Pattern Recognition Letters 135 (2020) 146–152

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Steganographic universal adversarial perturbations

Salah Ud Din*, Naveed Akhtar, Shahzad Younis, Faisal Shafait, Atif Mansoor, Muhammad Shafique

School of Electrical Engineering and Computer Science (SEECS), National University of Sciences & Technology (NUST) H-12, Islamabad 44000, Pakistan

ARTICLE INFO

Article history:
Received 19 July 2019
Revised 24 March 2020
Accepted 17 April 2020
Available online 25 April 2020

Keywords:
Adversarial attack
Steganography
Deep neural networks
Wavelet transform

ABSTRACT

We propose a steganography based technique to generate adversarial perturbations to fool deep models on any image. The proposed perturbations are computed in a transform domain where a single secret image embedded in any target image makes any deep model misclassify the target image with high probability. The attack resulting from our perturbation is ideal for black-box setting, as it does not require any information about the target model. Moreover, being a non-iterative technique, our perturbation estimation remains computationally efficient. The computed perturbations are also imperceptible to humans while they achieve high fooling ratios for the models trained on large-scale ImageNet dataset. We demonstrate successful fooling of ResNet-50, VGG-16, Inception-V3 and MobileNet-V2, achieving up to 89% fooling of these popular classification models.

1. Introduction

Deep Neural Networks (DNNs) have demonstrated outstanding performance for many challenging tasks in speech recognition [1], natural language processing [2] and classification [3–5]. Convolutional Neural Networks (CNNs) based image classification [4] has revolutionized the use of deep learning in computer vision. In the last few years, researchers have been continuously devising deep learning based solutions for many complex tasks in the areas of medical science [6], safety and security [7], and self-driving vehicles [8].

As deep learning is now being also used in security-critical applications, the vulnerability of the state-of-the-art DNNs to adversarial attacks has recently attracted significant interest of researchers [9]. These attacks come in the form of a small perturbation to the input to fool the network to change its prediction altogether. At the same time, the perturbation remains imperceptible to humans. There are several existing techniques for crafting such perturbations [10–12]. These techniques either cause the network to predict a specific class for the input, i.e. targeted attack, or make it predict any incorrect output, i.e. non-targeted attack. In the domain of natural images, Moosavi-Dezfooli et al. [13] computed image-agnostic perturbations that can fool networks on any image, in contrast to the commonly used perturbations that fool deep models on individual images. However, the perturbation computed by Moosavi-Dezfooli et al. are not completely imperceptible to the Human visual system. Moreover, their technique is computationally expensive as it must iterate over a large training data to compute the perturbation. Furthermore, it requires complete information of the weights of the target network, hence it is inherently not suitable for black-box settings where this information is not available.

In this paper, we propose another kind of image-agnostic (i.e. universal) perturbation that is computed efficiently and also does not require any network information. Moreover, it remains imperceptible to Humans, see Fig. 1. We leverage steganography to compute the desired perturbations, where we hide a secret image inside the image to be classified (i.e. host image). The perturbation is essentially performed in a transform domain (wavelet transform), as opposed to the existing convention of manipulating images in the pixel domain. We address the key challenge of identifying the appropriate frequency band/component that can be embedded in the host images for fooling the networks while preserving the perturbation imperceptibility. This completely eradicates the need of computing network gradients for the perturbation estimation. Interestingly, we find that a single secret image can be used to fool multiple networks trained on large-scale datasets with high probability. Nevertheless, we analyze multiple secret images for thoroughness. We demonstrate successful fooling of the state-of-the-art ImageNet models, that includes ResNet-50 [3], Inception-V3 [14], VGG-16 [15] and MobileNet-V2 [16].

* Corresponding author.
E-mail address: saladin@cae.nust.edu.pk (S. Ud Din).

https://doi.org/10.1016/j.patrec.2020.04.025
0167-8655/© 2020 Elsevier B.V. All rights reserved.

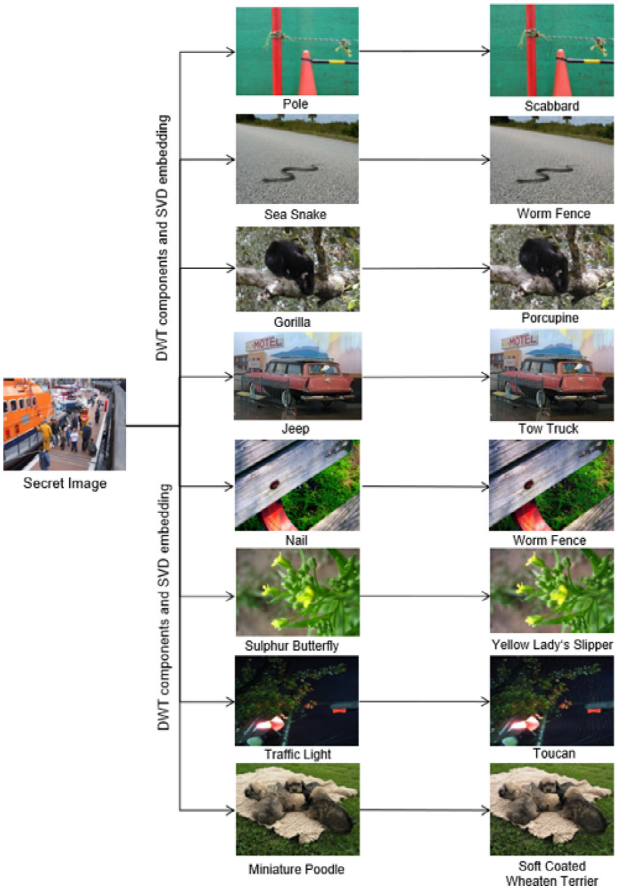

Fig. 1. Adversarial examples created with Discrete Wavelet Transform and Singular Value Decomposition based Steganography that successfully fool the state-of-the-art DNNs. A secret image (left) is embedded in the host images (middle) in the frequency domain to get adversarial examples (right). Labels predicted by ResNet-50 are also indicated.

2. Related work

Adversarial attacks on DNNs provide an opportunity to estimate a network's robustness in adversarial settings before its deployment in the real-world. They have recently attracted significant attention of the research community [9]. Szegedy et al. [11] first exposed the susceptibility of the state-of-the-art DNNs to small perturbations to input images that can lead them to incorrect predictions. Since then, several techniques have emerged to generate adversarial perturbations. Goodfellow et al. [17] presented the Fast Gradient Sign Method (FGSM) to efficiently generate adversarial perturbations for a given image using one step gradient ascend. Instead of one large step, Kurakin et al. [18] proposed to take multiple small steps in an iterative method to compute the gradients until the perturbation achieves the required fooling.

Dong et al.[19] presented an iterative method for computing adversarial perturbations based on momentum that avoids local maxima during the iterative process. They showed that the developed algorithm can also deceive adversarially trained networks. Successful targeted attack using an adversarial patch has been demonstrated by Brown et al. [20] that can deceive a network on a large number of images. In their technique, adversarial patches are crafted under Expectation Over Transformation framework. The patches are then added/replaced at a specific location in the natural image to make it an adversarial example. The adversarial perturbations generated by Wagner [21] are designed to attack the defensive distillation algorithm of Papernot et al. [22]. The generated perturbations are forced to be quasi-imperceptible by restricting their l_0, l_2 or l_∞ norm.

Papernot et al.[10] generated adversarial examples by putting an upper bound on the l_0-norm of the perturbation rather than on l_2 or l_∞ norm. Their method computes the adversarial saliency map for the network gradients after modifying the pixels of the input image one at a time. The algorithm alters the values of only those pixels that have greater affect on fooling Su et al. [12]. modified only one pixel of the image to create adversarial example. Their method uses Differential Evolution [23] to estimate the spatial location and RGB values of the adversarial pixel in the image. Another iterative method of computing adversarial perturbations is DeepFool algorithm proposed by Moosavi-Dezfooli et al. [24]. Their method is based on linearizing the decision boundaries of the deep neural network around the input image. A small perturbation vector is added to the image at every iteration that takes the perturbed image towards the decision boundary. The perturbation signal at every iteration is accumulated to compute the final perturbation.

All the above mentioned methods can fool deep models on individual images for which the perturbation is computed. Moosavi-Dezfooli et al. [13] proposed universal adversarial perturbations to fool neural networks on any image. These perturbations are generated by restricting their l_2 or l_∞ norm. Although impressive fooling ratios are achieved by their perturbations, the resulting adversarial patterns become quasi-imperceptible in the images. Moreover, a large amount of training data is required to compute their perturbations. Universal perturbations are also constructed by Khrulkov and Oseledets [25] using smaller number of images. They obtained the perturbations by taking singular values of the hidden layers' Jacobian matrices.Mopuri et al. [26] computed data independent adversarial perturbations using fast-feature-fool method.

The universal adversarial perturbations discussed above require training data for computing the manipulative signal for network fooling. The iterative nature of the techniques to compute those perturbations also makes them computationally expensive. For instance, for the perturbations based on network gradient estimates, e.g. [13], GPU based hardware is required for computing a single perturbation in a reasonable time. In contrast to these techniques, the method proposed in this work does not any require training data. Furthermore, no information regarding the targeted network is required. This also makes our adversarial perturbation an ideal choice for black-box attacks. Moreover, requiring no gradient computations makes our technique computationally efficient.

3. Problem formulation

We use the following notations for the formal presentation of the problem. Let $x \in \mathbb{R}^{m\times n}$ be a clean image that is given as an input to a well-trained DNN classifier $f(.)$. We ignore the number of image channels for notational simplification. The classifier maps the image to its correct label 'ℓ' with a high probability i.e. $P(\{f(x) : x \to \ell\}) \to 1$. Our goal is to generate a perturbation signal 'η' which when embedded in the clean input image satisfies the following constraint:

$$P(f(x+\eta) \neq f(x)) \geq \psi, \quad (1)$$

where ψ is referred to as the fooling ratio, defined as:

$$\psi = \frac{|\{f(x_i+\eta) \neq f(x_i)\}|}{M}, \quad \forall i \in \{1, 2, \dots, M\} \quad (2)$$

where M is the total number of samples in our dataset. Notice that, whereas 'η' is shown to be directly added to the images in the above formulation, it is not necessary to treat it as an additive signal during its estimation. This is one of the key differences in our treatment of 'η' and its more common handling in the existing methods that restrict themselves to the pixel domain to treat 'η' as an additive noise.

The existing adversarial attacks are directly aimed at pixel manipulation. Although we represent 'η' in Eq. (1) following the common convention, we actually compute the perturbation in a transform domain, which results in a manipulated image that is obtained with the inverse transform. This will be clarified in the next Section. Another major difference between our computation of 'η' and its conventional treatment in the literature is in terms of restricting its norm to control the perturbation perceptibility. The use of transform domain allows us to manipulate the images without paying particular attention to the pixel domain norm restrictions. As will be seen shortly, the smooth holistic manipulations resulting from the proposed technique intrinsically result in imperceptible patterns embedded in the adversarial images. Following the common convention, we alternatively refer to the manipulated images as the 'adversarial examples' in this paper.

4. Proposed approach

To create an adversarial example out of a natural image, we tap into the advances of Discrete Wavelet Transform (DWT) and Singular Value Decomposition (SVD) based steganography [27], [28]. The key intuition is that DWT has unique characteristics in terms of identifying the frequency regions where the external information can be hidden effectively in an image without being easily perceivable [27]. The main concept of our technique is that, we hide the low frequency components of a secret image inside the low frequency components of the host image, and also manipulate the low-to-high frequency components of the resulting image with the affine transformations of the secret image. The choice of hiding the low frequency components of the secret image and its manipulated low-to-high frequency components is based on the intrinsic properties of these components. In this work, we sometimes refer to the manipulated image as the stego-image to better contextualize our method in the area of steganography. The stego-image/adversarial example is finally used to deceive the classifier.

4.1. No training data and target model required

The popular techniques in the literature to generate universal adversarial perturbation (e.g. [13,29]) require training images that are used to optimize the perturbation signals, generally with respect to a target model to be attacked. This makes the techniques both time consuming as well as unattractive, because the target model can often be unknown in practice. The technique proposed in this work is agnostic to the target models, and it also does not require any particular training data. All that is required is a suitable secret image that can be embedded in any image to make the latter an adversarial example. This is a highly desirable property for a universal adversarial attack. It is noted that we make a careful selection of the secret image. Further details on the selection of the secret image are provided in Section 5.1.

4.2. 2D Discrete wavelet transform decomposition

We make use of 2-Dimensional 'Haar' based DWT [30,31] to decompose an image into its four components, denoted by LL, LH, HL, HH. The LL-component is obtained by Low-pass filtering the image in the horizontal direction and the vertical dimension, leading to a feature map that contains the low frequency information of the source image. The LH-component performs the Low-pass filtering in the horizontal direction and High-pass filtering in the vertical direction. This order reverses in the HL-component, while the HH-component uses High-pass filtering in both directions. Each of the resulting feature maps are components of the original image that have their own unique properties. We illustrate the 2D Haar DWT components of an example image in Fig 2.

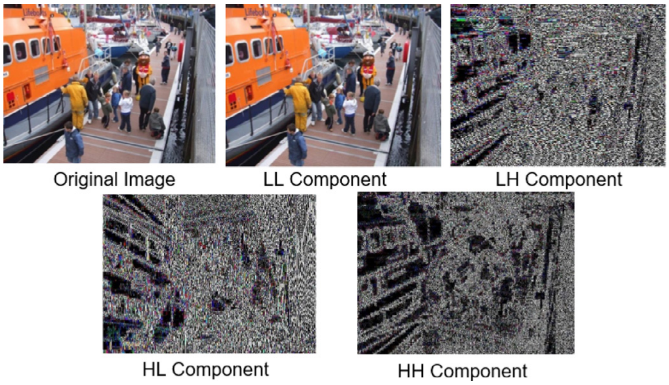

Fig. 2. Illustration of different components of 2D Wavelet Transform of the image. The LL and HH components respectively contain the low and high frequency information while HL and LH components contain both the high and low frequency information.

4.3. Singular value decomposition (SVD)

We use the Singular Value Decomposition (SVD) to merge the LL-components of the host and secret images, as explained below. An LL-component contains most of the image information in the low frequency bands where slight manipulation does not cause drastic changes in the perception. Using the SVD we can decompose an image component, say $\hat{x} \in \mathbb{R}^{m\times n}$ a follows:

$$\hat{x} = U * S * V^{\intercal}, \tag{3}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, and $S \in \mathbb{R}^{m\times n}$ is the diagonal matrix of singular values. In this work, we apply SVD to each channel of an RGB image individually for further processing.

4.4. Constructing the adversarial examples

Let x_1 be the host image and x_2 be the secret image. The aforementioned DWT decomposition is applied to both x_1 and x_2 to get their DWT components. First, we are interested in x_{1LL} - the LL-component of x_1 and x_{2LL} - the LL-component of x_2. We apply SVD to these components and fuse their singular values as follows:

$$S = (1-\alpha)S_{x_{1LL}} + \alpha S_{x_{2LL}}, \tag{4}$$

where S_{comp} denotes the singular value matrix (see Eq. 3) of the DWT component in the subscript, and α is the hyper-parameter of our technique. We note that the above fusion of singular values is performed individually for all the channels in our images.

We reconstruct the LL-component of the adversarial image, say y, using the fused singular values in Eq. (4) along the U and V matrices of x_1. Hence, we refer to this component of the adversarial example as y_{1LL} in the text to follow. Notice that, the dominant low frequency features in the adversarial image are mainly influenced by the original (i.e. host) image. On the other extreme, we borrow the HH-component of the adversarial image directly from the secret image, i.e. x_{2HH}. Human perception often finds the HH-component of DWT close to white noise, see the DWT HH Fig. 4. Replacing this component of an image with that of another image does not drastically change the Human perception of the original image. However, the high frequency variations in the quantized RGB values significantly distort the image for a DNN.

In order to strengthen the original perception of the host image, we also use the LH-component of the host image as

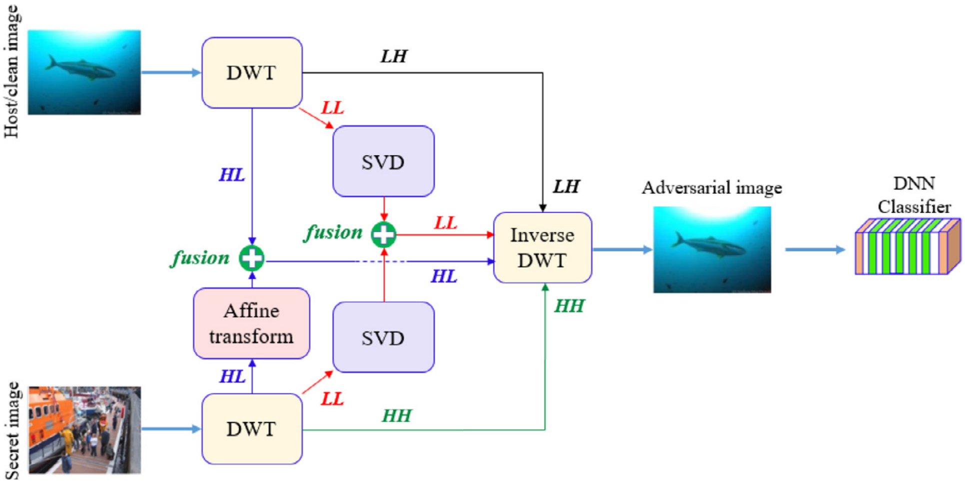

Fig. 3. Image agnostic perturbations are computed using DWT and SVD based steganography. Host and secret images undergo 2D haar wavelet transform to get four frequency components (LL, LH, HL, HH). Each channel of the LL-component of both images is decomposed into singular values which are then fused to form the LL-component of the desired adversarial example. The LH and HH components of the desired image are borrowed from the host and secret image, respectively. The HL-component is also a fusion of the respective components of the two original images. Inverse transform is computed over these four components to form and adversarial example with imperceptible perturbations.

the LH-component of the adversarial image. However, we introduce slight modifications to the HL-component of the host image as follows. We take x_{2HL}, and rotate it at four different angles, i.e. $-5^o, 5^o, -10^o, 10^o$. The rotated components are averaged to form $\tilde{x}_{2HL}$ and then fused with x_{1HL} under a convex combination as follows:

$$y_{2HL} = (1-\beta) * x_{1HL} + \beta * \tilde{x}_{2HL}, \tag{5}$$

where β is a hyper-parameter that controls the contribution of each component in the fusion process. Notice that we write L.H.S. of Eq. (5) as y_{2HL} because of the fusion. It is worth mentioning that whereas we systematically choose the rotation angles for the affine transformations, random rotations in $[-10^o, 10^o]$ also work equally well. The main intuition behind performing the affine transformations to x_{2HL} is to confuse a DNN with those transformations that can naturally occur in the images. Other affine transformations can also be explored for the same purpose.

Finally, to reconstruct the adversarial/stego image, we use the inverse DWT. The inverse DWT is executed over the components $\{y_{1LL}, x_{1LH}, y_{2HL}$ and $x_{2HH}\}$. The overall process of generating the adversarial image is illustrated graphically in Fig. 3. The resulting adversarial examples have embedded perturbations that are generally very hard to perceive by humans. However, state-of-the-art CNNs get confused in classifying those images, despite showing high confidence on the correct predictions of the host/original images.

5. Experiments and results

To validate our stagnographic universal adversarial attack, we perform extensive experiments with ImageNet 2012 [32]. We select four popular DNNs, i.e. ResNet-50 [3], Inception-V3 [14], VGG-16 [15] and MobileNet-V2 [16] and fool those on the ImageNet validation set. Our choice of the networks is based on their versatility and established performance. Recall that our technique performs non-targeted black-box attacks. As such, no training data is required for our attack, except the secret image. This allows us to use all the images in the validation set of the ImageNet as our test data (except the secret image). This is in contrast to the existing methods for computing the universal adversarial perturbations, e.g. [13], that consume a large number of images from the dataset for training, and report result only on a smaller subset of the ImageNet validation set.

5.1. Secret image selection

Our technique embeds a secret image in the host image to create adversarial examples. This gives us full freedom to use any image as the secret image. However, it was observed in our experiments that a secret image with more edges is preferable under our scheme. To identify more desirable secret images, we can take advantage of conventional filtering techniques. In this work, we use Sobel filter to extract edge information from the images to decide their suitability as the secret image. The used filter operator is given in Eq. 6.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \otimes x_2 \quad \text{and} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \otimes x_2, \tag{6}$$

where $\otimes$ denotes the convolution, and G_x and G_y are the feature maps preserving the vertical and horizontal edge information.

In order to choose the secret image we use an empirical threshold on the Edge Pixels (EP) identified by the filter. The EP value counts the number of pixels in the edges extracted by the filter. We illustrate the five secret images used in our experiments in Fig. 4. The top row of the figure shows the secret images with their EP values. The bottom row shows the resulting adversarial examples. We have intentionally chosen the original/clean image with a relatively plain background to clearly illustrate the perceptibility of the perturbations. As can be seen, based on the edge information in the secret image, the EP values decrease from Fig. 4(a) to (e). This also results in reducing the perturbation perceptibility in the adversarial examples. However, secret images with smaller EP values also result in the adversarial examples that have less fooling ratios.

In Table 1, we summarize the fooling rates resulting on ImageNet validation set using the five secret images shown in Fig. 4. We emphasize, that the results are on 49,999 ImageNet samples. It can be seen that with the images having EP values around 15K (Fig. 4), the fooling ratios for the state-of-the-art ImageNet models is significant, reaching up to 89% fooling for the MobileNet. These

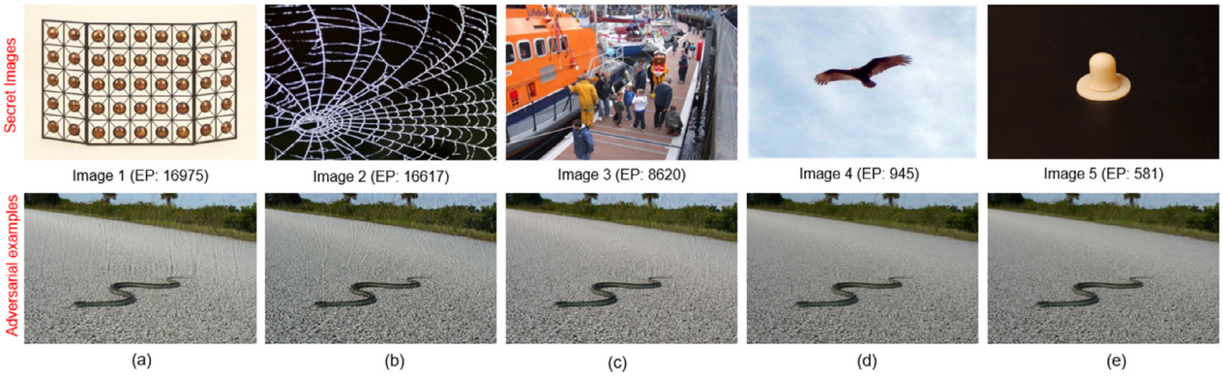

Fig. 4. Adversarial examples formed by different secret images. The top row shows the secret images along their Edges Pixel (EP) values. The bottom row presents the adversarial examples generated using the corresponding secret images. With the larger number of edges in a secret image, the fooling rate increases (see Table 1), however it also makes the perturbation quasi-imperceptible.

Table 1
Fooling rates (%) using the five secret images illustrated in Fig. 4. The rates are computed for 49,999 images of ImageNet validation set. We use hyper-parameter values $\alpha = 0.1,\ \beta = 1$.

	Fooling Rate (%)			
Different Secret images	**ResNet-50**	**VGG-16**	**Inception V3**	**MobileNet V2**
Image 1	84.77	87.19	79.19	89.13
Image 2	82.11	84.96	74.51	86.61
Image 3	71.76	73.88	64.59	77.52
Image 4	42.72	38.65	41.25	48.26
Image 5	42.61	38.45	41.78	48.74

Table 2
Ablation analysis with different steganographic modifications applied to the clean image. Image 3 in Fig. 4 is used as the secret image.

	Fooling Rate (%)			
Modification	**ResNet-50**	**VGG-16**	**Inception V3**	**MobileNet V2**
Image embedding	29.94	23.69	28.91	34.51
HH component	30.96	23.7	30.19	36.91
HL component	60.8	61.73	54.25	66.74
All Hybrid	71.76	73.88	64.59	77.51

results are especially intriguing because the fooling is achieved using a non-optimization based method. Moreover, the attack is conducted in a true black-box setup, where we have not assumed any information about the target model. A single secret image is able to form adversarial examples that generalize well across the networks having varied architectures.

5.2. Discussion

As an ablation study of the overall technique, we modify the method of embedding the secret image in the clean image and analyse the results. As the first instance, we simply embed the Image 3 (as shown in Fig. 4) using steganography. The fooling ratios for the four networks are reported against the *Image embedding* modification in Table 2. In the second experiment, we additionally modify the HH-component of the DWT using the procedure discussed in Section 4. The result of these experiments are reported against *HH component* modification in the Table. Similarly, the last two rows of the Table report the fooling ratios when additional HL-component modification is performed, and when all the proposed modifications (in Section 4) are performed. It is clear from the Table that each of our proposed modification to the individual component of DWT adds to the eventual fooling ratio achieved by our technique. Note that, we deliberately use Image 3 instead of Image 2 (Fig. 4) in this Table to emphasize that this trend remains generic for different secret images used under the proposed technique.

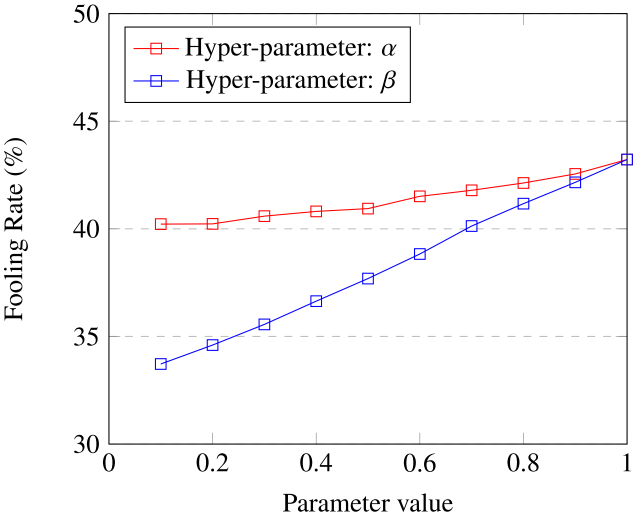

Fig. 5. ResNet-50 fooling rate (%) for secret image 4 using different values of α and β. The x-axis values for α are in the range [0, 0.1]. 10x scaling is used for better readability.

Recall that the proposed approach requires two hyper-parameters that govern the fooling ratio and perceptibility of the perturbations in the adversarial examples. The first hyper-parameter is α that decides the strength with which the secret image is embedded in the host image while the other hyper-parameter is β which determines the strength of embedding the vertical edges in the host image. The fooling ratios of ResNet-50 network with different values of α and β are summarized in Fig. 5. As the value of α increases from 0.01 to 0.1, the strength of embedding the secret image in the host images increases and hence the fooling ratio increases. Varying β values from 0.1 to 1 in Eq. (5), multiple y_{HL} components are generated. These HL components are then used to generate adversarial examples, and thier fooling prowess is analyzed. Fig. 5 clearly shows that as the value of β increases, the fooling ratio also increases. With an increase in the values of α and β, fooling ratio increases until a point (α = 0.1 and β = 1) where we get maximum fooling ratio with imperceptibility. On further increasing these values, the fooling ratio increases further, albeit slightly. Nevertheless, it also makes the perturbations perceptible for the Human visual system. The plot is shown for Image 4 in Fig. 4.

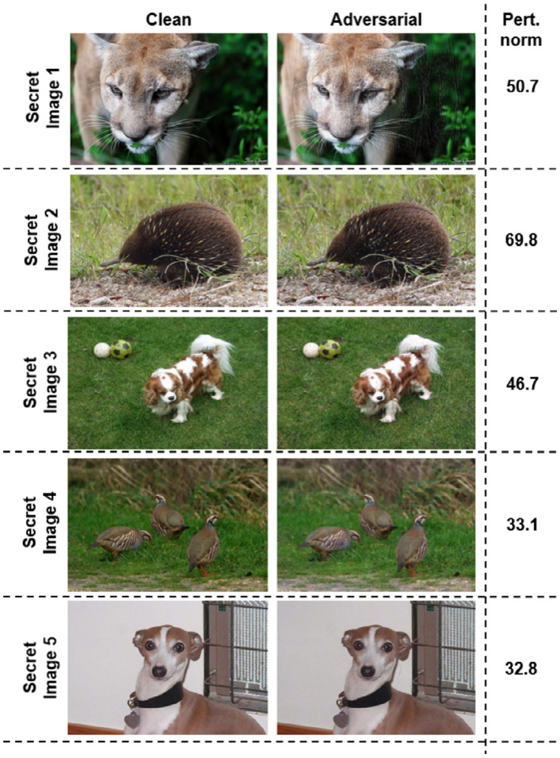

Fig. 6. Random adversarial examples with perturbation norms (ℓ_2) close to the mean perturbation norms reported in Table 3. The perturbations remain largely imperceptible to the Human visual system.

Table 3
ℓ_2-norm of the perturbations for different secret images.

Different Secret images	l_2-norm (Min Val)	l_2-norm (Max Val)	l_2-norm (Mean Val)
Image 1	11.49	103.96	48.85
Image 2	14.15	103.66	49.42
Image 3	8.56	98.33	34.94
Image 4	4.46	96.74	29.75
Image 5	3.5	95.71	28.13

5.3. Perturbation perceptibility

Adversarial examples are considered more effective when the underlying perturbations to the image remain imperceptible to humans. Since we do not modify the image in the pixel domain, our method naturally results in hard to perceive perturbations. Nevertheless, the resulting adversarial images do differ from the original image in terms of e.g. brightness, sharpness. To quantify the differences, we summarize the ℓ_2-norm of the difference between the original and adversarial images in Table 3. The reported values are for all five secret images shown in Fig. 4. The values are given for 8-bit images with range [0–255]. As can be seen, as the edges in the secret images increase, the difference between the original and adversarial images increases. It should be noted that we report the ℓ_2-norm of the difference following the existing conventions. Since our technique essentially modifies the image in the frequency domain, even significant ℓ_2-norm perturbations still remain largely imperceptible to humans under our technique. We provide example visualizations to corroborate this claim in Fig. 6, where random examples having perturbation norms close to the mean values of the norms in Table 3 are shown.

6. Conclusion

We developed an adversarial attack on deep learning inspired by steganography. The proposed attack embeds a secret image inside any host image to fool any network on the resulting adversarial example. This doubly-universal attack achieves high fooling rates (~ 80%) on a variety of state-of-the-art networks under tru balck-box settings. To perform the attack, we mixed the low frequency information of the secret and the host image, while replacing the high frequency information of the host image with that of the secret image. It is observed that secret images with larger number of edges are more suitable for the proposed attack. Depending upon the secret image, the resulting adversarial perturbations remain imperceptible to quasi-imperceptible, while maintaining good fooling rates across the networks. The proposed non-optimization attack is performed holistically on the images by computing their discrete wavelet transforms and singular value decompositions.

Declaration of Competing Interest

None

References

[1] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, et al., Deep neural networks for acoustic modeling in speech recognition, IEEE Signal Process. Mag. 29 (2012) 82–97.

[2] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.

[3] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[6] D. Shen, G. Wu, H.-I. Suk, Deep learning in medical image analysis, Annu. Rev. Biomed. Eng. 19 (2017) 221–248.

[7] M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics, J. Big. Data. 2 (1) (2015) 1.

[8] S. Ramos, S. Gehrig, P. Pinggera, U. Franke, C. Rother, Detecting unexpected obstacles for self-driving cars: Fusing deep learning and geometric modeling, in: Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), IEEE, Los Angeles, CA, 2017, pp. 1025–1032.

[9] N. Akhtar, A. Mian, Threat of adversarial attacks on deep learning in computer vision: a survey, IEEE Access 6 (2018) 14410–14430.

[10] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: Proceedings of the 2016 IEEE European Symposium on Security and Privacy, IEEE, Saarbrucken, Germany, 2016, pp. 372–387.

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2013. arXiv preprint arXiv:1312.6199.

[12] J. Su, D.V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Comput. 23 (5) (2019) 828–841.

[13] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1765–1773.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014. arXiv preprint arXiv:1409.1556.

[16] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.

[17] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2014. arXiv preprint arXiv:1412.6572.

[18] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, 2016. arXiv preprint arXiv:1607.02533.

[19] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9185–9193.

[20] T.B. Brown, D. Mané, A. Roy, M. Abadi, J. Gilmer, Adversarial patch, 2017. arXiv preprint arXiv:1712.09665.

[21] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), IEEE, San Jose, CA, 2017, pp. 39–57.

[22] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in: 2016 IEEE Symposium on Security and Privacy (SP), IEEE, San Jose, CA, 2016, pp. 582–597.

[23] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.

[24] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2574–2582.

[25] V. Khrulkov, I. Oseledets, Art of singular vectors and universal adversarial perturbations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8562–8570.

[26] K.R. Mopuri, U. Garg, R.V. Babu, Fast feature fool: A data independent approach to universal adversarial perturbations, 2017. arXiv preprint arXiv:1707.05572.

[27] P.-Y. Chen, H.-J. Lin, et al., A DWT based approach for image steganography, International Journal of Applied Science and Engineering 4 (3) (2006) 275–290.

[28] C.-C. Lai, C.-C. Tsai, Digital image watermarking using discrete wavelet transform and singular value decomposition, IEEE Trans. Instrum. Meas. 59 (11) (2010) 3060–3063.

[29] O. Poursaeed, I. Katsman, B. Gao, S. Belongie, Generative adversarial perturbations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4422–4431.

[30] S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Transactions on Pattern Analysis & Machine Intelligence (7) (1989) 674–693.

[31] I. Daubechies, Ten lectures on wavelets, 61, Siam, 1992.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.