DINOHash: Learning Adversarially Robust Perceptual Hashes from Self-Supervised Features

Shree Singhi [1 2] **Aayush Gupta** [3] **Lukas Struppek** [4 5]

Abstract

Open-source plays a vital role in machine learning research, yet projects focused on infrastructure and robustness often remain under-recognized. We present DINOHash, an open-source framework for robust perceptual image hashing designed for the provenance detection of AI-generated images. Unlike digital watermarking methods—which are easily broken by compression, cropping, or model leaks—DINOHash does not modify the image, making it inherently more secure, auditable, and resistant to transformations. DINOHash is built on top of the open-sourced DINOv2 features and incorporates adversarial training to ensure resilience against adversaries. DINOHash offers a reliable foundation for researchers and practitioners to build secure content verification systems. To promote ease of adoption, the model is released in PyTorch and ONNX formats as well as on npm.

Code: `https://github.com/proteus-photos/dinohash-perceptual-hash`.

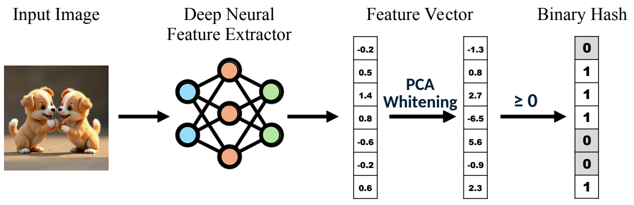

Figure 1. **Deep neural feature extractor architecture**. The pipeline consists of a feature extractor network and a Locality Sensitive Hashing (LSH) step. The PCA whitening decorrelates components of the feature vector, so that the hashes of two randomly selected images have the least probability of being identical.

1. Introduction

Open-source software has become a cornerstone of machine learning research and deployment, yet there has been no open source neural perceptual hash projects so far that are reproducible and robust. In this work, we present *DINOHash*, an open-source, adversarially robust perceptual hashing framework designed for the provenance verification of AI-generated images. DINOHash is engineered not only as a state-of-the-art hashing method but also as a usable, reproducible, and extensible software artifact to serve the ML community.

The increasing prevalence of AI-generated media has made content authenticity a pressing issue. Traditional solutions like digital watermarking (Coalition for Content Provenance and Authenticity, 2023; Wang et al., 2023a) often fall short under real-world transformations (e.g., compression, cropping, screenshots), and introduce vulnerabilities when model weights are publicly available. Furthermore, watermarking modifies the image itself and typically requires centralized infrastructure or secret keys, which limits transparency and hinders open adoption.

In contrast, perceptual hashing computes compact, transformation-invariant representations directly from image semantics using deep neural networks. It enables matching and moderation without altering content, making it a better fit for open, decentralized provenance systems. However, existing deep hashing methods (Apple Inc., 2021) have shown susceptibility to adversarial perturbations (Struppek et al., 2022; Bhatia & Meng, 2022), which undermines their trustworthiness.

DINOHash addresses these limitations by leveraging the open-sourced self-supervised DINOv2 (Oquab et al., 2023) vision transformer as a feature extractor, coupled with a custom adversarial training strategy to enhance robustness against both common and adversarial image transformations.

[1]Indian Institute of Technology, Roorkee [2]Zellic [3]MIT [4]German Research Center for AI [5]Technical University of Darmstadt, Germany. Correspondence to: Shree Singhi <$shree_s@mfs.iitr.ac.in$>.

Proceedings of the ICML 2025 Workshop on Championing Open-source Development in Machine Learning (CODEML '25). Copyright 2025 by the author(s).

2. Open-Source Development

Beyond the model itself, we emphasize sustainable development practices: the codebase is open-sourced, with support for both PyTorch and ONNX formats to ensure interoperability across research and production environments. All components—including the code, the base model (DINOv2), the trained model (on HuggingFace), and the training dataset (DiffusionDB)—are also publicly available, and our experiments are designed for reproducibility and extensibility. We include both clear instructions to re-train the model, as well as npm packages for Javascript developers to use our work. All our repositories are MIT licensed (note DINOv2 weights are CC-BY-SA-4.0; DiffusionDB is CC-BY-NCSA and those licenses continue to apply). We include documentation generated by DeepWiki that updates weekly.

3. Background and Related Work

In recent years, the field of deep hashing has seen significant advancements with numerous algorithms developed based on convolutional neural networks (Liong et al., 2015; Liu et al., 2016; Zhao et al., 2015; Zhang & Peng, 2019). These methods promise more robust hash computation compared to traditional hashing techniques. At their core, deep hashing methods utilize deep neural networks to extract salient image features, subsequently computing a compact binary hash value based on these learned representations. A notable example of deep perceptual hashing is Apple's NeuralHash (Apple Inc., 2021), which was developed for client-side scanning aimed at detecting Child Sexual Abuse Material (CSAM).

Formally, a perceptual hash function H can be defined as a mapping $H\colon \mathbb{R}^{H\times W\times C} \rightarrow \{0,1\}^k$, taking an image x (with dimensions height H, width W, and channels C) to a k-bit binary hash vector, $H(x) = \big(h_1(x), \dots, h_k(x)\big)$. Perceptual hashing algorithms typically comprise two main components. First, a deep feature extractor $D(x)\colon \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^k$ processes the input image x to produce a real-valued feature vector $z \in \mathbb{R}^k$. This vector numerically encodes the image's semantic and structural content. Second, a binarization or fuzzy hashing technique is applied to convert the feature vector z into the k-bit binary hash. A common approach for this binarization is locality-sensitive hashing (LSH) (Wei et al., 2024; Jafari et al., 2021), which aims to assign similar feature vectors to the same or nearby hash buckets with high probability. For binary hashing, this often involves defining random hyperplanes in the feature space, where each hash bit is determined by which side of the corresponding hyperplane the feature vector z lies on. This can be done using a Heaviside step function. Although NeuralHash (Apple Inc., 2021) is claimed to be robust to standard image transformations, its design did not explicitly account for adversarial robustness during training, rendering the system susceptible to such manipulations. Given the widespread deployment of hashing systems, accounting for adversarial efforts (Szegedy et al., 2014; Goodfellow et al., 2015) is crucial to ensure their trustworthiness and reliability. While deep perceptual hashing promises robustness to standard transformations, previous research (Struppek et al., 2022; Bhatia & Meng, 2022; Prokos et al., 2023; Madden et al.) has compellingly demonstrated that these algorithms are vulnerable to adversarial manipulations.

4. DINOHash: Increasing Robustness of Perceptual Hashing

To improve deep perceptual hashing robustness, we propose *DINOHash*, a novel system with adversarial training. Figure 1 shows its architecture, detailed below.

Architecture: At its core, our method leverages DINOv2 (Oquab et al., 2023), a powerful self-supervised vision transformer (Dosovitskiy et al., 2021) trained to produce rich feature embeddings that are inherently invariant to common image transformations such as cropping, color jittering, Gaussian blur, and polarization. Motivated by analyses demonstrating the strong performance of self-supervised models—particularly DINOv2's ability to capture robust and semantically meaningful features (Jiang et al., 2023; Darcet et al., 2024)—we chose it as the feature extractor backbone for DINOHash.

Our DINOHash architecture begins by utilizing a pre-trained DINOv2 ViT-B/14 model (with register tokens) to extract a real-valued feature embedding from the input image. To obtain the final binary hash, we apply two subsequent steps. First, Principal Component Analysis (PCA) is used to decorrelate the feature components and simultaneously reduce the dimensionality of the vector from 768 to 96. Second, we apply Locality-Sensitive Hashing (LSH) to binarize the reduced feature vector, yielding a 96-bit binary string. While our framework is flexible and can support different hash lengths, we configure the default output to 96 bits for direct comparability with systems like NeuralHash. Next, we describe our fine-tuning procedure to increase the adversarial robustness of the system.

Adversarial Training: Given a perceptual hashing model H, we aim to fine-tune its feature extractor such that an adversarially perturbed image x' receives the same hash as the original image x. The perturbation is constrained within an ℓ_∞-norm ball of radius ϵ, i.e., $\|x - x'\|_\infty \leq \epsilon$. Ideally, we want $H(x') = H(x)$. A naive loss function for this goal could be defined as:

$$\mathcal{L}_{\text{naive}} = \max_{\|x'-x\|_\infty \leq \epsilon} \|H(x') - H(x)\|_1. \qquad (1)$$

However, optimizing this loss risks the collapse of the hashing function, causing it to produce a constant output.

To overcome this, we leverage the original feature extraction network to provide a training signal, akin to knowledge distillation (Hinton et al., 2015) in an adversarial setting. We fine-tune a copy of the original model, referred to as the student model H_{stud}. The loss function targets the output of the original model H_{orig}:

$$\mathcal{L}_{\text{robust}} = \max_{\|x'-x\|_\infty \le \epsilon} \left\| H_{\text{stud}}(x') - H_{\text{orig}}(x) \right\|_1 . \quad (2)$$

Here, H_{orig} denotes the original hashing model before fine-tuning. Minimizing this objective ensures that the robust model H_{stud} maintains the hashing capabilities of the original model under adversarial perturbations, increasing robustness without collapsing. The inner maximization to find the adversarial example (that maximizes the hash difference) x' is solved using APGD (Croce & Hein, 2020), a strong adversarial attack algorithm.

Since the step function used for binarization is non-differentiable, we optimize the model in the logit space prior to binarization. Specifically, we use a cross-entropy loss $\mathcal{L}_{\text{CE}}$ on the logits. We also found that adding a clean loss term on the same batch—penalizing differences between the student and original models on clean images—helped stabilize training and speed up convergence. Let $\hat{H}_{\text{stud}}$ and $\hat{H}_{\text{orig}}$ denote the logit outputs of the robust and original models, respectively. The final loss function used to train the robust model $\hat{H}_{\text{stud}}$ on image x is:

$$\begin{aligned} \mathcal{L}_{\text{total}}(x) = \mathcal{L}_{\text{CE}} \left(\hat{H}_{\text{stud}}(x'), \hat{H}_{\text{orig}}(x) \right) \\ + \alpha \cdot \mathcal{L}_{\text{CE}} \left(\hat{H}_{\text{stud}}(x), \hat{H}_{\text{orig}}(x) \right) , \end{aligned} \quad (3)$$

where α is a hyperparameter balancing adversarial and clean consistency losses.

Statistical Test: Let $H \in \{0,1\}^k$ be the hash of an image x in the database, and let H' be the corresponding hash of a query image x'. The detection test compares the number of matching bits M between H and H' against a threshold τ, i.e., if

$$M(H, H') \ge \tau \quad \text{where} \quad \tau \in \{0, \ldots, k\}, \quad (4)$$

then the two images are considered a match. This formulation is necessary to define a decision boundary for visual similarity. When x and x' are unrelated images, we assume the hash bits $h_1, \ldots, h_k$ are i.i.d. Bernoulli random variables with parameter 0.5. Then $M(H, H')$ follows a binomial distribution with parameters $(k, 0.5)$.

The False Positive Rate (FPR) is the probability that two different images falsely pass the detection test, i.e., $M(H, H') \ge \tau$. The False Negative Rate (FNR) is the probability that two transformations of the same image fail the detection test. Varying τ creates a trade-off between FPR and FNR, which should be tuned based on the sensitivity of the application.

5. Experimental Evaluation

To directly improve the robustness of deep perceptual hashing, we propose *DINOHash*, a novel system whose resilience is significantly enhanced through an adversarial training procedure.

5.1. Experimental Setup

Models: We compare the performance of DINOHash against several perceptual hashing algorithms: NeuralHash, DCT-DWT, and Stable Signature. DCT-DWT is a classical image hashing algorithm known for its robustness to color distortions. Stable Signature (Wang et al., 2023a), a state-of-the-art watermarking method designed for image generation, is included as a baseline for comparison.

Datasets and Hyperparameters: We use 2 million randomly sampled images from the DiffusionDB dataset (Wang et al., 2023b) as our primary benchmark for evaluating the hashing systems. For evaluating the Stable Signature watermarking method, we generate 2 million images using prompts sampled from DiffusionDB, then apply the watermarking process. We fine-tune DINOHash on DiffusionDB for 20,000 steps with a batch size of 256, using the AdamW optimizer (Loshchilov & Hutter, 2017) with a weight decay of 10^{-4}. Adversarial examples are generated using APGD with 10 steps.

Metrics: We evaluate the systems using the True Positive Rate (TPR) and False Positive Rate (FPR) at various thresholds τ for the similarity score M. The threshold τ is varied from 0 to 96. The score M is defined as the number of matching bits (i.e. $96 -$ Hamming distance) between two hashes. Higher values indicate stronger similarity. We also report the average bit accuracy, defined as the percentage of matching bits between the hash or watermark of the transformed image and that of the original image.

Image Transformations: To evaluate robustness, we apply complex image transformation pipelines designed to simulate common image editing operations, including JPEG compression, crops, text overlays, and Gaussian blurs. Exact details along with a visualization of these complex transformation pipelines and their effects on sample images is provided in Figure 4.

5.2. Robustness to Image Transformations

Figure 2 presents the mean bit accuracy of the evaluated algorithms, while Figure 3 shows the Receiver Operating Characteristic (ROC) curves for each hashing or watermark-

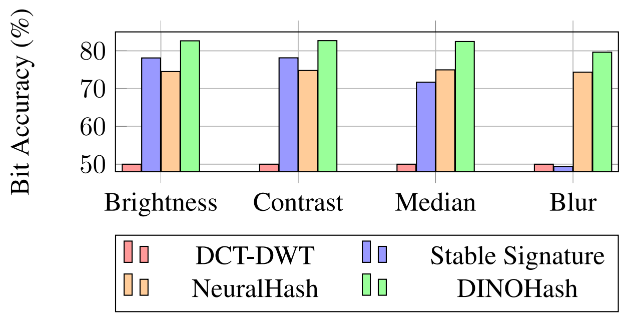

Figure 2. **Bit Accuracy Comparison.** The mean bit accuracy of different methods after a simulated screenshot followed by the respective transformations.

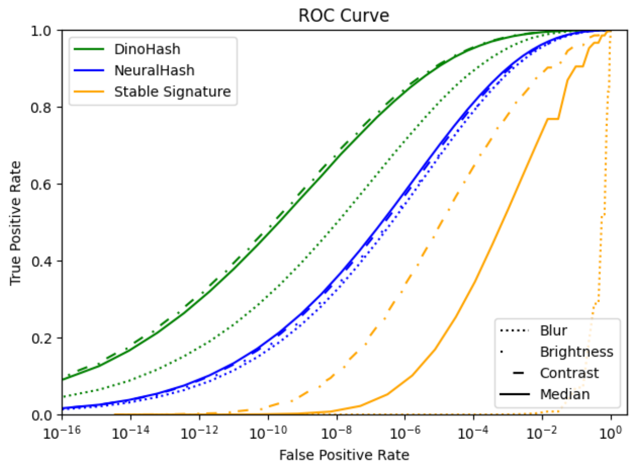

Figure 3. Detection results. TPR/FPR curves for different methods after a simulated screenshot operation, followed by the full transformation pipeline, including random erasing and text overlay.

ing method under various image transformations. We exclude the TPR-FPR plot for DCT-DWT due to its near-random performance across all settings.

We find that all methods perform similarly under brightness and contrast adjustments, likely due to the relatively non-destructive nature of these transformations. In contrast, DINOHash consistently outperforms both NeuralHash and Stable Signature across the full range of transformations and evaluation metrics, demonstrating substantially higher robustness. We omit results for the untransformed (clean) setting, as all hashing methods trivially achieve 100% matching accuracy in the absence of perturbations. Notably, this assumption does not necessarily hold for watermarking techniques, where discrepancies between the watermark generation and extraction models can lead to mismatches even without transformations.

5.3. Adversarial Robustness

We evaluate the adversarial robustness of DINOHash compared to various baseline models in Table 1 by performing a 100-step white-box APGD attack on each model and calculating its bit accuracy. Across all baseline systems, the attack is highly effective, driving the TPR down to near-zero. This indicates a complete collapse in matching capability under even moderate perturbations, reflecting a severe lack of adversarial robustness. In stark contrast, DINOHash maintains a high TPR, highlighting its resilience to gradient-based adversarial attacks. This suggests that DINOHash is substantially more reliable in adversarial settings, where image tampering may be subtle and intentionally obfuscated.

ϵ	DINOHash	DINOHash (untrained)	NeuralHash	Stable Signature
$\frac{4}{255}$	**72.2%**	0.3%	0.1%	0.0%
$\frac{8}{255}$	**64.8%**	0.0%	0.0%	0.0%

Table 1. Adversarial robustness of models under adversarial perturbations with the APGD attack performed using 100 steps.

We also report results for an untrained version of DINOHash, where the feature extractor has not been fine-tuned. This variant performs significantly worse than the trained version, validating that the robustness is not an artifact of the backbone alone but stems from meaningful learning during fine-tuning.

These findings underscore two key points: (1) that non-adversarially trained perceptual hash functions are highly vulnerable to targeted manipulation, and (2) that DINOHash's learning-based approach offers both semantic fidelity and robustness to adversarial perturbations, making it a more secure choice for real-world applications involving content provenance or tamper detection.

6. Conclusion

We introduce *DINOHash*, the first open-source deep perceptual hashing algorithm that explicitly incorporates adversarial training to achieve robust and reliable image matching under a wide range of perturbations. By leveraging self-supervised representations from DINOv2 and introducing a tailored adversarial fine-tuning procedure, DINOHash achieves significantly improved robustness over existing baselines. Importantly, DINOHash is released as an open-source framework, supporting adoption and fostering reproducibility and future research into robust perceptual hashing systems.

As concerns over provenance and content integrity grow, DINOHash offers a practical and extensible foundation for real-world applications. Future work could explore integrating fully homomorphic encryption to further enhance the privacy and security of hash comparisons in sensitive or decentralized environments.

Impact Statement

This paper's goal is to advance the field of image provenance and perceptual hashing. DINOHash enables reliable provenance checks for AI-generated imagery to curb misinformation and copyright abuse. However, the same technology can be used for large-scale automated content filtering or political censorship. False positives—especially on under-represented demographics—can lead to unjust account suspensions or information suppression. Because our pretrained weights are open, adversaries may craft hash-evasion attacks; our adversarial training improves—but does not eliminate—this risk, so we recommend combining DINOHash with complementary defences (e.g. watermarks) and retraining the model from scratch for proprietary production usecases.

References

Apple Inc. CSAM Detection - Technical Summary, 2021. URL `https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf`. Accessed: 05-July-2024.

Bhatia, J. S. and Meng, K. Exploiting and defending against the approximate linearity of apple's neuralhash, 2022. URL `https://arxiv.org/abs/2207.14258`.

Coalition for Content Provenance and Authenticity. C2PA technical specification. *Content Authenticity Initiative*, 2023.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Darcet, T., Oquab, M., Mairal, J., and Bojanowski, P. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations*, 2024.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Jafari, O., Maurya, P., Nagarkar, P., Islam, K. M., and Crushev, C. A survey on locality sensitive hashing algorithms and their applications, 2021. URL `https://arxiv.org/abs/2102.08942`.

Jiang, D., Liu, Y., Liu, S., Zhao, J., Zhang, H., Gao, Z., Zhang, X., Li, J., and Xiong, H. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023.

Liong, V. E., Lu, J., Wang, G., Moulin, P., and Zhou, J. Deep hashing for compact binary codes learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2475–2483, 2015. doi: 10.1109/CVPR.2015.7298862.

Liu, H., Wang, R., Shan, S., and Chen, X. Deep supervised hashing for fast image retrieval. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2064–2072, 2016. doi: 10.1109/CVPR.2016.227.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Madden, J., Bhavsar, M., Dorje, L., and Li, X. Robustness of practical perceptual hashing algorithms to hash-evasion and hash-inversion attacks. In *The Third Workshop on New Frontiers in Adversarial Machine Learning*.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al. Dinov2: Learning robust visual features without supervision. In *International Conference on Computer Vision and Pattern Recognition*, 2023.

Prokos, J., Fendley, N., Green, M., Schuster, R., Tromer, E., Jois, T., and Cao, Y. Squint hard enough: Attacking perceptual hashing with adversarial machine learning. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 211–228, 2023.

Struppek, L., Hintersdorf, D., Neider, D., and Kersting, K. Learning to break deep perceptual hashing: The use case neuralhash. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1795–1807, 2022.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing Properties of Neural Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Wang, J., Wang, M., Yang, Z., Wang, H., Li, Y., Zhang, L., and Qi, G.-S. Stable signature: Structural watermarking for diffusion models. In *arXiv preprint arXiv:2312.12391*, 2023a.

Wang, Z. J., Montoya, E., Munechika, D., Yang, H., Hoover, B., and Chau, D. H. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models, 2023b. URL https://arxiv.org/abs/2210.14896.

Wei, J., Peng, B., Lee, X., and Palpanas, T. Det-lsh: A locality-sensitive hashing scheme with dynamic encoding tree for approximate nearest neighbor search. *Proceedings of the VLDB Endowment*, 17(9):2241–2254, May 2024. ISSN 2150-8097. doi: 10.14778/3665844.3665854. URL http://dx.doi.org/10.14778/3665844.3665854.

Zhang, J. and Peng, Y. Ssdh: Semi-supervised deep hashing for large scale image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(1):212–225, January 2019. ISSN 1558-2205. doi: 10.1109/tcsvt.2017.2771332. URL http://dx.doi.org/10.1109/TCSVT.2017.2771332.

Zhao, F., Huang, Y., Wang, L., and Tan, T. Deep semantic ranking based hashing for multi-label image retrieval, 2015. URL https://arxiv.org/abs/1501.06272.

A. Visualization of Image Augmentations.

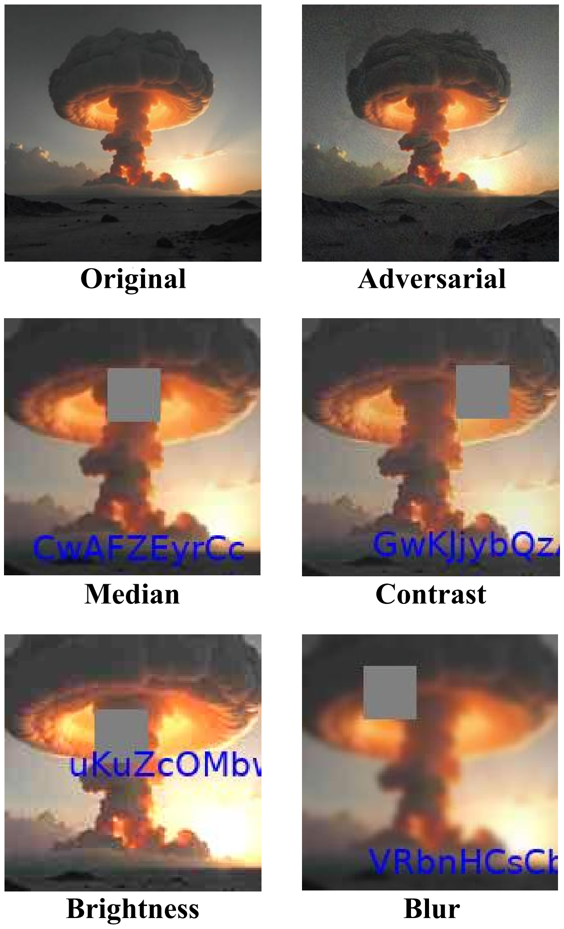

Figure 4. Transformations used to benchmark hashing and watermarking methods. Pre-processing is screenshot simulation, followed by random erasing, text overlay, and random erasing as post-processing. Each pipeline begins with a base transformation consisting of:

- A 20% *crop* from each side (top, bottom, left, and right), resulting in a $60\% \times 60\%$ center crop.
- JPEG *compression* with a quality factor of 30%.

We then apply one of the following operations:

- *Brightness* adjustment by a random factor of $\pm 30\%$.
- *Contrast* adjustment by a random factor of $\pm 30\%$.
- Gaussian *blur* with standard deviation $\sigma = 2$.

To further stress-test the models, we apply two additional augmentations *after* the above sequence:

- *Random erasing:* a square region with side length 20% of the image dimension is erased.
- *Random text overlay:* a string of 10 random alphanumeric characters is placed over the image.