

Article

BASN—Learning Steganography with a Binary Attention Mechanism

Pin Wu ¹, Xuting Chang ¹ and Yang Yang ¹ and Xiaoqiang Li ^{1,2,*}

¹ School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; wupin@shu.edu.cn (P.W.); changxuting@shu.edu.cn (X.C.); adamcavendish@shu.edu.cn (Y.Y.)

² Shanghai Institute for Advanced Communication & Data Science, Shanghai University, Shanghai 200444, China

* Correspondence: xqli@shu.edu.cn

Received: 15 January 2020; Accepted: 24 February 2020; Published: 27 February 2020



Abstract: Secret information sharing through image carriers has aroused much research attention in recent years with images' growing domination on the Internet and mobile applications. The technique of embedding secret information in images without being detected is called image steganography. With the booming trend of convolutional neural networks (CNN), neural-network-automated tasks have been embedded more deeply in our daily lives. However, a series of wrong labeling or bad captioning on the embedded images has left a trace of skepticism and finally leads to a self-confession like exposure. To improve the security of image steganography and minimize task result distortion, models must maintain the feature maps generated by task-specific networks being irrelative to any hidden information embedded in the carrier. This paper introduces a binary attention mechanism into image steganography to help alleviate the security issue, and, in the meantime, increase embedding payload capacity. The experimental results show that our method has the advantage of high payload capacity with little feature map distortion and still resist detection by state-of-the-art image steganalysis algorithms.

Keywords: convolutional neural network; steganography; attention mechanism

1. Introduction

Image steganography aims at delivering a modified cover image to secretly transfer hidden information inside with little awareness of the third-party supervision. On the other side, steganalysis algorithms are developed to find out whether an image is embedded with hidden information or not, and, therefore, resisting steganalysis detection is one of the major indicators of steganography security. In the meantime, with the booming trend of convolutional neural networks, a massive amount of neural-network-automated tasks are coming into industrial practices like image auto-labeling through object detection [1,2] and classification [3,4], face recognition [5], pedestrian re-identification [6], etc. Image steganography is now facing a more significant challenge from these automated tasks, whose embedding distortion might influence the task result in a great manner and irresistibly lead to suspicion. Figure 1 is an example that LSB (Least Significant Bit)-Matching [7] steganography completely alters the image classification result from goldfish to proboscis monkeys. Under such circumstances, a steganography model even with outstanding invisibility to steganalysis methods still cannot be called secure where the spurious label might re-arouse suspicion and, finally, all efforts are made in vain (Source code will be published at: <https://github.com/adamcavendish/BASN-Learning-Steganography-with-Binary-Attention-Mechanism>).

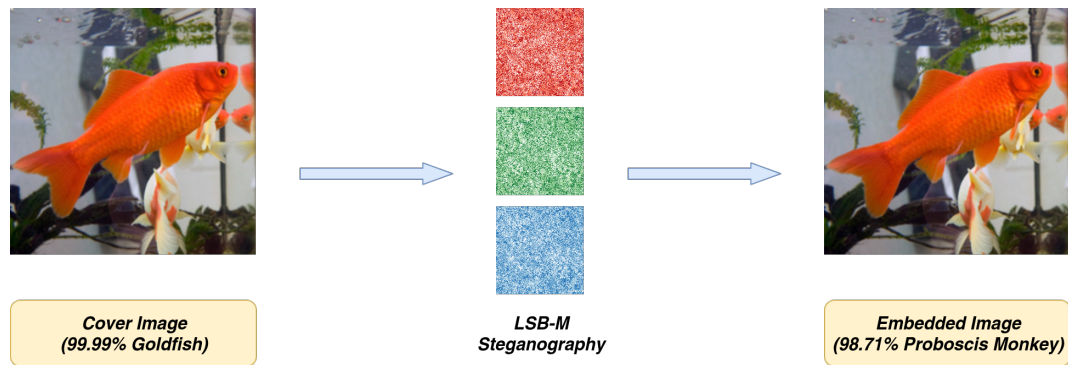


Figure 1. LSB-matching embedded image misclassification.

The cover image and embedded image both use an ImageNet pretrained ResNet-18 [3] network for classification. The percentage before the predicted class label represents a network’s confidence in prediction. The red, green, and blue noisy images in the center represent the altered pixel locations in corresponding channels during steganography. There are only three kinds of colors within these images where white stands for no modification, the lighter one stands for a +1 modification, and the darker one stands for a −1 modification.

Most previous steganography models focus on resisting steganalysis algorithms or raising embedding payload capacity. BPCS (Bit-Plane Complexity Segmentation) [8,9] and PVD (Pixel-Value Differencing) [10–12] use adaptive embedding based on local complexity to improve embedding visual quality. HuGO [13] and S-UNIWARD [14] resist steganalysis by minimizing a suitably defined distortion function. Wang [15] adopts a gray level co-occurrence matrix to calculate the image texture complexity. Meng’s method [16] improves security and robustness with the aid of Faster R-CNN’s object detection on texture complex areas. Xue [17] proposes optimal dispersion degree on halftone images to measure region texture complexity. Huang [18] adds a texture-based loss to help GAN hide information. Liao [19] establishes a framework with image texture complexity and distortion distribution strategies to embed hidden information across multiple cover images. Hu [20] adopts a deep convolutional generative adversarial network to achieve steganography without embedding. Wu [21] and Baluja [22] achieve a vast payload capacity by focusing on image-into-image steganography.

In this paper, we propose a Binary Attention Steganography Network (abbreviated as **BASN**) architecture to achieve a relatively high payload capacity (2–3 bpp, bits per pixel) with minimal distortion to other neural-network-automated tasks. It utilizes convolutional neural networks with two attention mechanisms, which minimize embedding distortion to the human visual system and neural network feature maps, respectively. Additionally, multiple attention fusion strategies are suggested to balance payload capacity with security, and a fine-tuning mechanism are put forward to improve the hidden information extraction accuracy.

2. Binary Attention Mechanism

Binary attention mechanism involves two attention models including an image texture complexity (ITC) attention model and a minimizing feature distortion (MFD) attention model. The attention mechanism in both models serves as a hint for steganography showing where to embed or extract and how much information the corresponding pixel might tolerate. The ITC model mainly focuses on deceiving the human visual system from noticing the differences out of altered pixels. The MFD model minimizes the high-level features extracted between clean and embedded images so that neural networks will not give out diverge results. With the help of the MFD model, we align the latent space of the cover image and the embedded image so that we can infer the original cover attention map using solely the embedded image. Afterwards, the hidden information is extracted at the locations and capacity indicated by the inferred attention map. The conjoint effort of both models can maintain a

high security and robustness against neural-network-automated tasks in addition to human perceptual invisibility.

The embedding and extraction overall architecture are shown in Figure 2 where both models are trained for the ability to generate their corresponding attention. The training process and the details of each model are elaborated in Sections 2.2 and 2.3. After attention is placed on the binary attention mechanism, we may adopt several fusion strategies to create the final amount of attention used for embedding and extraction. The fusion strategies are compared for their pros and cons in Section 3.

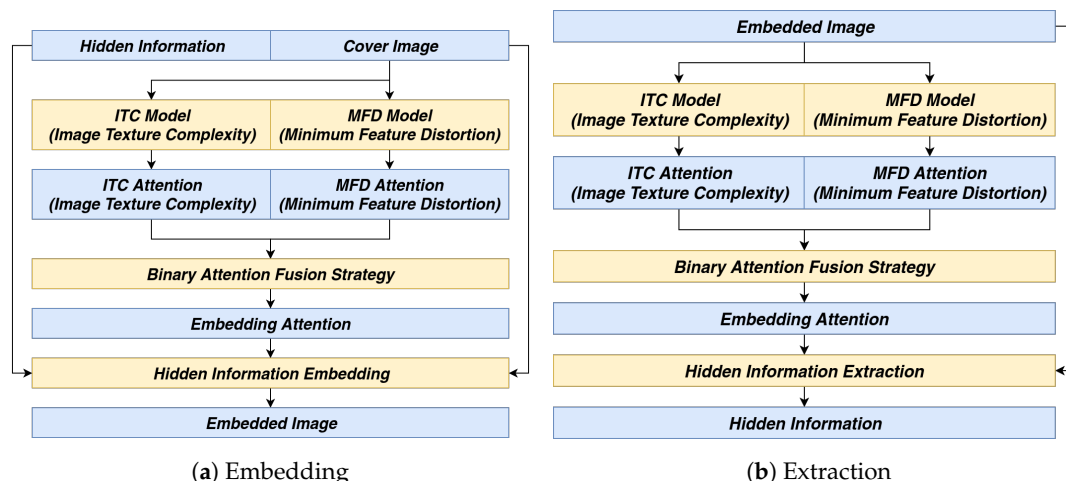


Figure 2. The embedding and extraction architecture.

2.1. Evaluation of Image Texture Complexity

To evaluate an image’s texture complexity, variance is adapted in most approaches. However, using variance as the evaluation mechanism enforces very strong pixel dependencies. In other words, every pixel is correlated to all other pixels in the image.

We propose a variance pooling evaluation mechanism to relax cross-pixel dependencies (see Equation (1)). Variance pooling applies on patches but not the whole image to restrict the influence of pixel value alterations within the corresponding patches. Especially in the case of training when optimizing local textures to reduce its complexity, pixels within the current area should be most frequently changed while far distant ones are intended to be reserved for keeping the overall image contrast, brightness, and visual patterns untouched:

$$\text{VarPool2d}(X_{i,j}) = \mathbb{E}_{k_i} \left[\mathbb{E}_{k_j} \left[X_{i+k_i, j+k_j}^2 \right] \right] - \mathbb{E}_{k_i} \left[\mathbb{E}_{k_j} \left[X_{i+k_i, j+k_j} \right]^2 \right] \tag{1}$$

$$k_i \in \left[-\frac{n}{2}, \frac{n}{2} \right], k_j \in \left[-\frac{n}{2}, \frac{n}{2} \right]$$

In Equation (1), X is a two-dimensional random variable which can either be an image or a feature map and i, j are the indices of each dimension. Operator $\mathbb{E}[\cdot]$ calculates the expectation of the random variable. VarPool2d applies a similar kernel mechanism as other two-dimensional pooling or convolution operations [23,24], and k_i, k_j indicates the kernel indices of each dimension.

To further show the impact of gradients updating between variance and variance pooling during backpropagation, we applied the gradients backpropagated directly to the image to visualize how gradients influence the image itself during training (see Equations (2) and (3) for training loss and Figure 3 for the impact comparison):

$$\mathcal{L}_{\text{Variance}} = \text{Var}(X) \tag{2}$$

$$\mathcal{L}_{\text{VarPool2d}} = \mathbb{E} [\text{VarPool2d}_{n=7}(X)] \tag{3}$$

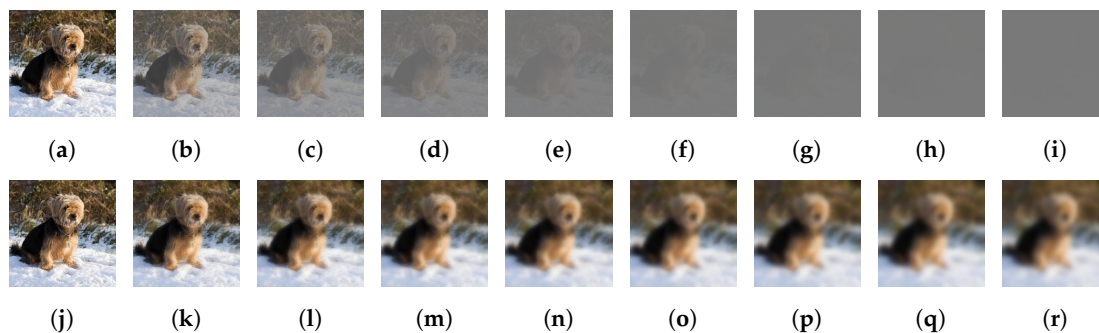


Figure 3. The gradient impact comparison between variance and variance pooling during training.

The first row shows the impact of variance while the second shows that of variance pooling. The visualization interval is 5000 steps of gradient backpropagation on the corresponding image.

2.2. ITC Attention Model

The ITC (Image Texture Complexity) attention model aims to embed information without being noticed by the human visual system, or, in other words, making just a noticeable difference (JND) to cover images to ensure the largest embedding payload capacity [25]. In texture-rich areas, it is possible to alter pixels to carry hidden information without being noticed. Finding the ITC attention means finding the positions of the image pixels and their corresponding capacity that tolerate mutations.

Here, we introduce two concepts:

1. A hyper-parameter θ representing the ideal embedding payload capacity that the input image might achieve.
2. An ideal texture-free image C_θ corresponding to the input image that is visually similar but with the lowest texture complexity possible regarding the restriction of at most θ changes.

With the help of these concepts, we can formulate the aim of ITC attention model as:

For each cover image C , the ITC model f_{itc} needs to find an attention $A_{itc} = f_{itc}(C)$ to minimize the texture complexity evaluation function V_{itc} :

$$\text{minimize } V_{itc}(A_{itc} \cdot C_\theta + (1 - A_{itc}) \cdot C) \tag{4}$$

$$\text{subject to } \frac{1}{N} \sum_i A_{itc} \leq \theta \tag{5}$$

The θ in Equation (5) is used as an upper bound to limit down the attention area size. If trained without it, model f_{itc} is free to output an all-ones matrix A_{itc} to acquire an optimal texture-free image. It is well known that an image with the least amount of texture is a solid color image, which does not help find the correct texture-rich areas.

In an actual training process, the detailed model architecture is shown in Figure 4 and two parts of the equation are slightly modified to ensure better training results. First, the ideal texture-free image C_θ in Equation (4) does not indeed exist but is available through approximation nonetheless. In this paper, median pooling with a kernel size of 7 is used to simulate the ideal texture-free image. It helps eliminate detailed textures within patches without touching object boundaries (see Figure 5 for comparison among different smoothing techniques and Figure 6 for comparison among various kernel sizes). Second, we adopt soft bound limits in place of a hard upper bound in the form of Equation (6) (visualized in Figure 7). Soft limits help generate smoothed gradients and provide optimizing directions.

<i>Input Image</i>
<i>Conv2D (3, 3x3, 32)</i>
<i>ELU (Exponential Linear Units)</i>
<i>Conv2D (32, 3x3, 64)</i>
<i>Batch-Normalization</i>
<i>ELU (Exponential Linear Units)</i>
<i>Conv2D (64, 3x3, 32)</i>
<i>Batch-Normalization</i>
<i>ELU (Exponential Linear Units)</i>
<i>Conv2D (32, 3x3, 3)</i>
<i>Sigmoid</i>
<i>ITC Attention (Image Texture Complexity)</i>

Figure 4. ITC attention model architecture.

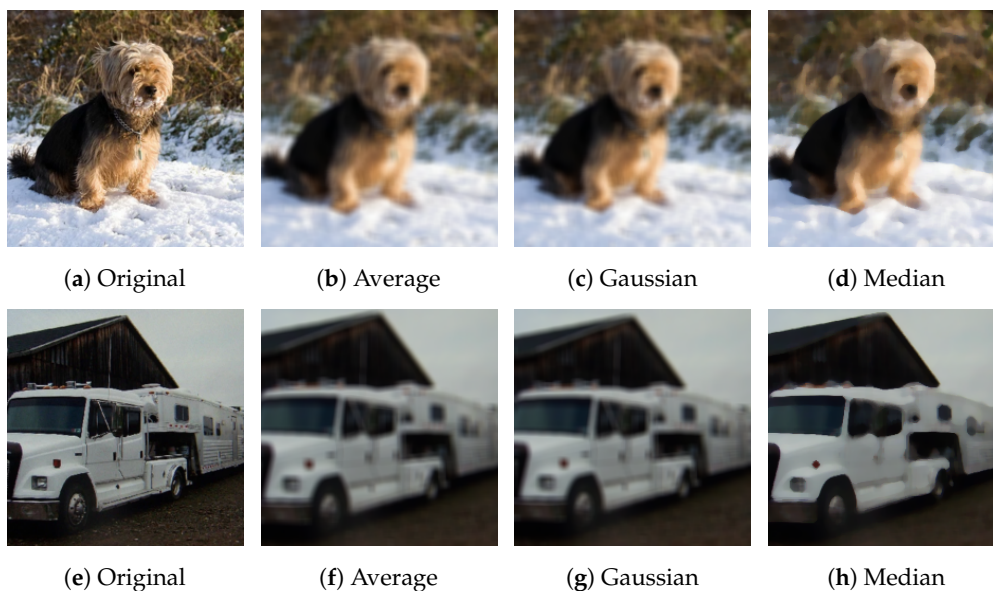


Figure 5. Image smoothing effect comparison.

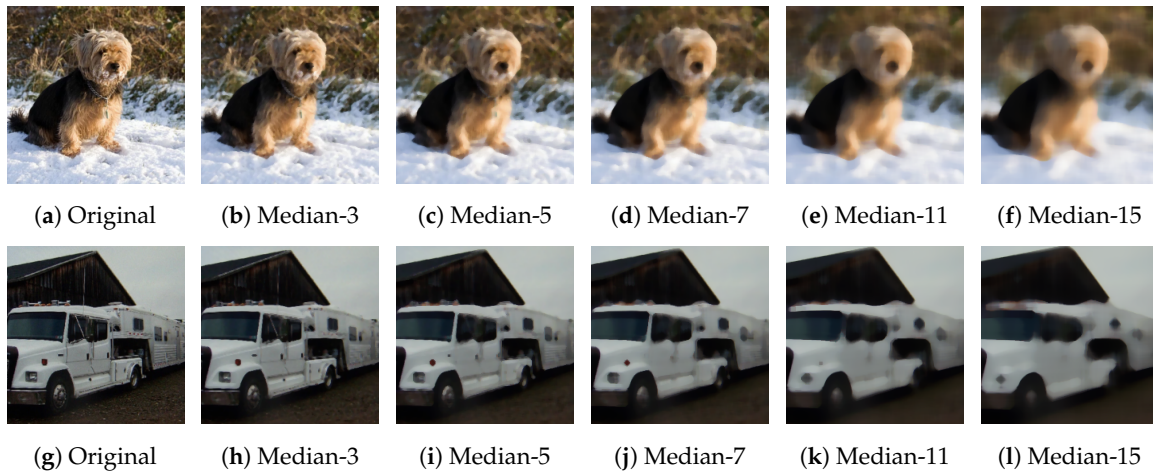


Figure 6. Various kernel size effect comparison.

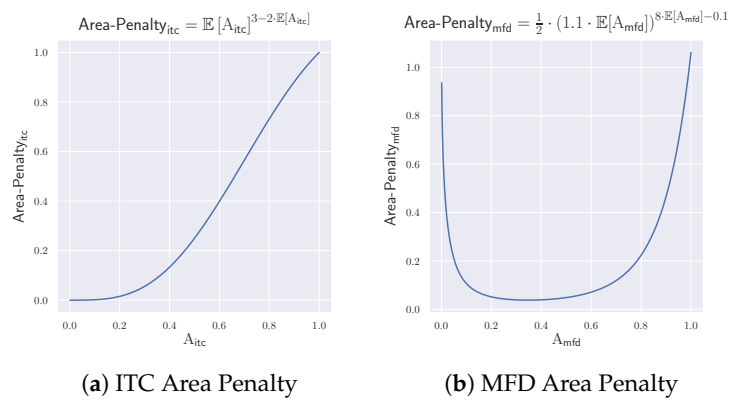


Figure 7. Soft area penalties.

$$\text{Area-Penalty}_{itc} = \mathbb{E}[A_{itc}]^{3-2 \cdot \mathbb{E}[A_{itc}]} \tag{6}$$

The overall loss on a training ITC attention model is listed in Equations (7) and (8). We use a λ factor to balance the weight between VarLoss and Area-Penalty. Figure 8 shows the effect of ITC attention on image texture complexity reduction. The attention area reaches 21.2% on average, and the weighted images gain an average of 86.3% texture reduction in the validation dataset:

$$\text{VarLoss} = \mathbb{E}[\text{VarPool2d}(A_{itc} \cdot C_\theta + (1 - A_{itc}) \cdot C)] \tag{7}$$

$$\text{Loss}_{itc} = \lambda \cdot \text{VarLoss} + (1 - \lambda) \cdot \text{Area-Penalty}_{itc} \tag{8}$$

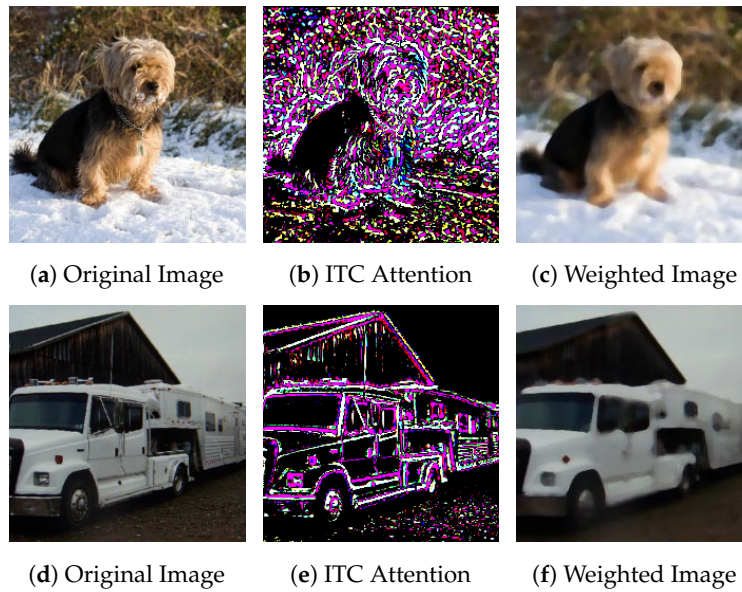


Figure 8. The effect of ITC attention on texture complexity reduction.

2.3. MFD Attention Model

MFD (Minimizing Feature Distortion) attention model aims to embed information with the least impact on neural network extracted features. Its attention also indicates the position of image pixels and their corresponding capacity that tolerate mutations.

For each cover image C , MFD model f_{mfd} needs to find an attention $A_{mfd} = f_{mfd}(C)$ that minimizes the distance between cover image features $f_{nn}(C)$ and embedded image features $f_{nn}(S)$ after embedding information into cover image according to its attention:

$$S = f_{embed}(C, A_{mfd}) \tag{9}$$

$$\text{minimize } \mathcal{L}_{fmr1}(f_{nn}(C), f_{nn}(S)) \tag{10}$$

$$\text{subject to } \alpha \leq \frac{1}{N} \sum_i^N A_{mfd} \leq \beta \tag{11}$$

Here, C stands for the cover image and S stands for the corresponding embedded image. $\mathcal{L}_{fmr1}(\cdot)$ is the feature map reconstruction loss and α, β are thresholds limiting the area of attention map acting the same role as θ in the ITC attention model.

The MFD model instances colored in purple that share the same weight. The ResNet-18 model instances colored in yellow use the same weight and are frozen.

The actual ways of training the MFD attention model is split into two phases (see Figure 9). The first training phase aims to initialize the weights of encoder blocks using the left path shown in Figure 9 as an autoencoder. In the second training phase, all the weights of decoder blocks are reset and take the right path to generate MFD attention. The encoder and decoder block architectures are shown in Figure 10.

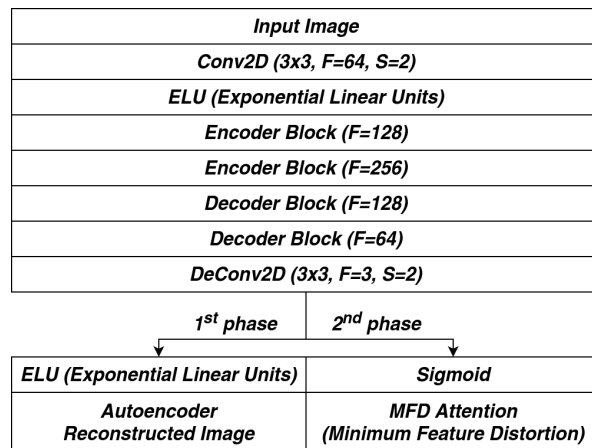


Figure 9. MFD attention model architecture.

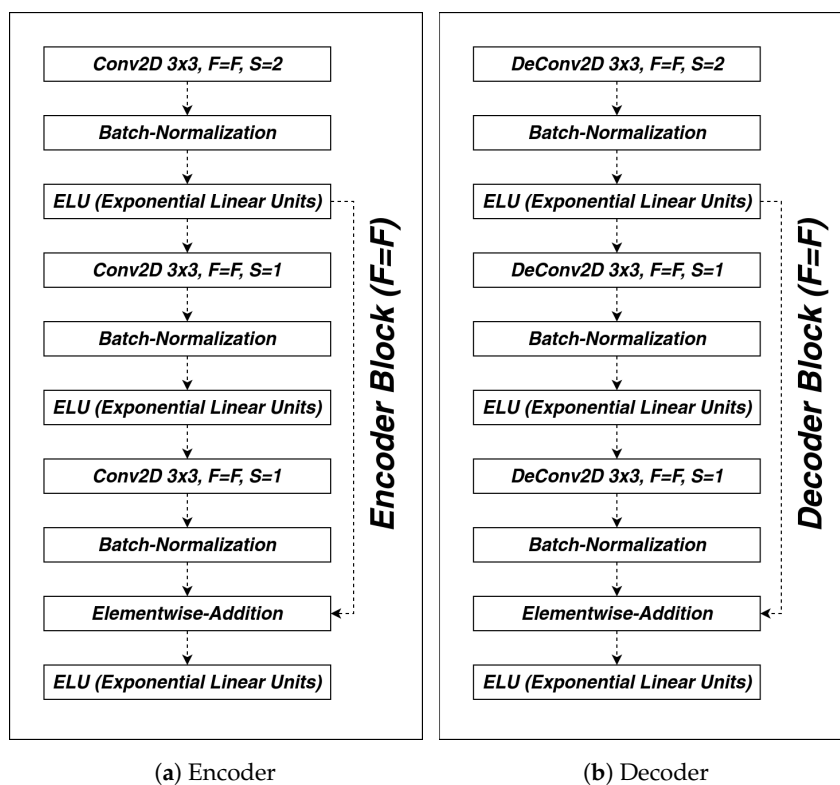


Figure 10. The encoder and decoder block of the MFD attention model.

The overall training pipeline in the second phase is shown in Figure 11. The weights of two MFD blocks colored in purple are shared while the weights of two task specific neural network blocks colored in yellow are frozen. In the training process, task specific neural network works only as a feature extractor and therefore it can be simply extended to multiple tasks by reshaping and concatenating feature maps together. Here, we adopt ResNet-18 [3] as an example for minimizing embedding distortion to the classification task.

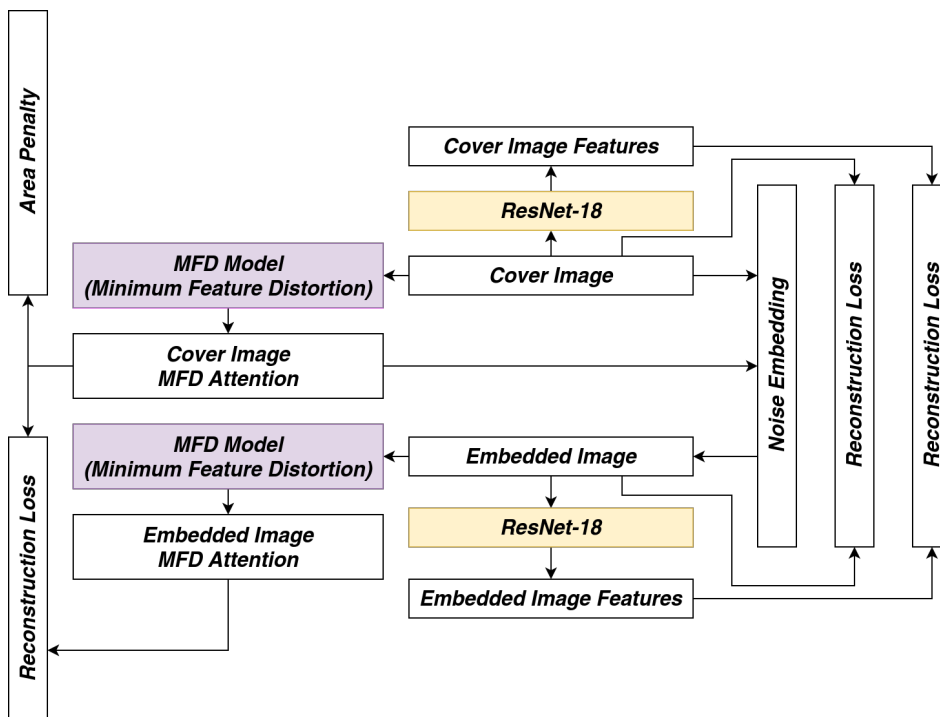


Figure 11. MFD attention mechanism training pipeline.

The overall loss on training MFD attention model (phase 2) is listed in Equation (12). The $\mathcal{L}_{\text{fmr1}}$ (Feature Map Reconstruction Loss) uses L_2 loss to reconstruct between cover image extracted feature maps and embedded ones (Equation (13)). The $\mathcal{L}_{\text{cer1}}$ (Cover Embedded Image Reconstruction Loss) and $\mathcal{L}_{\text{atr1}}$ (Attention Reconstruction Loss) uses L_1 loss to reconstruct between the cover images and the embedded images and their corresponding attention (Equation (14)). The $\mathcal{L}_{\text{atap}}$ (Attention Area Penalty) also applies a soft bound limit in forms of Equation (16) (visualized in Figure 7). The visual effect of MFD attention embedding with random noise is shown in Figure 12:

$$\text{Loss}_{\text{mfd}} = \mathcal{L}_{\text{fmr1}} + \mathcal{L}_{\text{cer1}} + \mathcal{L}_{\text{atr1}} + \mathcal{L}_{\text{atap}} \tag{12}$$

$$\mathcal{L}_{\text{fmr1}} = \mathbb{E} \left[(\text{Cover-Feature-Map} - \text{Embedded-Feature-Map})^2 \right] \tag{13}$$

$$\mathcal{L}_{\text{cer1}} = \mathbb{E} [|\text{Cover-Image} - \text{Embedded-Image}|] \tag{14}$$

$$\mathcal{L}_{\text{atr1}} = \mathbb{E} [|\text{Cover-Attention} - \text{Embedded-Attention}|] \tag{15}$$

$$\mathcal{L}_{\text{atap}} = \text{Area-Penalty}_{\text{mfd}} = \frac{1}{2} \cdot (1.1 \cdot \mathbb{E}[A_{\text{mfd}}])^{8 \cdot \mathbb{E}[A_{\text{mfd}}] - 0.1} \tag{16}$$



Figure 12. The visual effect of MFD attention on embedding with random noise.

3. Fusion Strategies, Fine-Tuning Process, and Inference Techniques

The fusion strategies help merge ITC and MFD attention models into one attention model, and thus they are substantial to be consistent and stable. In this paper, two fusion strategies being minima fusion and mean fusion are put forth as Equations (17) and (18). The minima fusion strategy aims to improve security while the mean fusion strategy generates more payload capacity for embedding:

$$A_f = \min(A_{itc}, A_{mfd}) \tag{17}$$

$$A_f = \frac{1}{2}(A_{itc}, A_{mfd}) \tag{18}$$

After a fusion strategy is applied, the fine-tuning process is required to improve attention reconstruction on embedded images. The fine-tuning process is split into two phases. In the first phase, the ITC model is fine-tuned as Figure 13. The two ITC model instances colored in purple share the same network weights, and the MFD model weights are frozen. Besides from the ITC variance loss (Equation (7)) and the ITC area penalty (Equation (6)), the loss additionally involves an attention reconstruction loss using L_1 loss similar to \mathcal{L}_{atrl} in Equation (12). In the second phase, the new ITC model from the first phase is frozen, while the MFD model is fine-tuned using its original loss (Equation (12)) as Figure 14. The use of color has the same meaning as that of the first phase.

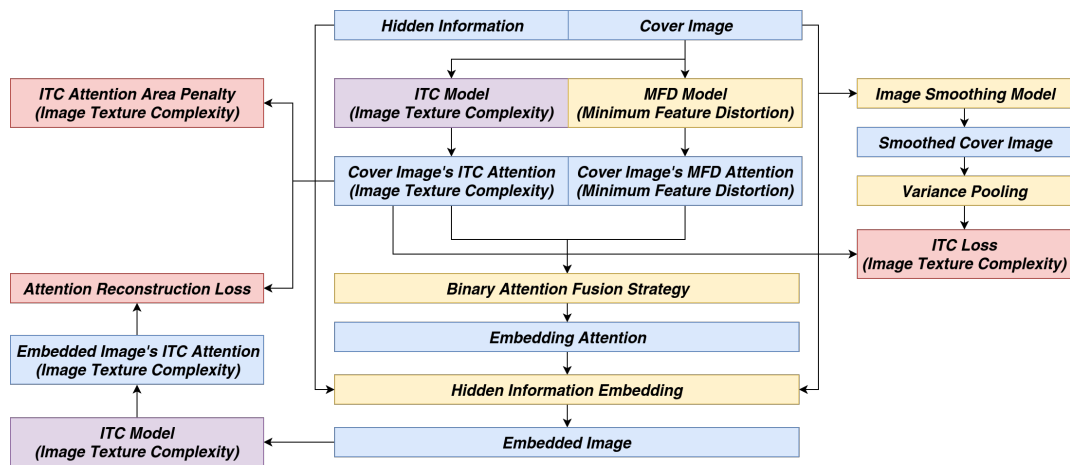


Figure 13. The 1st phase fine-tune pipeline.

All the loss functions are colored in red. The fine-tune target model is colored in purple and all the instances share the same weight.

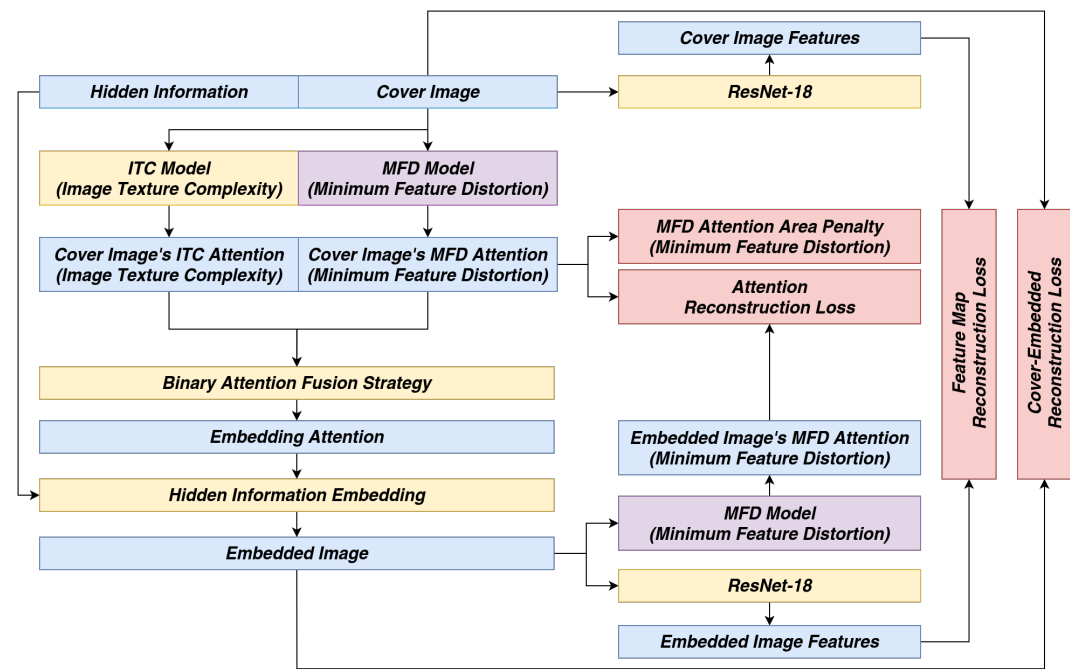


Figure 14. The 2nd phase fine-tune pipeline.

All the loss's ITC Attention are colored in red. The fine-tune target model is colored in purple and all the instances share the same weight.

The ITC model, after fine-tuning, appears to be more interested in the texture-complex areas while ignoring the areas that might introduce noises into the attention (see Figure 15).

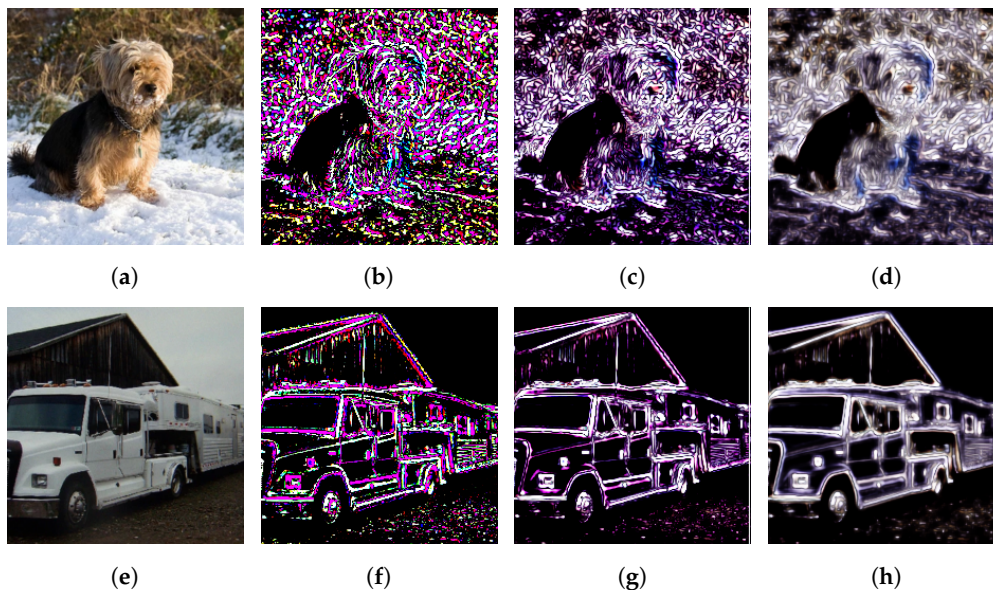


Figure 15. ITC attention after fine-tuning.

The first column shows the original image, the second column shows the ITC attention before any fine-tuning, the third column shows the ITC attention after fine-tuning for a minima fusion strategy, and the fourth column shows the ITC attention after fine-tuning for the mean fusion strategy.

When using the model for inference after fine-tuning, two extra techniques are proposed to strengthen steganography security. The first technique is named *Least Significant Masking (LSM)* which masks the lowest several bits of the attention during embedding. After the hidden information is embedded, the masked bits are restored to the original data to disturb the steganalysis methods. The second technique is called *Permutative Straddling*, which sacrifices some payload capacity to straddle between hidden bits and cover bits [26]. It is achieved by scattering the effective payload bit locations across the overall embedded locations using a random seed. The overall hidden bits are further re-arranged sequentially in the effective payload bit locations. The random seed is required to restore the hidden data.

4. Experiments

4.1. Experiment Configurations

To demonstrate the effectiveness of our model, we conducted experiments on an ImageNet dataset [27]. In particular, an ILSVRC2012 dataset with 1,281,167 images is used for training and 50,000 for testing. Our work is trained on one Nvidia GTX1080 GPU, and we adopt a batch size of 32 for all models. Optimizers and learning rate setup for the ITC model, the MFD model 1st phase, and the MFD model 2nd phase are the Adam optimizer [28] with 0.01, Nesterov momentum optimizer [29] with 1×10^{-5} , and Adam optimizer with 0.01, respectively.

All of the validation processes use the compressed version of *The Complete Works of William Shakespeare* [30] provided by Project Gutenberg [31]. It can be downloaded here at [32].

The error rate uses BSER (Bit Steganography Error Rate) shown in Equation (19).

$$BSER = \frac{\text{Number of redundant bits or missing bits}}{\text{Number of hidden information bits}} \times 100\% \tag{19}$$

4.2. Different Embedding Strategies Comparison

Table 1 presents a performance comparison among different fusion strategies and different inference techniques. These techniques offer several ways to trade off between error rate and payload

capacity. Figure 16 visualizes the fused attention and its corresponding embedding results of mean fusion strategy with 1-bit *Least Significant Masking*. Even with Mean-LSM-1 strategy, a strategy with most payload capacity, the embedded image arouses little visual awareness of the hidden information. Moreover, with *Permutative Straddling*, it is further possible to precisely handle the payload capacity during transmission. Just as shown in Table 1, the payload of Mean-LSM-1 and Mean-LSM-2 are both controlled down to 1.2 bpp.

Table 1. Different embedding strategies comparison.

Model	BSER (%)	Payload (bpp)
Min-LSM-1	1.06%	1.29
Min-LSM-2	0.67%	0.42
Mean-LSM-1	2.22%	3.89
Mean-LSM-2	3.14%	2.21
Min-LSM-1-PS-0.6	0.74%	0.60
Min-LSM-1-PS-0.8	0.66%	0.80
Mean-LSM-1-PS-1.2	0.82%	1.20
Mean-LSM-2-PS-1.2	0.93%	1.20

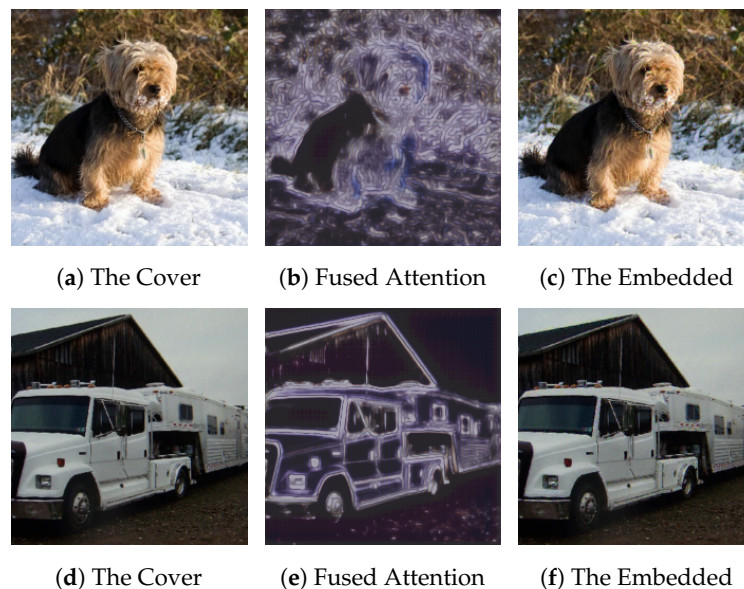


Figure 16. Steganography using Mean Fusion with 1-bit LSM.

In the model name part, the value after LSM is the number of bits masked during embedding process and the value after PS is the maximum payload capacity the embedded image is limited to during permutative straddling.

4.3. Steganalysis Experiments

To ensure that our model is robust to steganalysis methods, we test our models using StegExpose [33] with linear interpolation of detection threshold from 0.00 to 1.00 with 0.01 as the step interval. The ROC curve is shown in Figure 17 where true positive stands for an embedded image correctly identified in which there are hidden data inside while a false positive means that a clean figure is falsely classified as an embedded image. The green solid line with a slope of 1 is the baseline of an intuitive random guessing classifier. The figure shows a comparison among our several models, StegNet [21] and Baluja-2017 [22] plotted in dash-line-connected scatter data. It demonstrates that StegExpose can only work a little better than random guessing and most BASN models perform better than StegNet and Baluja-2017.

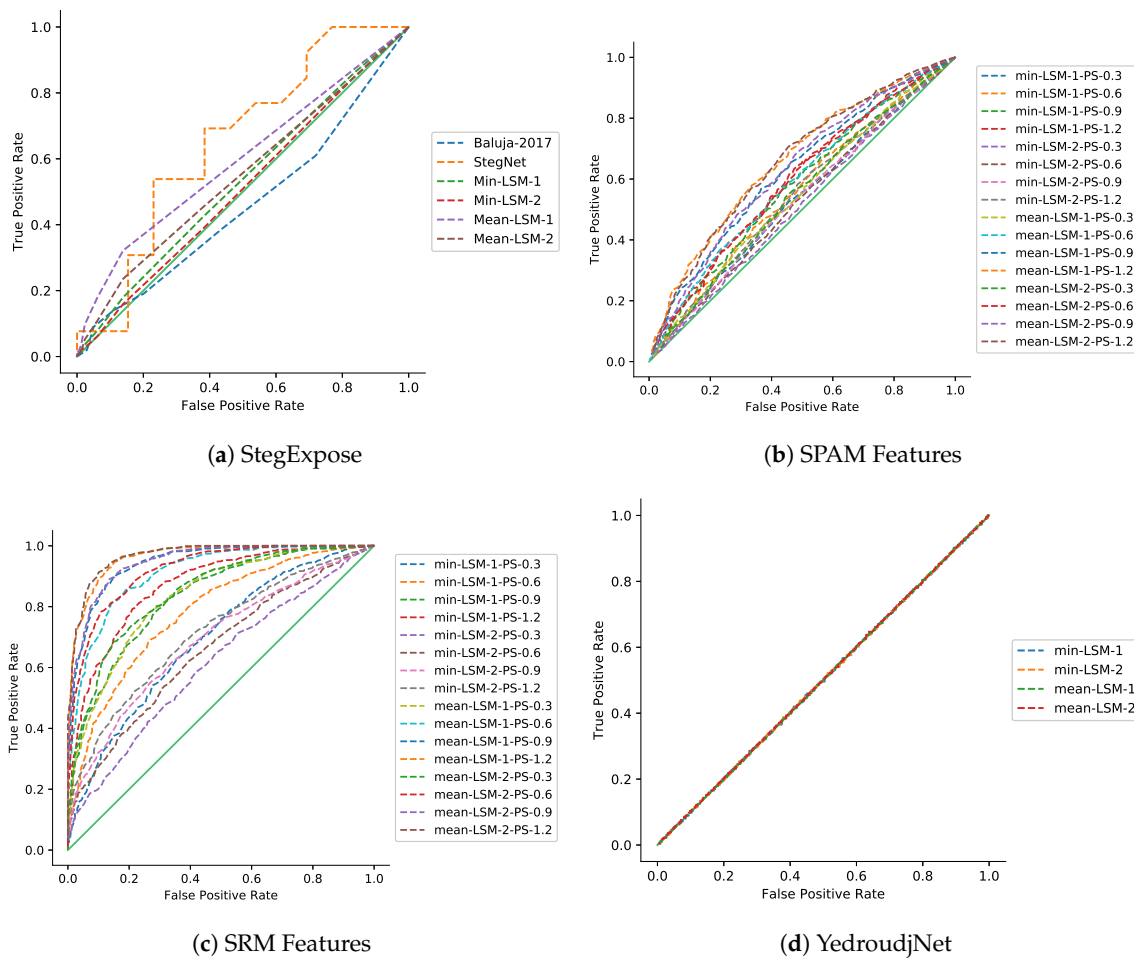


Figure 17. ROC curves: Steganalysis with StegExpose, SPAM features, SRM features and Yedroudj-Net.

Our model is also further examined with learning-based steganalysis methods including SPAM (Subtractive Pixel Adjacency Model) [34], SRM (Spatial Rich Model) [35], and YedroudjNet [36]. All of these models are trained with the same cover and embedded images as ours. Their corresponding ROC curves are shown in Figure 17. The SRM [35] method works quite well on our model with a larger payload capacity; however, in real-world applications, we can always keep our dataset private and thus ensure high security in resisting detection from learning-based steganalysis methods.

4.4. Feature Distortion Analysis

Figure 18 is a histogram of the feature distortion rate before and after hidden information embedding, or namely the impact of steganography against the network’s original task. A more concentrated distribution in the middle of the diagram indicates better preservation of the neural network’s original features and, as a result, a more consistent task result is ensured after steganography. As we can see in Figure 18, our model has little influence on the targeted neural-network-automated tasks, which, in this case, is classification. Even with the Mean-LSM-1 strategy, images that carry more than 3 bpp of hidden information are still very concentrated and take an average of only 2% of distortion.

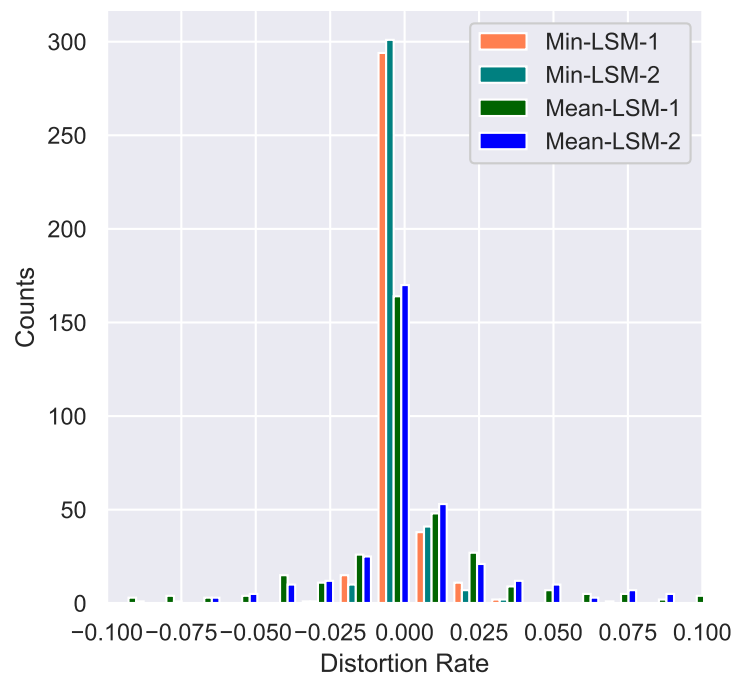


Figure 18. ResNet-18 classification feature distortion rate.

5. Conclusions

This paper proposes an image steganography method based on a binary attention mechanism to ensure that steganography has little influence on neural-network-automated tasks. The first attention mechanism, the image texture complexity (ITC) model, help track down the pixel locations and their tolerance of modification without being noticed by the human visual system. The second mechanism, the minimizing feature distortion (MFD) model, further keeps down the embedding impact through feature map reconstruction. Moreover, some attention fusion and fine-tuning techniques are also proposed in this paper to improve security and hidden information extraction accuracy. The imperceptibility of secret information by our method is proved such that the embedding images can effectively resist detection by several steganalysis algorithms. The major drawback of our approach is that it requires a lot of GPU memory and quite a long time for training and fine-tuning before it can be applied in evaluation.

With the increase of deep learning, more applications will appear geared with neural networks. We believe that hiding from neural-network-automated tasks in addition to human visual systems is occupying a more essential position. Plenty of future works need to be done in this direction including improving the performance of the steganography model to run in real time along with other automated tasks, sustaining the hidden information in the embedded image even after being post-processed by other neural-network-automated tasks like image super-resolution, etc.

Author Contributions: Conceptualization, P.W.; Methodology, X.C. and Y.Y.; Resources, X.L.; Writing—original draft, Y.Y.; Writing—review & editing, X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Natural Science Foundation of Shanghai grant number 19ZR1417700.

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

References

1. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
2. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
4. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First, AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
5. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
6. Zhong, Z.; Zheng, L.; Zheng, Z.; Li, S.; Yang, Y. Camera style adaptation for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 5157–5166.
7. Mielikainen, J. Lsb matching revisited. *IEEE Signal Proc. Lett.* **2006**, *13*, 285–287. [[CrossRef](#)]
8. Spaulding, J.; Noda, H.; Shirazi, M.N.; Kawaguchi, E. BPCS steganography using EZW lossy compressed images. *Pattern Recognit. Lett.* **2002**, *23*, 1579–1587. [[CrossRef](#)]
9. Sun, S. A new information hiding method based on improved BPCS steganography. *Adv. Multimed.* **2015**, *5*. [[CrossRef](#)]
10. Wu, D.C.; Tsai, W.H. A steganographic method for images by pixel-value differencing. *Pattern Recognit. Lett.* **2003**, *24*, 1613–1626. [[CrossRef](#)]
11. Wu, H.C.; Wu, N.I.; Tsai, C.S.; Hwang, M.S. Image steganographic scheme based on pixel-value differencing and LSB replacement methods. *IEE Proc. Vis. Image Signal Proc.* **2005**, *152*, 611–615. [[CrossRef](#)]
12. Wang, C.M.; Wu, N.I.; Tsai, C.S.; Hwang, M.S. A high quality steganographic method with pixel-value differencing and modulus function. *J. Syst. Softw.* **2008**, *81*, 150–158. [[CrossRef](#)]
13. Pevný, T.; Filler, T.; Bas, P. Using high-dimensional image models to perform highly undetectable steganography. In Proceedings of the International Workshop on Information Hiding, Calgary, AB, Canada, 28–30 June 2010; pp. 161–177.
14. Holub, V.; Fridrich, J.; Denemark, T. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Inf. Secur.* **2014**, *2014*, 1. [[CrossRef](#)]
15. Wang, L.; Tan, X.; Xu, Y.; Zhai, L. An Adaptive JPEG Image Steganography Based on Texture Complexity. *J. Wuhan Univ. (Nat. Sci. Ed.)* **2017**, *63*, 421–426.
16. Meng, R.; Rice, S.G.; Wang, J.; Sun, X. A fusion steganographic algorithm based on faster R-CNN. *Comput. Mater. Contin.* **2018**, *55*, 1–16.
17. Xue, Y.; Liu, W.; Lu, W.; Yeung, Y.; Liu, X.; Liu, H. Efficient halftone image steganography based on dispersion degree optimization. *J. Real-Time Image Proc.* **2019**, *16*, 601–609. [[CrossRef](#)]
18. Huang, J.; Cheng, S.; Lou, S.; Jiang, F. Image steganography using texture features and GANs. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
19. Liao, X.; Yin, J. Two embedding strategies for payload distribution in multiple images steganography. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 1982–1986.
20. Hu, D.; Wang, L.; Jiang, W.; Zheng, S.; Li, B. A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access* **2018**, *6*, 38303–38314. [[CrossRef](#)]
21. Wu, P.; Yang, Y.; Li, X. Image-into-Image Steganography Using Deep Convolutional Network. In Proceedings of the 2018 Pacific Rim Conference on Multimedia, Hefei, China, 21–22 September 2018; pp. 792–802.
22. Baluja, S. Hiding images in plain sight: Deep steganography. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 2069–2079.

23. Riesenhuber, M.; Poggio, T. Hierarchical models of object recognition in cortex. *Nat. Neurosci.* **1999**, *2*, 1019–1025. [[CrossRef](#)] [[PubMed](#)]
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 25, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
25. Zhang, X.; Lin, W.; Xue, P. Just-noticeable difference estimation with pixels in images. *J. Vis. Commun. Image Represent.* **2008**, *19*, 30–41. [[CrossRef](#)]
26. Westfeld, A. F5—A Steganographic Algorithm. In *Information Hiding*; Moskowitz, I.S., Ed.; Springer: Berlin, Germany, 2001; pp. 289–302.
27. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009.
28. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the importance of initialization and momentum in deep learning. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013; pp. 1139–1147.
30. Shakespeare, W. *The Complete Works of William Shakespeare*; Wordsworth Editions Ltd.: Hertfordshire, UK, 1997.
31. Gutenberg, P. Project Gutenberg. Available online: <https://www.gutenberg.org/> (accessed on 13 November 2018).
32. Gutenberg, P. The Complete Works of William Shakespeare by William Shakespeare. Available online: <http://www.gutenberg.org/ebooks/100> (accessed on 13 November 2018).
33. Boehm, B. StegExpose—A Tool for Detecting LSB Steganography. *arXiv* **2014**, arXiv:1410.6656.
34. Pevny, T.; Bas, P.; Fridrich, J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 215–224. [[CrossRef](#)]
35. Fridrich, J.; Kodovsky, J. Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882. [[CrossRef](#)]
36. Yedroudj, M.; Comby, F.; Chaumont, M. Yedroudj-Net: An Efficient CNN for Spatial Steganalysis. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 2092–2096.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).