.

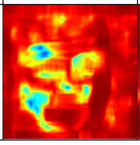
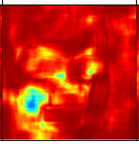
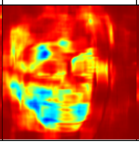
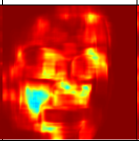
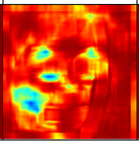
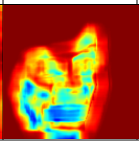
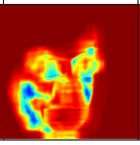
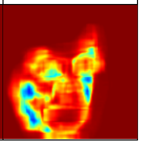
3.1.2. Watermark Resistance

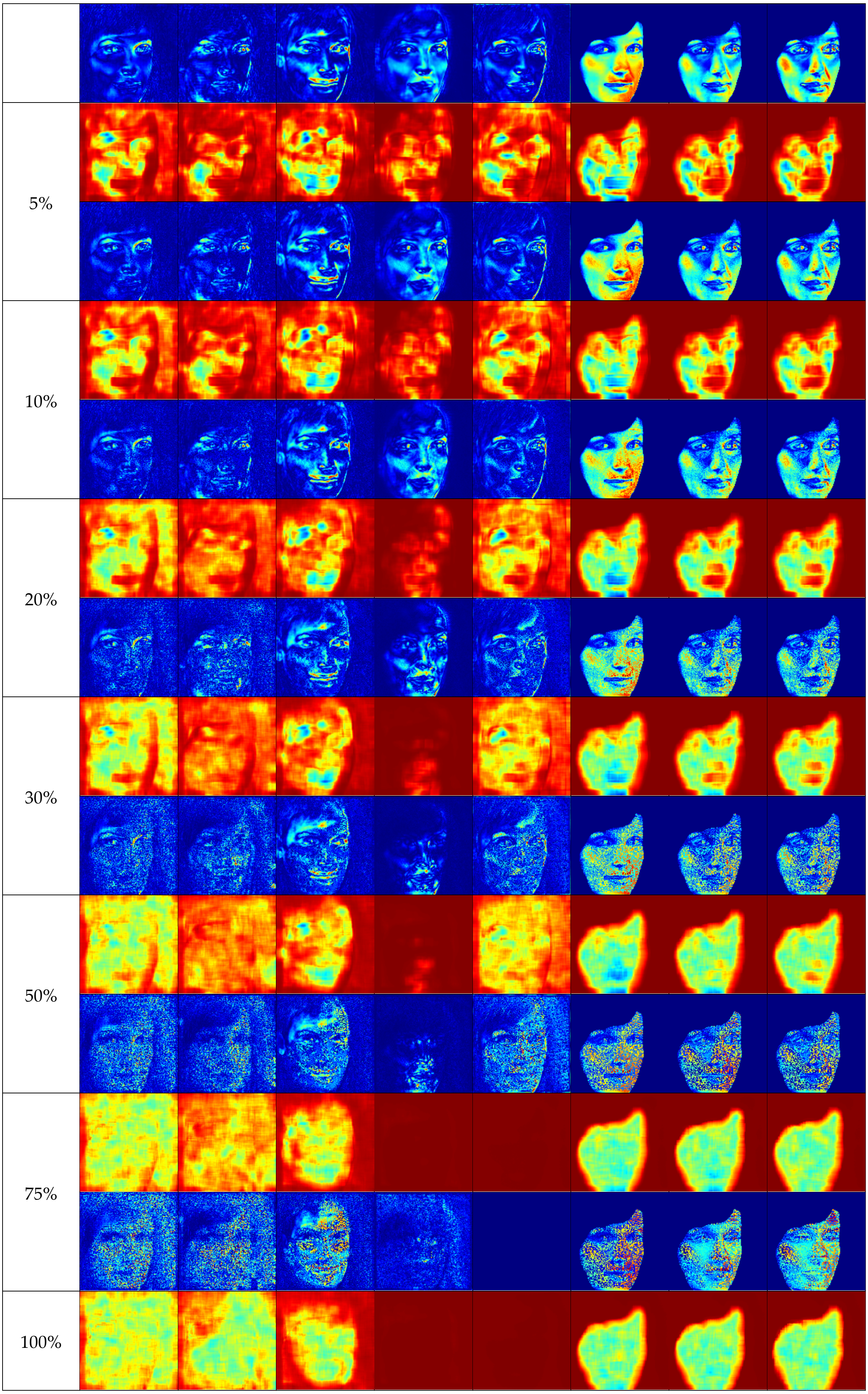
Another analysis concerned the behavior of the watermark after processing by face swap. The Pearson correlation between the original image with the watermark and the image after face swap was measured (Figure 7a). A high correlation value indicates that a significant portion of the image remained consistent, and the watermark was preserved.

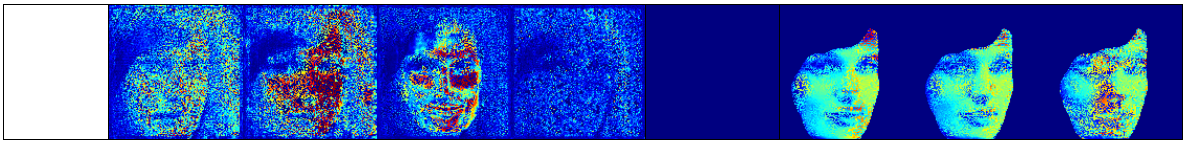
Due to the possibility that a visible watermark could affect the detector or generator and prevent substitution, the analysis was supplemented with individual examples with local correlation and heat map comparison (Table 3).

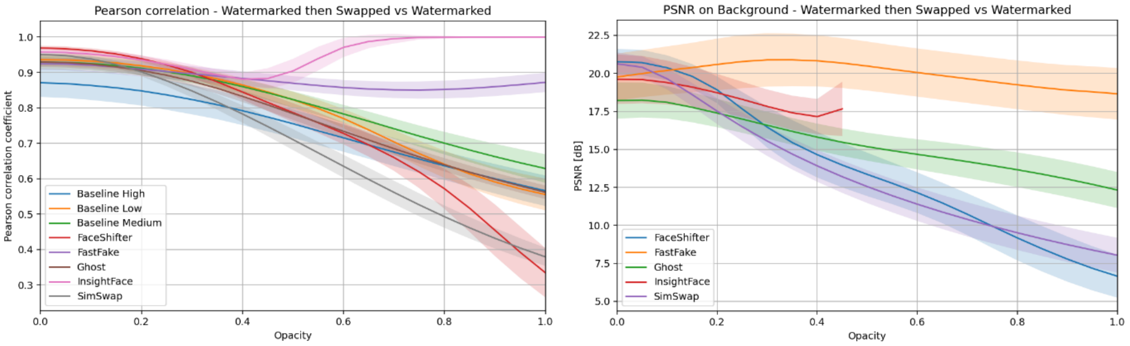
Additionally, Figure 7b shows PSNR values calculated only for the background area to assess the impact of modifications outside the face region. The Baseline method was omitted from this analysis because the background remained unchanged in this case.

Table 3. Local Pearson correlation and heatmaps of differences between the image with the watermark and the image after face swap – visible watermark case.

	SimSwap	Face Shifter	Ghost	Fast Fake	Insight Face	Baseline High Mask	Baseline Medium Mask	Baseline Low Mask
0%								


5%
10%
20%
30%
50%
75%
100%





a) **b)**

Figure 7. Graphs: a) Pearson correlation, b) PSNR background between the image with the watermark and the image after face swap – visible watermark case.

The results show that the correlation decreases with increasing opacity, except for InsightFace and FastFake. In InsightFace, the lack of face detection results in the image remaining unchanged, whereas in FastFake, the changes are minimal. Unit tests show that, compared to the reference method – Baseline, where only the face area is modified, SimSwap and FaceShifter interfere globally, degrading the watermark regardless of the opacity value. GHOST modifies the image to a lesser extent, and among the GAN-based methods, FastFake preserves the background best, as confirmed by the PSNR graph.

It is worth noting that there is no single "target" correlation value that would be considered ideal in this context. The Baseline method can be used as a reference point in the analysis – it only modifies the face area, leaving the background unchanged. Results significantly lower than those obtained for Baseline suggest that the deepfake algorithm introduces additional, undesirable modifications to the entire image.

A likely reason is the use of a segmentation-based cost function in FastFake, which rewards the preservation of image elements that do not belong to the face. The PSNR plot for InsightFace has been truncated because the method does not work at high opacity watermark values.

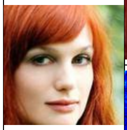
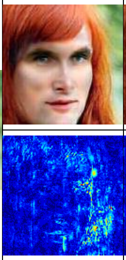
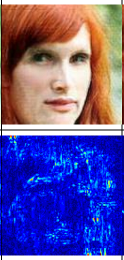
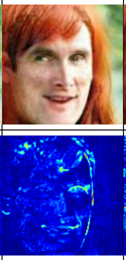
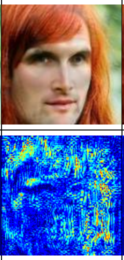
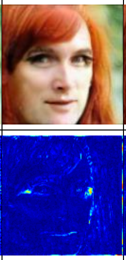
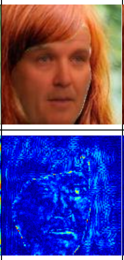
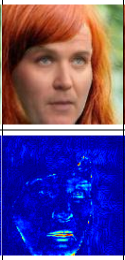
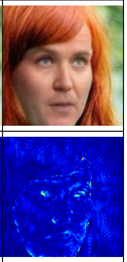
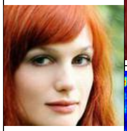
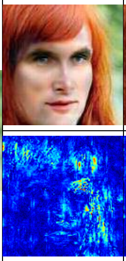
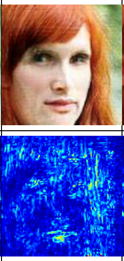
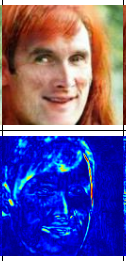
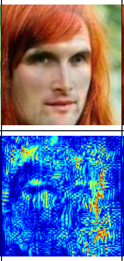
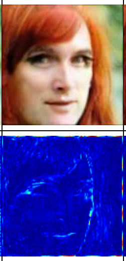
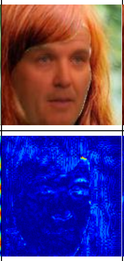
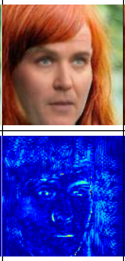
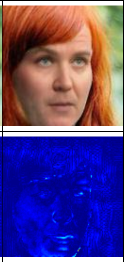
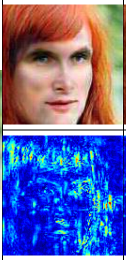
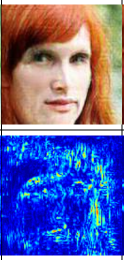
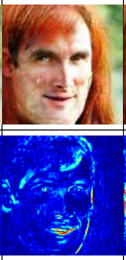
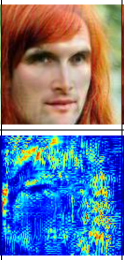
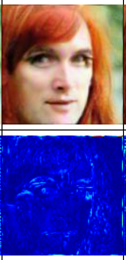
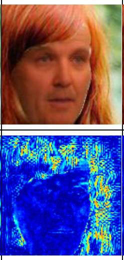
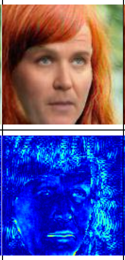
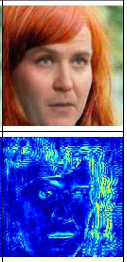
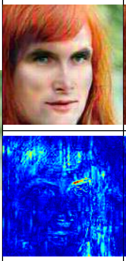
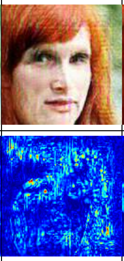
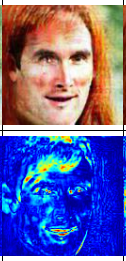
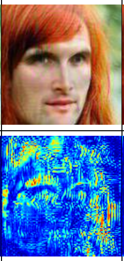
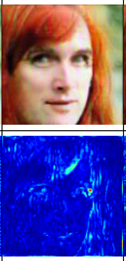
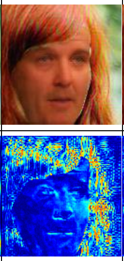
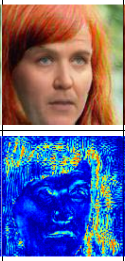
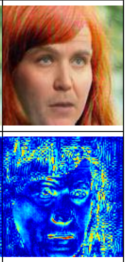
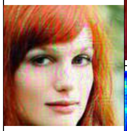
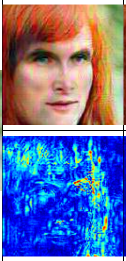
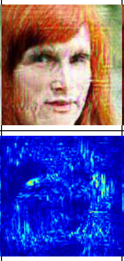
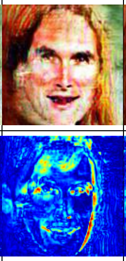
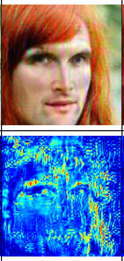
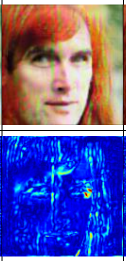
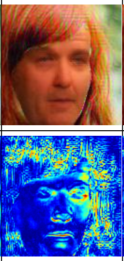
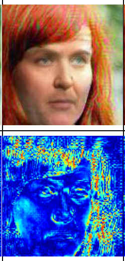
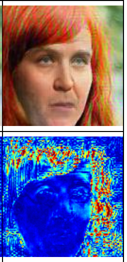
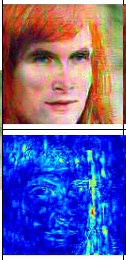
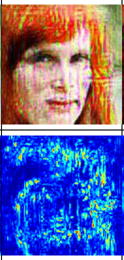
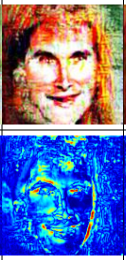
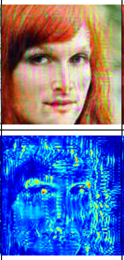
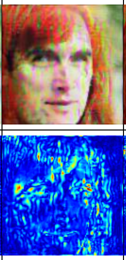
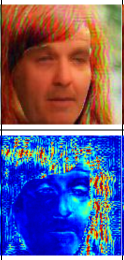
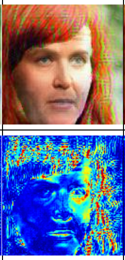
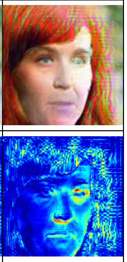
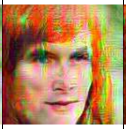
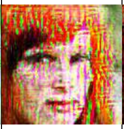
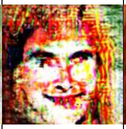
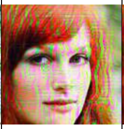
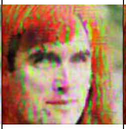
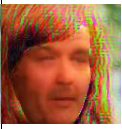
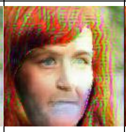
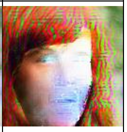
3.2. Hidden Watermark

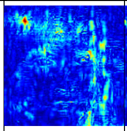
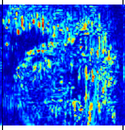
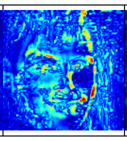
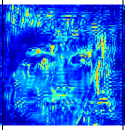
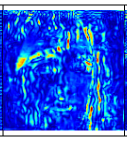
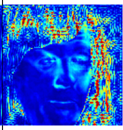
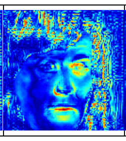
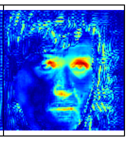
This section presents the results for a watermark trained using a neural network, designed to remain invisible to the viewer.

3.2.1. The Impact of Watermarks on the Face Swap Algorithm

As in the case of visible watermarks, the first stage of the analysis involved comparing the impact of introducing a hidden watermark on the performance of face swap algorithms. Table 4 presents individual examples illustrating the differences between images after face swapping performed on material with a watermark and images after face swapping without marking, together with corresponding heatmaps of areas of change depending on the opacity parameter value.

Table 4. Heatmaps of differences between images after face swap with a watermark and images after face swap without a watermark – hidden watermark case.

	Image with Watermark	SimSwap	Face Shifter	Ghost	Fast Fake	Insight Face	Baseline High Mask	Baseline Medium Mask	Baseline Low Mask
5%									
10%									
20%									
30%									
50%									
75%									
100%									

For numerical analysis, the cosine distance between feature vectors obtained from the ArcFace model was used again, as shown in Figure 8.

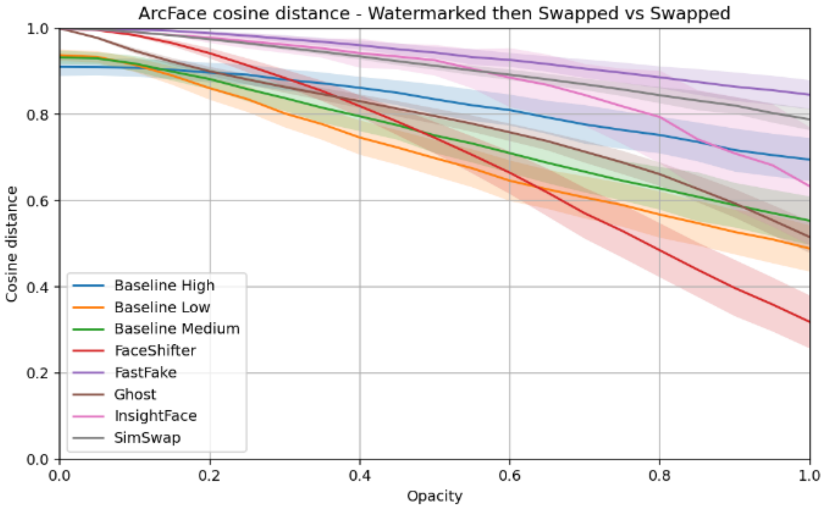


Figure 8. ArcFace cosine distance between the face-swapped image with a watermark and the face-swapped image without a watermark – hidden watermark case.

The results obtained largely reflect the trends observed in the case of visible marking, but with significant differences. In the case of hidden watermarks, SimSwap and FaceShifter did not exhibit typical mode collapse, although this phenomenon did occur in the GHOST model. FastFake, as before, gradually reduced its performance with increasing opacity, but InsightFace, unlike in the case of visible watermarks, continued to work throughout the opacity range, with its ArcFace metric curve showing an evident decline, which may suggest a collapse.

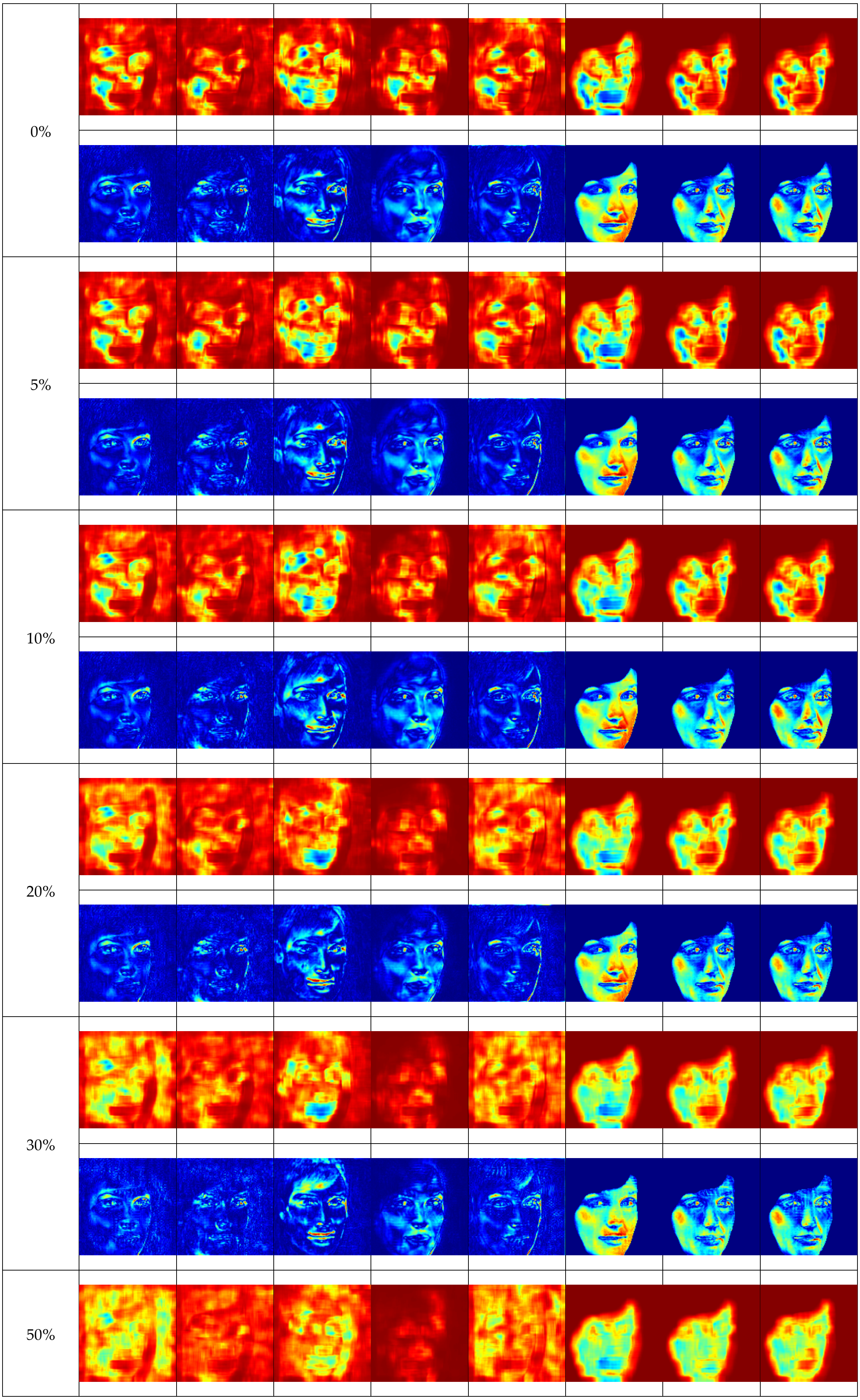
The most considerable differences in heatmaps were observed for the GHOST and FastFake models. In GHOST, image degradation was global and associated with a loss of background consistency. In contrast, in FastFake, it can be assumed that the watermark was treated as an important artifact that the model attempted to preserve at the expense of face replacement quality, which may result from the cost function adopted by the model.

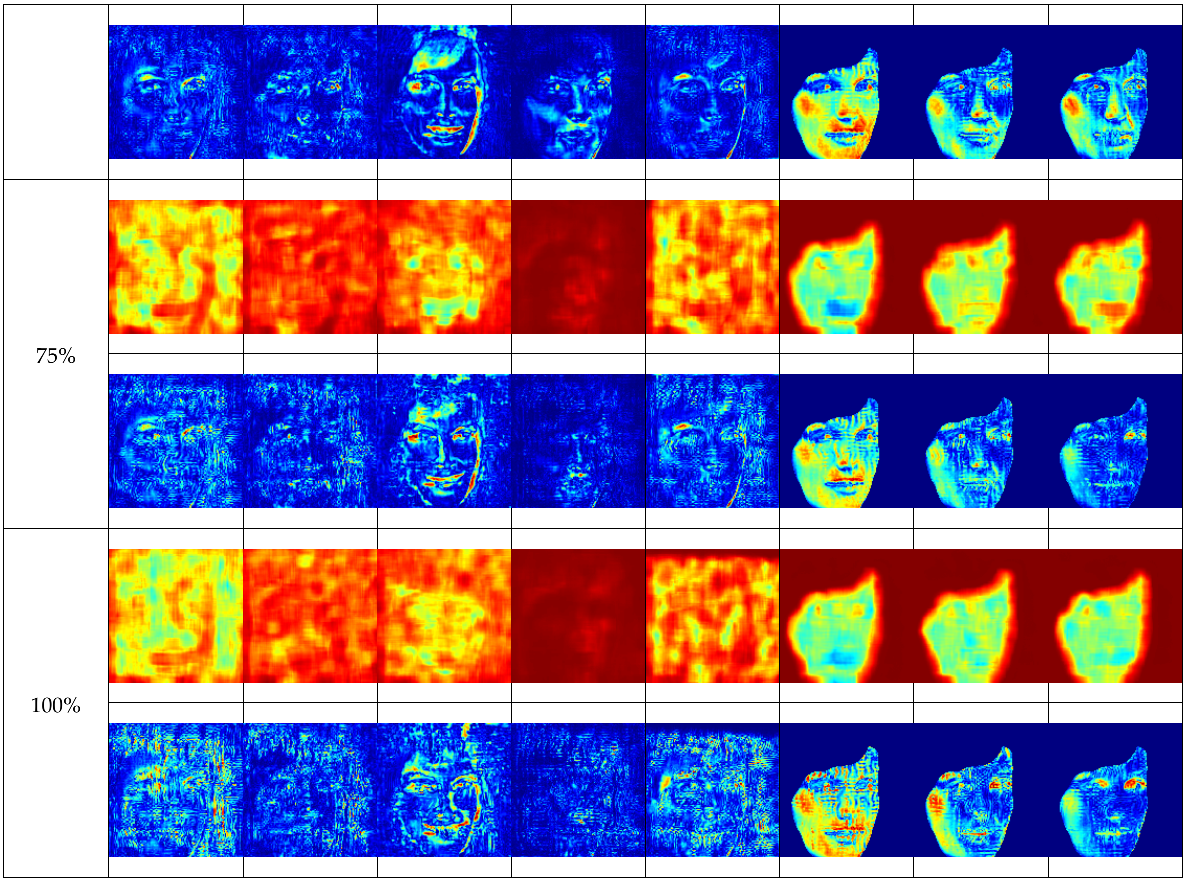
3.2.2. Watermark Resistance

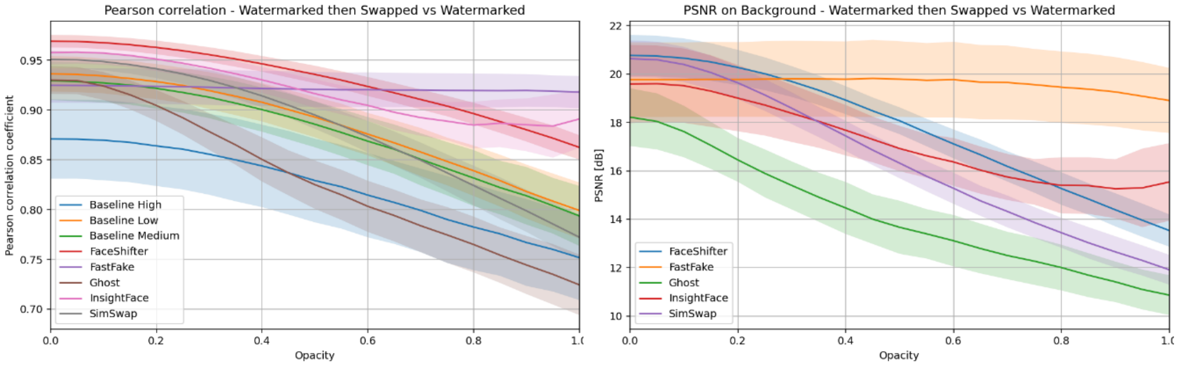
As in the case of the visible mark, the resistance of the watermark to face swap processing was analyzed based on Pearson's correlation and PSNR, which were calculated exclusively for the background area. Table 5 shows the local correlation values and heatmaps of the images, while Figure 9 shows the correlation and PSNR graphs for the background area.

Table 5. Local Pearson correlation and heatmaps of differences between the image with the watermark and the image after face swap – hidden watermark.

	SimSwap	Face Shifter	Ghost	Fast Fake	Insight Face	Baseline High Mask	Baseline Medium Mask	Baseline Low Mask


0%
5%
10%
20%
30%
50%





a) **b)**

Figure 9. Graphs: a) Pearson correlation, b) PSNR of the background between the image with the watermark and the image after face swap – hidden watermark case.

The results indicate that the correlation decreases as opacity increases. Similar to the visible mark, the InsightFace method demonstrates relative stability. In contrast, FastFake exhibits artifacts that are visible on correlation maps and heatmaps—global in nature, unlike the typical local reconstruction observed with the visible mark.

The GHOST algorithm caused the most significant degradation of the background, while FastFake, at high opacity values, retained relatively more background detail. SimSwap and FaceShifter, on the other hand, retained more information at low watermark opacity, suggesting that less interference with the image helps maintain the integrity of the hidden mark.

Discussion

Our results indicate that invisible watermarks, when trained for robustness to common degradations, generally retain more of their signal after face swap than visible marks. Importantly, the dominant effect is not confined to the face region. Several families of face swap models introduce non-local edits, which in turn degrade watermarks placed far from the face. This challenges the common operational assumption that keeping the mark away from the manipulated area is sufficient.

To our knowledge, no prior study has evaluated the same combination of factors: two watermark types with tunable strength, six heterogeneous face swap families, and region-wise analysis against a locality-preserving baseline. Therefore, direct numeric comparisons to the literature would be misleading. We therefore interpret PSNR, Pearson correlation, and ArcFace distance as relative indicators within our controlled sweeps, not as absolute performance claims. In this setting, two consistent patterns emerge. First, invisible watermarks exhibit higher background correlation after manipulation than visible ones, suggesting that robustness training partially aligns with the statistics of generator-induced changes. Second, background drift varies strongly by architecture: methods that couple identity transfer to global feature or adversarial losses tend to alter off-face regions more than models with segmentation-weighted objectives.

These observations refine how prior findings should be read. Classical robustness tests - blur, noise, resampling - approximate some but not all properties of deepfake generators. Nonlinear, model-dependent transforms can yield non-monotonic regimes where increasing watermark strength both harms identity transfer and worsens watermark retention. At the same time, in other regimes, lower strength invites unintentional smoothing that also erodes the signal. The practical implication is that watermark evaluation must be region-aware, strength-aware, and architecture-aware. Allocating energy near facial boundaries or with high local contrast increases the risk that generators reinterpret the mark as an artifact to be corrected. Moreover, strength tuning should avoid ranges that trigger detector rejection or significant shifts in global statistics.

Our study has limits that suggest clear next steps. We focus on still images at a fixed working resolution and on face swapping as the manipulation type. Extending to video with temporal consistency checks, as well as to other manipulation families (reenactment, lip-sync, complete synthesis), and to higher-resolution pipelines will test whether the same non-local effects persist. A standardized benchmark that couples region masks with background-only metrics would help align future studies.

In summary, the central insight is experimental and operational rather than purely algorithmic: under modern face-swapping techniques, the background placement of a watermark does not guarantee preservation. Robust content protection should incorporate region-wise evaluation against a locality baseline, report results as within-experiment effect sizes, and consider detector behavior as part of the end-to-end system.

Author Contributions: Investigation, T.W.; methodology, T.W.; supervision, Z.P.; validation, T.W.; resources T.W.; writing—original draft, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by the project of the Military University of Technology titled: “New Neural Network Architectures for Signal and Data Processing in Radiocommunications and Multimedia.” Project No. UGB 22-054/2025

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAD	Adaptive Attentional Denormalization
AdaIN	Adaptive Instance Normalization
AEI-Net	Adaptive Embedding Integration Network
ArcFace	Additive angular margin face recognition model
BCE	Binary Cross-Entropy
BCEWithLogitsLoss	Binary Cross-Entropy with logits
BiSeNet	Bilateral Segmentation Network
CV	Computer Vision
DIP	Digital Image Processing
FiLM	Feature-wise Linear Modulation
GAN	Generative Adversarial Network
ID	Identity embedding
KL	Kullback-Leibler divergence
LPIPS	Learned Perceptual Image Patch Similarity
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
QR	Quick Response code
ResNet	Residual Network
SDK	Software Development Kit
SSIM	Structural Similarity Index Measure
SOTA	State of the Art
U-Net	U-shaped convolutional neural network
VAE	Variational Autoencoder
VGGFace2	VGG Face dataset (version 2)

References

1. Qureshi, A.; Megías, D.; Kuribayashi, M. Detecting Deepfake Videos Using Digital Watermarking. In Proceedings of the 2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC); December 2021; pp. 1786–1793.
2. Duszejko, P.; Walczyna, T.; Piotrowski, Z. Detection of Manipulations in Digital Images: A Review of Passive and Active Methods Utilizing Deep Learning. *Appl. Sci.* **2025**, *15*, 881, doi:10.3390/app15020881.
3. Mahmud, B.U.; Sharmin, A. Deep Insights of Deepfake Technology : A Review 2023.
4. Westerlund, M. The Emergence of Deepfake Technology: A Review. *Technol. Innov. Manag. Rev.* **2019**, *9*, 40–53, doi:http://doi.org/10.22215/timreview/1282.
5. Amerini, I.; Barni, M.; Battiato, S.; Bestagini, P.; Boato, G.; Bruni, V.; Caldelli, R.; De Natale, F.; De Nicola, R.; Guarnera, L.; et al. Deepfake Media Forensics: Status and Future Challenges. *J. Imaging* **2025**, *11*, 73, doi:10.3390/jimaging11030073.
6. Lai, Z.; Yao, Z.; Lai, G.; Wang, C.; Feng, R. A Novel Face Swapping Detection Scheme Using the Pseudo Zernike Transform Based Robust Watermarking. *Electronics* **2024**, *13*, 4955, doi:10.3390/electronics13244955.
7. Zhu, J.; Kaplan, R.; Johnson, J.; Fei-Fei, L. HiDDeN: Hiding Data With Deep Networks 2018.
8. Tancik, M.; Mildenhall, B.; Ng, R. StegaStamp: Invisible Hyperlinks in Physical Photographs 2020.
9. Yao, Y.; Grosz, S.; Liu, S.; Jain, A. Hide and Seek: How Does Watermarking Impact Face Recognition? 2024.
10. Begum, M.; Uddin, M.S. Digital Image Watermarking Techniques: A Review. *Information* **2020**, *11*, 110, doi:10.3390/info11020110.
11. Walczyna, T.; Piotrowski, Z. Quick Overview of Face Swap Deep Fakes. *Appl. Sci.* **2023**, *13*, 6711, doi:10.3390/app13116711.
12. Chen, R.; Chen, X.; Ni, B.; Ge, Y. SimSwap: An Efficient Framework For High Fidelity Face Swapping. In Proceedings of the Proceedings of the 28th ACM International Conference on Multimedia; October 12 2020; pp. 2003–2011.
13. Li, L.; Bao, J.; Yang, H.; Chen, D.; Wen, F. FaceShifter: Towards High Fidelity And Occlusion Aware Face Swapping 2020.
14. Groshev, A.; Maltseva, A.; Chesakov, D.; Kuznetsov, A.; Dimitrov, D. GHOST—A New Face Swap Approach for Image and Video Domains. *IEEE Access* **2022**, *10*, 83452–83462, doi:10.1109/ACCESS.2022.3196668.

15. Walczyna, T.; Piotrowski, Z. Fast Fake: Easy-to-Train Face Swap Model. *Appl. Sci.* **2024**, *14*, 2149, doi:10.3390/app14052149.
16. Deepinsight/Insightface 2025.
17. Zhao, Y.; Wang, C.; Zhou, X.; Qin, Z. DARI-Mark: Deep Learning and Attention Network for Robust Image Watermarking. *Mathematics* **2023**, *11*, 209, doi:10.3390/math11010209.
18. Kaczyński, M.; Piotrowski, Z. High-Quality Video Watermarking Based on Deep Neural Networks and Adjustable Subsquares Properties Algorithm. *Sensors* **2022**, *22*, 5376, doi:10.3390/s22145376.
19. Wadhera, S.; Kamra, D.; Rajpal, A.; Jain, A.; Jain, V. A Comprehensive Review on Digital Image Watermarking 2022.
20. Horé, A.; Ziou, D. Image Quality Metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition; August 2010; pp. 2366–2369.
21. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, 2015; pp. 234–241.
22. Perez, E.; Strub, F.; Vries, H. de; Dumoulin, V.; Courville, A. FiLM: Visual Reasoning with a General Conditioning Layer 2017.
23. Huang, G.; Liu, Z.; Maaten, L. van der; Weinberger, K.Q. Densely Connected Convolutional Networks 2018.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition 2015.
25. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric 2018.
26. Cao, Q.; Shen, L.; Xie, W.; Parkhi, O.M.; Zisserman, A. VGGFace2: A Dataset for Recognising Faces across Pose and Age 2018.
27. Binderiya Usukhbayar Deepfake Videos: The Future of Entertainment 2020.
28. Deng, J.; Guo, J.; Yang, J.; Xue, N.; Kotsia, I.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5962–5979, doi:10.1109/TPAMI.2021.3087709.
29. Huang, X.; Belongie, S. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization 2017.
30. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the Proceedings of the 34th International Conference on Machine Learning; PMLR, July 17 2017; pp. 214–223.
31. Liu, B.; Zhu, Y.; Song, K.; Elgammal, A. Towards Faster and Stabilized GAN Training for High-Fidelity Few-Shot Image Synthesis 2021.
32. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks 2019.
33. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation 2018.
34. Estanislao, K. Hacksider/Deep-Live-Cam 2025.
35. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models 2020.
36. Kingma, D.P.; Welling, M. An Introduction to Variational Autoencoders. *Found. Trends® Mach. Learn.* **2019**, *12*, 307–392, doi:10.1561/2200000056.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.