MDPI

*Article*

# BASN — Learning Steganography with Binary Attention Mechanism

**Pin Wu [1], Xuting Chang [1], Yang Yang [1] and Xiaoqiang Li [1,2]***

[1] School of Computer Science, Shanghai University, China
[2] Shanghai Institute for Advanced Communication & Data Science, Shanghai University, China
* Correspondence: xqli@shu.edu.cn

Version January 15, 2020 submitted to Future Internet

**Abstract:** Secret information sharing through image carrier 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 empowered more deeply in our daily lives. However, a series of wrong labeling or bad captioning on the embedded images leaves a trace of skepticism and finally leads to a self-confession alike 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 meanwhile, 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 meanwhile, 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] and etc. Images 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-Matching [7] steganography completely alters the image classification result from goldfish to proboscis monkey. 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. [1]

Most previous steganography models focus on resisting steganalysis algorithms or raising embedding payload capacity. BPCS [8,9] and PVD [10–12] uses adaptive embedding based on local

[1] Source code will be published at: https://github.com/adamcavendish/BASN-Learning-Steganography-with-Binary-Attention-Mechanism

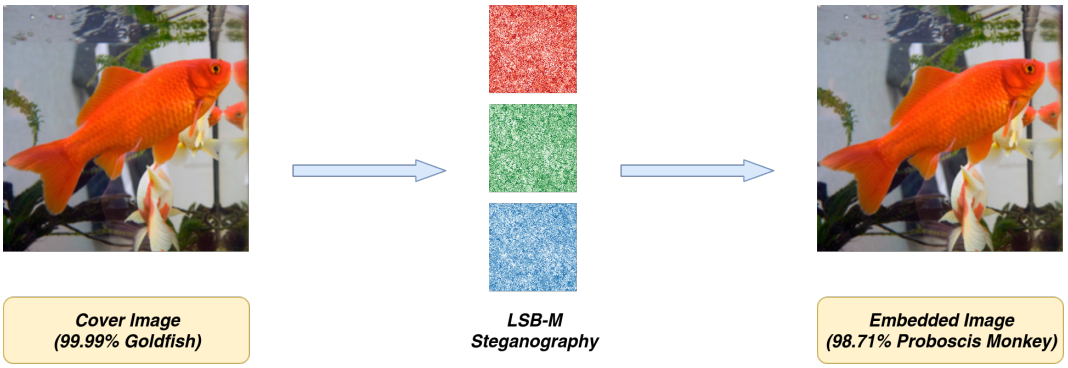



**Figure 1.** LSB-Matching Embedded Image Misclassification

The cover image and embedded image both use ImageNet pretrained ResNet-18 [3] network for classification. The percentage before the predicted class label represents 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're 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.

complexty to improve embedding visual quality. HuGO [13] and S-UNIWARD [14] resist steganalysis by minimizing a suitably defined distortion function. Hu [15] adopts deep convolutional generative adversarial network to achieve steganography without embedding. Wu [16] and Baluja [17] 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 minimizes 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.

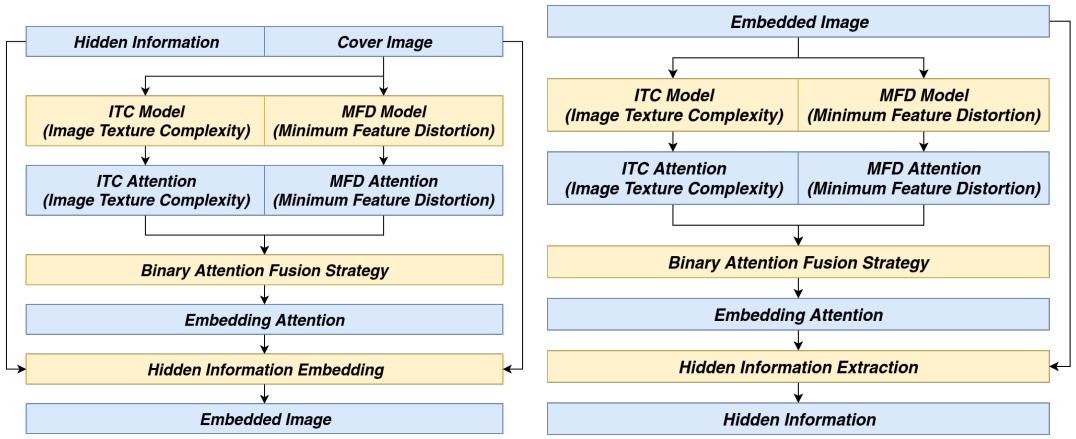
## 2. Binary Attention Mechanism

Binary attention mechanism involves two attention models including image texture complexity (ITC) attention model and minimizing feature distortion (MFD) attention model. The attention mechanism in both models serve as a hint for steganography showing where to embed or extract and how much information the corresponding pixel might tolerate. ITC model mainly focuses on deceiving the human visual system from noticing the differences out of altered pixels. 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 MFD model, we align the latent space of the cover image and the embedded image, which therefore recreating or inferring from the embedded image what attention or how much capacity was available in the original cover image is possible.

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

### *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.




**(a)** Embedding **(b)** Extraction

**Figure 2.** The Embedding and Extraction Architecture

We propose 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.

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

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

In Equation 1, $X$ is a 2-dimensional random variable which can be either an image or a feature map and $i, j$ are the indices of each dimension. Operator $\text{E}(\cdot)$ calculates the expectation of the random variable. VarPool2d applies similar kernel mechanism as other 2-dimensional pooling or convolution operations [18,19] 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 influences the image itself during training (See Equation 2,3 for training loss and Figure 3 for the impact comparison).

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

$$\mathcal{L}_{\text{VarPool2d}} = \text{E}\left(\text{VarPool2d}_{n=7}\left(X\right)\right) \tag{3}$$

### 2.2. ITC Attention Model

ITC (Image Texture Complexity) attention model aims to embed information without being noticed by the human visual system, or in other words, making just noticeable difference (JND) to cover images to ensure the largest embedding payload capacity [20]. 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 $\theta$ representing the ideal embedding payload capacity that the input image might achieve.

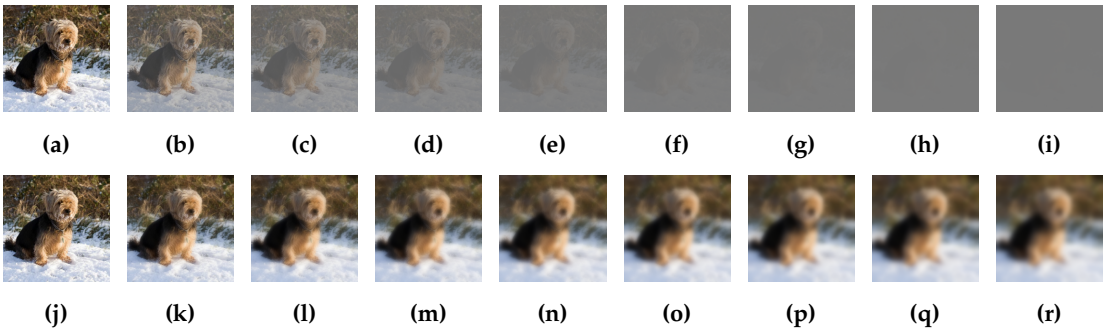



**Figure 3.** The gradient impact comparison between variance and variance pooling during training

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.

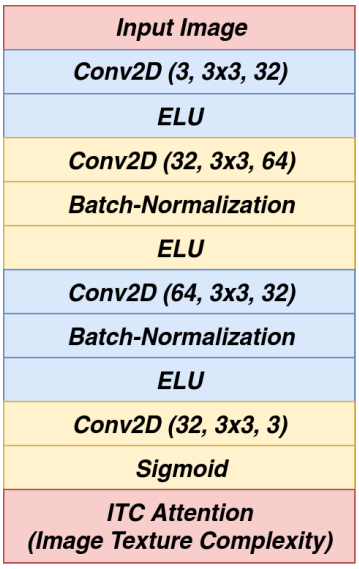



**Figure 4.** ITC Attention Model Architecture

2. An ideal texture-free image $C_\theta$ corresponding to the input image that is visually similar but with the lowest texture complexity possible regarding the restriction of at most $\theta$ changes.

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

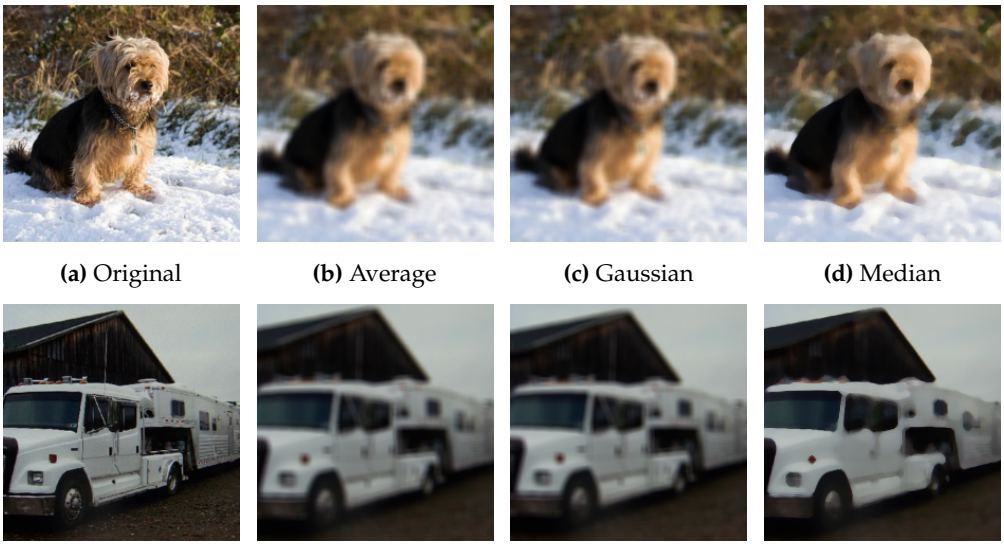
For each cover image $C$, ITC model $f_{\text{itc}}$ needs to find an attention $A_{\text{itc}} = f_{\text{itc}}(C)$ to minimize the texture complexity evaluation function $V_{\text{itc}}$:

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

$$\text{subject to} \quad \frac{1}{N}\sum_{i}^{N} A_{\text{itc}} \leq \theta \tag{5}$$

The $\theta$ in Equation 5 is used as an upper bound to limit down the attention area size. If trained without it, model $f_{\text{itc}}$ is free to output all-ones matrix $A_{\text{itc}}$ 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.

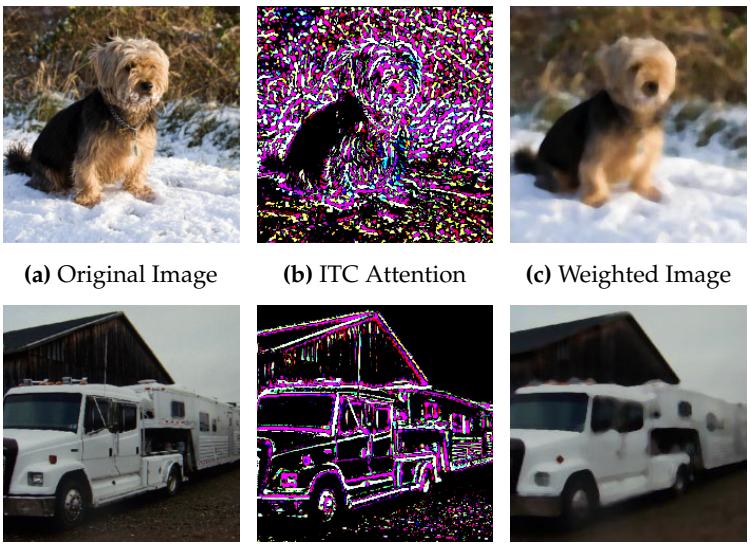
In actual training process, the detailed model architecture is shown in Figure 4 and two parts of the equation are slightly modified to ensure better training results. First, the ideal texture-free image $C_\theta$ in Equation 4 does not indeed exist but is available through approximation nonetheless. In this paper median pooling with a kernel size of 7 is used to simulate the ideal texture-free image. It



**(a)** Original **(b)** Average **(c)** Gaussian **(d)** Median

**(e)** Original **(f)** Average **(g)** Gaussian **(h)** Median

**Figure 5.** Image Smoothing Effect Comparison



**(a)** Original Image **(b)** ITC Attention **(c)** Weighted Image

**(d)** Original Image **(e)** ITC Attention **(f)** Weighted Image

**Figure 6.** The Effect of ITC Attention on Texture Complexity Reduction

helps eliminate detailed textures within patches without touching object boundaries (See Figure 5 for comparison among different smoothing techniques). Second, we adopt soft bound limits in place of hard upper bound in forms of Equation 6 (visualized in Figure 10). Soft limits help generate smoothed gradients and provide optimizing directions.

$$\text{Area-Penalty}_{\text{itc}} = \text{E}(A_{\text{itc}})^{3-2\cdot\text{E}(A_{\text{itc}})} \tag{6}$$

The overall loss on training ITC attention model is listed in Equation 7,8, and Figure 6 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.

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

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

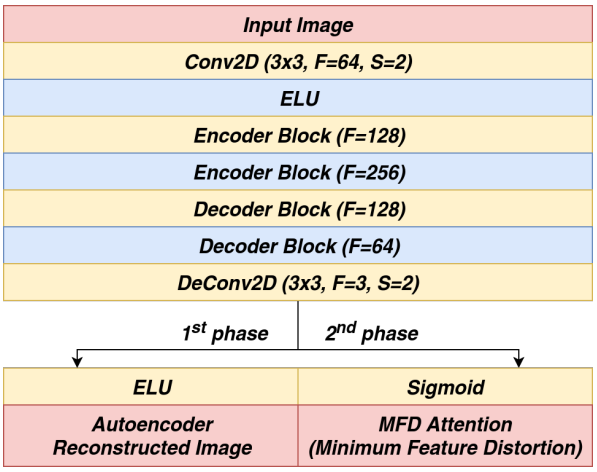



**Figure 7.** MFD Attention Model Architecture

### 2.3. MFD Attention Model

MFD (Minimizing Feature Distortion) attention model aims to embed information with 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 $f_{\text{mfd}}$ needs to find an attention $A_{\text{mfd}} = f_{\text{mfd}}(C)$ that minimizes the distance between cover image features $f_{\text{nn}}(C)$ and embedded image features $f_{\text{nn}}(S)$ after embedding information into cover image according to its attention.

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

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

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

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

The actual ways of training the MFD attention model is split into 2 phases (See Figure 7). The first training phase aims to initialize the weights of encoder blocks using the left path shown in Figure 7 as an autoencoder. In the second training phase, all the weights of decoder blocks are reset and takes the right path to generate MFD attentions. The encoder and decoder block architectures are shown in Figure 9.

The overall training pipeline in the second phase is shown in Figure 8. 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.

The overall loss on training MFD attention model (phase 2) is listed in Equation 12. The $\mathcal{L}_{\text{fmrl}}$ (Feature Map Reconstruction Loss) uses $L_2$ loss to reconstruct between cover image extracted feature maps and embedded ones. The $\mathcal{L}_{\text{cerl}}$ (Cover Embedded image Reconstruction Loss) and $\mathcal{L}_{\text{atrl}}$ (Attention Reconstruction Loss) uses $L_1$ loss to reconstruct between the cover images and the embedded images and their corresponding attentions. The $\mathcal{L}_{\text{atap}}$ (ATtention Area Penalty) also applies soft bound limit

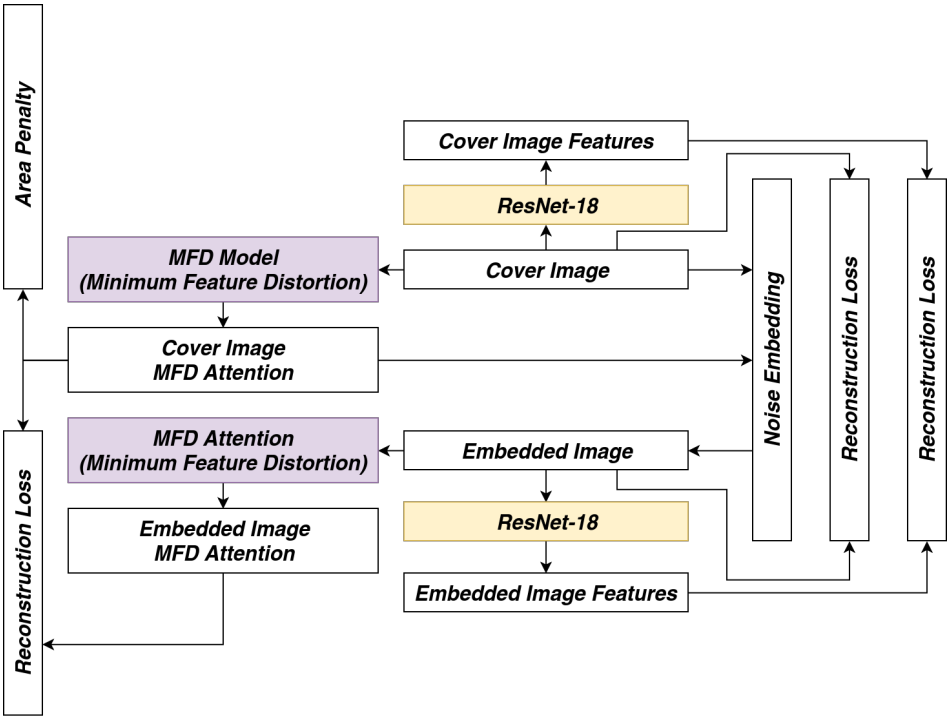
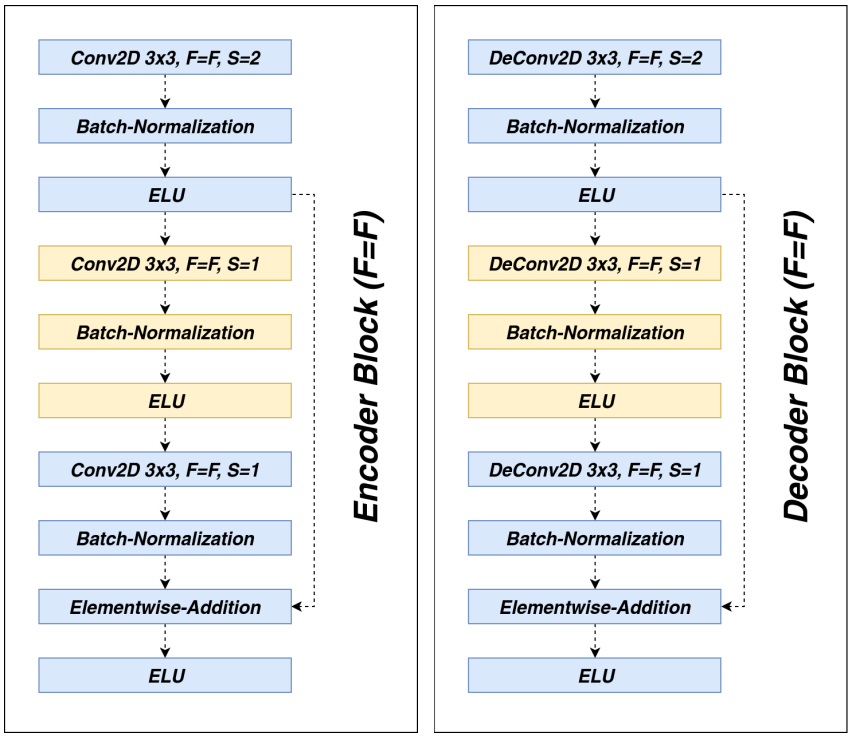
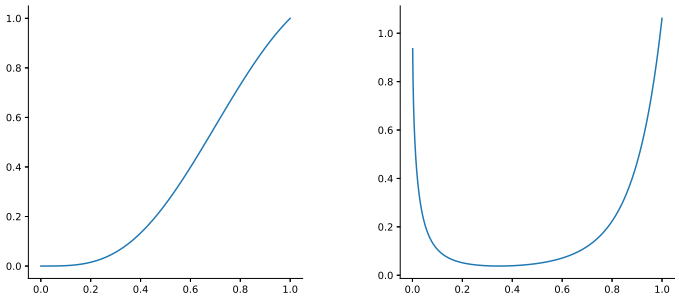



**Figure 8.** MFD Attention Mechanism Training Pipeline



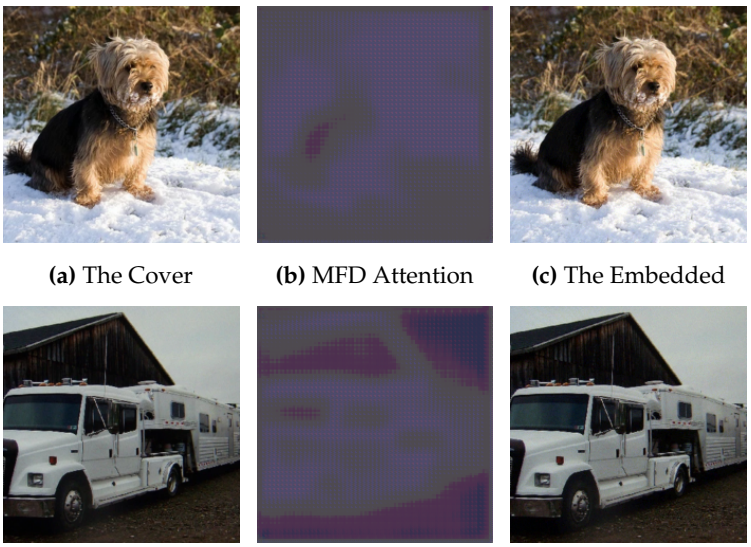

**(a)** Encoder **(b)** Decoder

**Figure 9.** The Encoder and Decoder Block of the MFD Attention Model




**(a)** ITC Area Penalty **(b)** MFD Area Penalty

**Figure 10.** Soft Area Penalties



**(a)** The Cover **(b)** MFD Attention **(c)** The Embedded

**(d)** The Cover **(e)** MFD Attention **(f)** The Embedded

**Figure 11.** The Visual Effect of MFD Attention on Embedding with Random Noise

in forms of Equation 13 (visualized in Figure 10). The visual effect of MFD attention embedding with random noise is shown in Figure 11.

$$\text{Loss}_{\text{mfd}} = \mathcal{L}_{\text{fmrl}} + \mathcal{L}_{\text{cerl}} + \mathcal{L}_{\text{atrl}} + \mathcal{L}_{\text{atap}} \tag{12}$$

$$\text{Area-Penalty}_{\text{mfd}} = \frac{1}{2} \cdot (1.1 \cdot \text{E}(A_{\text{mfd}}))^{8 \cdot \text{E}(A_{\text{mfd}}) - 0.1} \tag{13}$$

## 3. Fusion Strategies, Finetune 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 Equation 14 and 15. Minima fusion strategy aims to improve security while mean fusion strategy generates more payload capacity for embedding.

$$A_{\text{f}} = \min(A_{\text{itc}}, A_{\text{mfd}}) \tag{14}$$

$$A_{\text{f}} = \frac{1}{2}(A_{\text{itc}}, A_{\text{mfd}}) \tag{15}$$

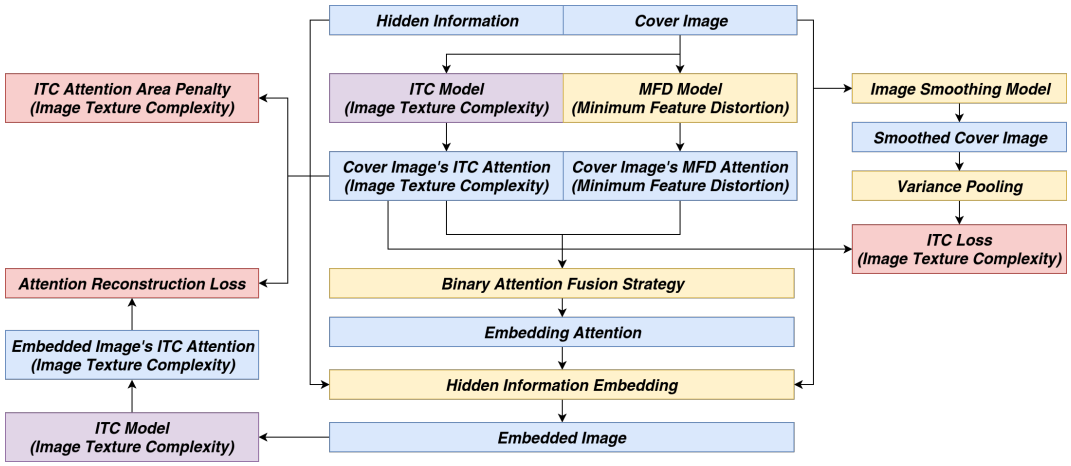



**Figure 12.** The $1^{st}$ Phase Finetune Pipeline

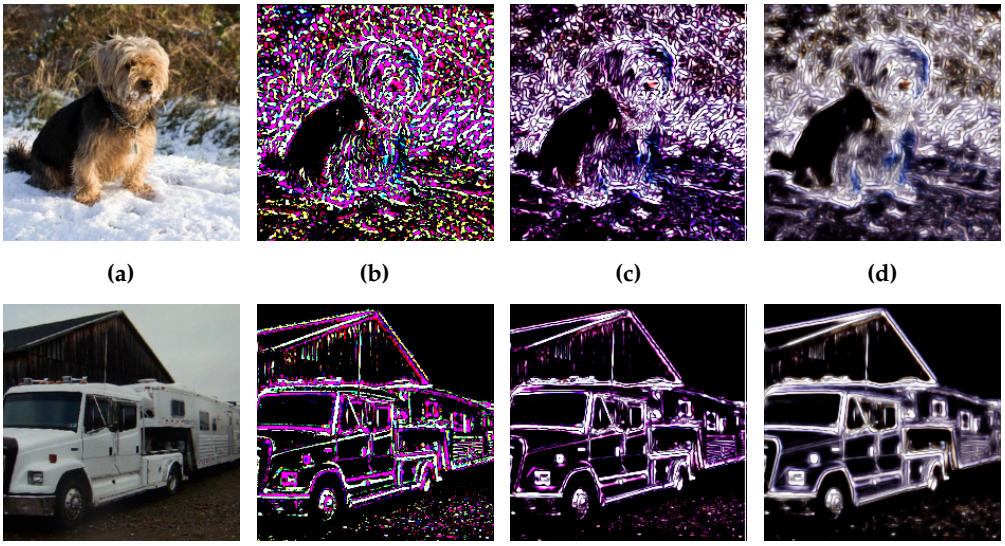



**Figure 13.** ITC Attention After Finetune

The first column shows the original image, the second column shows the ITC attention before any finetune, the third column shows the ITC attention after finetuning for minima fusion strategy, and the forth column shows the ITC attention after finetuning for mean fusion strategy.

After a fusion strategy is applied, finetuning process is required to improve attention reconstruction on embedded images. The finetune process is split into two phases. In the first phase, the ITC model is finetuned as Figure 12. The two ITC model colored in purple shares the same network weights and the MFD model weights are freezed. Besides from the image texture complexity loss (Equation 7) and the ITC area penalty (Equation 6), the loss additionally involves an attention reconstruction loss using $L_1$ loss similar to $\mathcal{L}_{\text{atrl}}$ in Equation 12. In the second phase, the new ITC model is freezed, and the MFD model is finetuned using its original loss (Equation 12).

The ITC model, after finetune, appears to be more interested in the texture-complex areas while ignores the areas that might introduce noises into the attention (See Figure 13).

When using the model for inference after finetuning, 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

| 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 |

**Table 1.** Different Embedding Strategies Comparison

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.

between hidden bits and cover bits [21]. 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. Experiments Configurations*

To demonstrate the effectiveness of our model, we conducted experiments on ImageNet dataset [22]. Specially, 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 ITC model, MFD model $1^{st}$ phase and MFD model $2^{nd}$ phase are Adam optimizer [23] with 0.01, Nesterov momentum optimizer [24] with 1e-5 and Adam optimizer with 0.01 respectively.

All the validation processes use the compressed version of *The Complete Works of William Shakespeare* [25] provided by Project Gutenberg [26]. It is downloaded here at [27].

The error rate uses BSER (Bit Steganography Error Rate) shown in Equation 16.

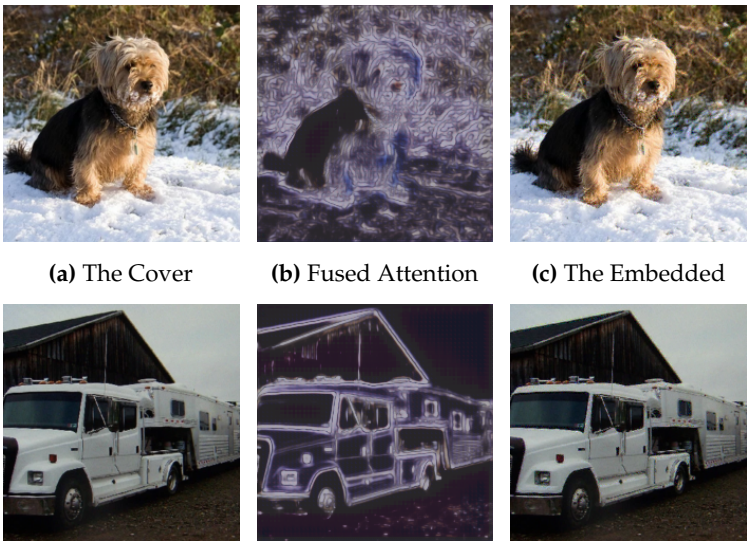
$$\text{BSER} = \frac{\text{Number of redundant bits or missing bits}}{\text{Number of hidden information bits}} \times 100\% \tag{16}$$

### *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 14 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.

### *4.3. Steganalysis Experiments*

To ensure that our model is robust to steganalysis methods, we test our models using StegExpose [28] 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 15 where true positive stands for an embedded image correctly identified that there are hidden data inside while false positive means that a clean figure



**(a)** The Cover **(b)** Fused Attention **(c)** The Embedded

**(d)** The Cover **(e)** Fused Attention **(f)** The Embedded

**Figure 14.** Steganography using Mean Fusion with 1-bit LSM

is falsely classified as an embedded image. The green solid line with slope of 1 is the baseline of an intuitive random guessing classifier. The figure shows a comparison among our several models, StegNet [16] and Baluja-2017 [17] 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.

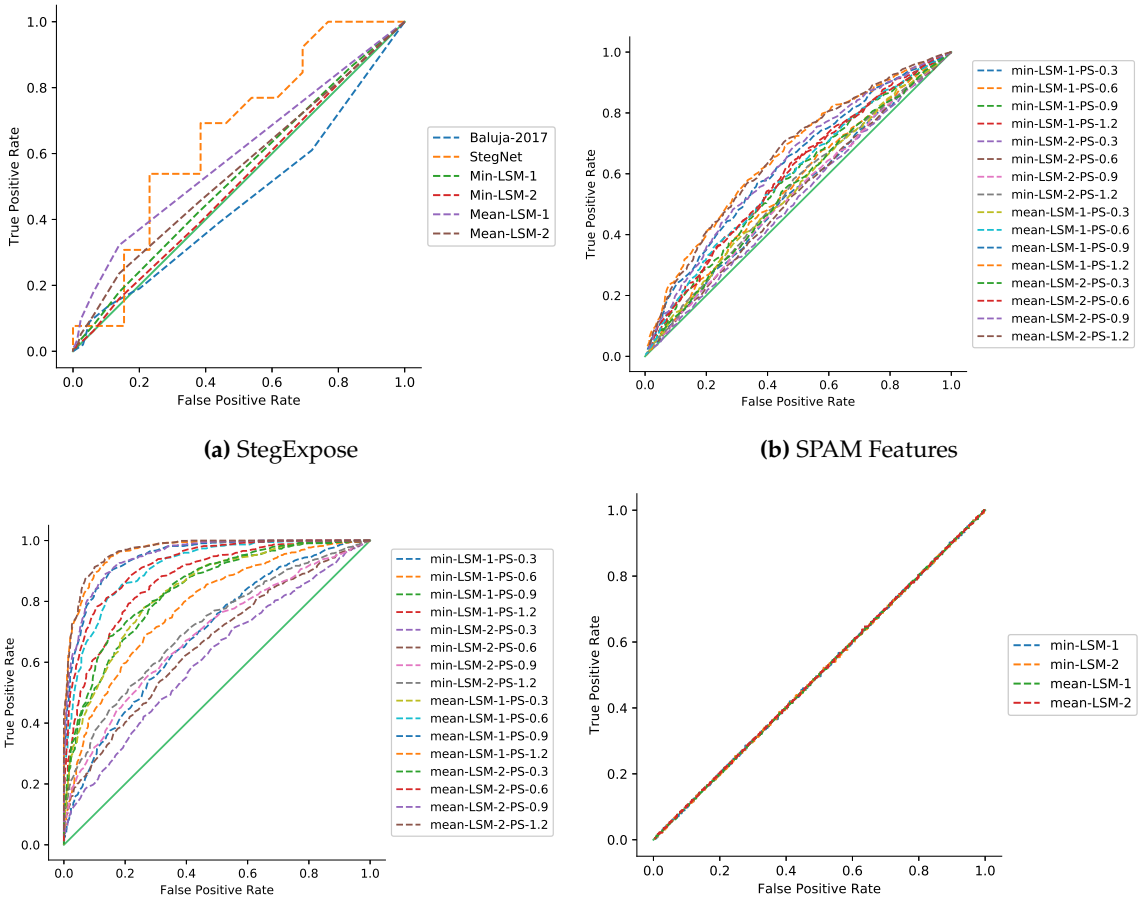
Our model is also further examined with learning-based steganalysis methods including SPAM [29], SRM [30] and YedroudjNet [31]. All of these models are trained with the same cover and embedded images as ours. Their corresponding ROC curves are shown in Figure 15. SRM [30] method works quite well on our model with a larger payload capacity, however in real-world applications we can always keep our dataset private and thus ensuring high security in resisting detection from learning-based steganalysis methods.

### *4.4. Feature Distortion Analysis*

Figure 16 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 16 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.

## 5. Conclusion

This paper proposes an image stagnography method based on a binary attention mechanism to ensure little influence steganography is made to neural-network-automated tasks. The first attention mechanism, 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, minimizing feature distortion (MFD) model, further keeps down the embedding impact through feature map reconstruction. Moreover, some attention fusion and finetune 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.




**(a)** StegExpose **(b)** SPAM Features

**(c)** SRM Features **(d)** YedroudjNet

**Figure 15.** ROC Curves: Steganalysis with StegExpose, SPAM Features, SRM Features and Yedroudj-Net

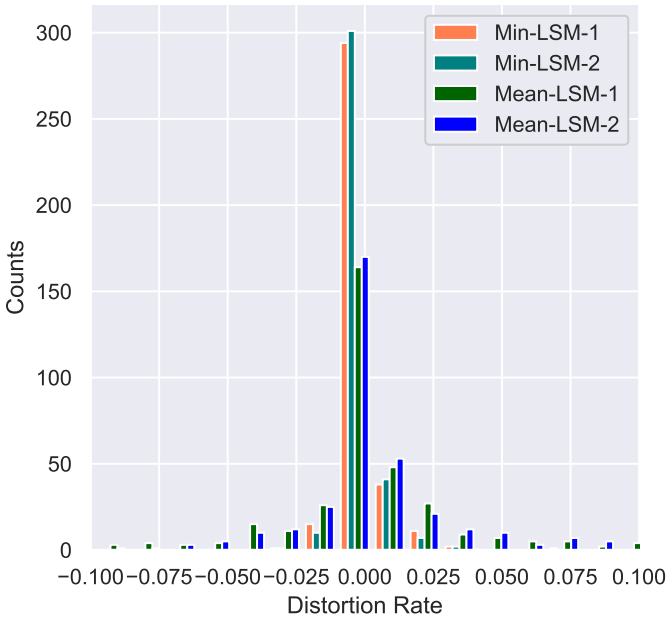



**Figure 16.** ResNet-18 Classification Feature Distortion Rate

## References


1. Girshick, R. Fast r-cnn. Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
2. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
3. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 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. Thirty-First AAAI Conference on Artificial Intelligence, 2017.
5. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.
6. Zhong, Z.; Zheng, L.; Zheng, Z.; Li, S.; Yang, Y. Camera style adaptation for person re-identification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5157–5166.
7. Mielikainen, J. Lsb matching revisited. *IEEE signal processing letters* **2006**, *13*, 285–287.
8. Spaulding, J.; Noda, H.; Shirazi, M.N.; Kawaguchi, E. BPCS steganography using EZW lossy compressed images. *Pattern Recognition Letters* **2002**, *23*, 1579–1587.
9. Sun, S. A new information hiding method based on improved BPCS steganography. *Advances in Multimedia* **2015**, *2015*, 5.
10. Wu, D.C.; Tsai, W.H. A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters* **2003**, *24*, 1613–1626.
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 Proceedings-Vision, Image and Signal Processing* **2005**, *152*, 611–615.
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. *Journal of Systems and Software* **2008**, *81*, 150–158.
13. Pevnỳ, T.; Filler, T.; Bas, P. Using high-dimensional image models to perform highly undetectable steganography. International Workshop on Information Hiding. Springer, 2010, pp. 161–177.
14. Holub, V.; Fridrich, J.; Denemark, T. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* **2014**, *2014*, 1.
15. 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.
16. Wu, P.; Yang, Y.; Li, X. Image-into-Image Steganography Using Deep Convolutional Network. Advances in Multimedia Information Processing – PCM 2018; Hong, R.; Cheng, W.H.; Yamasaki, T.; Wang, M.; Ngo, C.W., Eds.; Springer International Publishing: Cham, 2018; pp. 792–802.
17. Baluja, S. Hiding images in plain sight: Deep steganography. Advances in Neural Information Processing Systems, 2017, pp. 2069–2079.
18. Riesenhuber, M.; Poggio, T. Hierarchical models of object recognition in cortex. *Nature Neuroscience* **1999**, *2*, 1019–1025. doi:10.1038/14819.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E., ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F.; Burges, C.J.C.; Bottou, L.; Weinberger, K.Q., Eds.; Curran Associates, Inc., 2012; p. 1097–1105.
20. Zhang, X.; Lin, W.; Xue, P. Just-noticeable difference estimation with pixels in images. *Journal of Visual Communication and Image Representation* **2008**, *19*, 30–41. doi:10.1016/j.jvcir.2007.06.001.
21. Westfeld, A. F5—A Steganographic Algorithm. Information Hiding; Moskowitz, I.S., Ed.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2001; pp. 289–302.
22. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. 2009.
23. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv e-prints* **2014**. arXiv:1412.6980.
24. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the importance of initialization and momentum in deep learning. International conference on machine learning, 2013, pp. 1139–1147.
25. Shakespeare, W. *The Complete Works of William Shakespeare*; 1994.
26. Gutenberg, P. Project Gutenberg, 2018. [Online; Accessed 13-Nov-2018].

27. Gutenberg, P. The Complete Works of William Shakespeare by William Shakespeare - Free Ebook., 2018. [Online; Accessed 13-Nov-2018].
28. Boehm, B. StegExpose - A Tool for Detecting LSB Steganography. *arXiv e-prints* **2014**. arXiv: 1410.6656.
29. Pevny, T.; Bas, P.; Fridrich, J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security* **2010**, *5*, 215–224.
30. Fridrich, J.; Kodovsky, J. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security* **2012**, *7*, 868–882.
31. Yedroudj, M.; Comby, F.; Chaumont, M. Yedroudj-Net: An Efficient CNN for Spatial Steganalysis. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 2092–2096.