2019 19th International Symposium on Communications and Information Technologies (ISCIT)

Simultaneous convolutional neural network for highly efficient image steganography

Toan Pham Van, Thoi Hoang Dinh
R&D Lab, Sun* Inc
pham.van.toan@sun-asterisk.com, hoang.dinh.thoi@sun-asterisk.com

Ta Minh Thanh
Le Quy Don Technical University, 236 Hoang Quoc Viet, Bac Tu Liem, Ha Noi
thanhtm@mta.edu.vn

***Abstract*—Over the past decade, steganography has attracted a large number of researchers' attention because of its practical value to information security. Unlike cryptography which aims to protect the content of the message, steganography involves hiding the message within a transport layer. Most traditional methods directly hide information in one or two least significant bits. Recently, miscellaneous effort with deep learning approach has been devoted to hiding more information without losing the integrity of the transportation layer. In this paper, our work focuses on image steganography which means hiding an image (secret image) inside another image of the same size (cover image). A deep convolutional neural network with similar architecture to U-Net is employed as the hiding network, and a new training scheme is proposed to speed up the training phase. Through extensive experiments, it has been verified that the new network architecture, combined with the new training strategy, can result in lower mean square error of pixel difference whereas the training time is reduced by half.**

***Index Terms*—Image steganography, information security, deep learning, convolutional neural network**

I. Introduction

In any types of communication, preserving the the security and integrity of information is the utmost important task. There are two widely-used security mechanisms including cryptography and steganography. While cryptography focuses on protecting the content of the secret message, steganography is mainly about hiding the presence of the secret message itself. Steganography is the practice of concealing information within other public text, image, or speech data. Previously, steganography was primarily used in military-related applications. These days, steganography is widely applied in many fields of technologies. The advancement of hiding information technique, in fact, has brought great benefit not only in information security but also many other fields. Because of the great potential of steganography, more companies, businesses, and organizations have been researching and applying this technique to serve their interests.

In this paper, we mainly discuss the topic of image steganography which is a type of steganography focusing on hiding a ***secret image*** within another image called ***cover image***. One of the most common approaches to this problem is to hide the secret image directly in Least Significant Bits (LSB)[12] of the cover image. The effect of this method depends on the number of least significant bits used to hide the secret image. When the number of bits used to represent the secret image is smaller than the number of bits required to represent the container image, the accuracy of LSB is imperceptible. There are other methods which are based on pixel level strategy or statistical approach such as Bit-Plane Complexity Segmentation [14], but these methods are easy to be detected.

Recently, there are some researches which apply deep learning technique for image steganography. The common way is to build a pairs of deep convolutional neural networks. The first network is for hiding the secret image inside the cover image to produce the ***container image***. The second network takes the container image as input to reconstruct the original secret image or the ***revealed image***. Through error backpropagation, parameters of both network are updated to minimize the pixel-wise difference of pairs of cover - container image and secret - revealed image. Our approach is quite similar to the common method. The well-known U-net [22] with some minor modifications is employed as the hiding network, and another convolutional network is used as the revealing network. The original U-net is mainly applied to biomedical image segmentation; however, we find that the architecture of U-net is suitable for hiding network. The overall architecture of our network is demonstrated as in **Figure 1**. Besides, a new training scheme is also proposed. Instead of using just one optimizer as previous work, two separate optimizers are used in our experiment, one for each network. With the new scheme, we can easily alter the focus on minimizing the difference between pair of cover - container image or pair of secret - revealed image.

In summary, there are three main contributions in this paper. First, we explore the use of deep convolutional neural networks to address the problem of image steganography. Second, a new training strategy with two separate optimizers is proposed for our model with the separation in the optimization phase of hiding and revealing processes. Finally, we propose a simultaneous deep convolutional neural network which contains a pair of hiding and revealing network. All the mentioned factors help our approach both achieve the lower pixel wise difference and cut the training time by half when compared with other

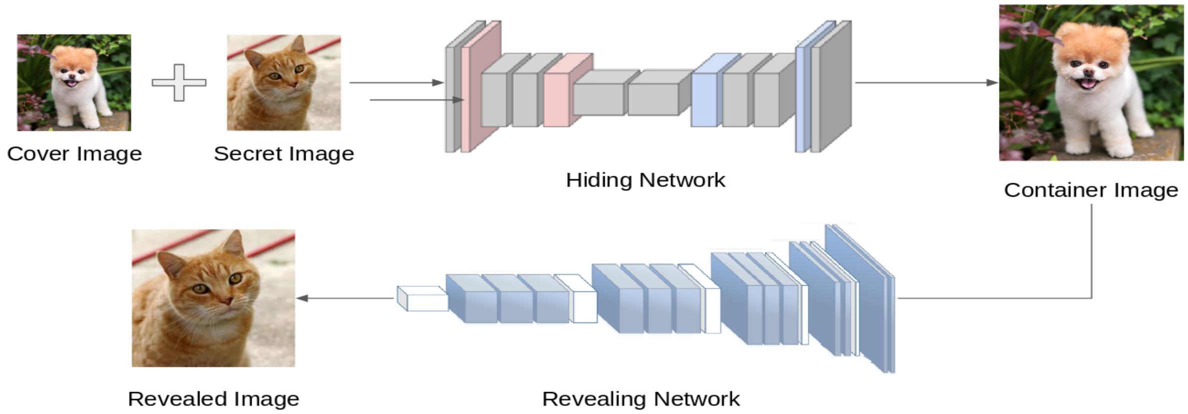

Fig. 1. The architecture of our image steganography system. The two components of the full system. Left: Hiding the secret image in the cover image with Unet. Right: Revival the hidden image with the revival CNN network.

deep learning based method.

The rest of this paper is organized as follows. Section 2 presents a brief review of related work. The system setup for the experiment and dataset is mentioned in Section 3. In Section 4, the simultaneous convolutional neural network and the separate optimization with two learning rate scheduler will be thoroughly discussed. Experimental result and evaluation are presented in Section 5. Finally, we conclude the paper in Section 6.

II. Related work

A. Steganography techniques

Steganography is a field specializing in research methods, techniques, and algorithms to embed secret information into a transportation layer without making any suspicion. Steganography can be divided into two main categories: ancient and digital steganography [8] [18]. Secret information and transportation layer can be of any media type such as text, image, video, and audio. In literature, a large number of steganography systems have been built with different approach. It includes many different methods, algorithms, and techniques as same as cryptography. Almost all digital file formats include text, images, audio, and video can be used for steganography. However, the formats containing redundant bits are more suitable for steganography. Redundant bits can be defined as the bits that, when not in use, do not affect the display of information [4]. Image and audio are two types of data that satisfy these conditions.

B. Traditional methods in image steganography

The image steganography is the most popular technique for cover content. There are many embedding methods can be implemented on image cover such as spatial domain based and transform domain based steganography.

In the spatial domain based steganography, the secret information is hidden directly into the values of pixels from the cover image. The hidden information is not noticeable on the image with high quality. Algorithms of such kind of methods are simple, fast, and efficient. Some techniques can be mentioned that related to such as pixel value differencing [25], palette based method [3], gray level value based method [27], and so on. However, the disadvantage of spatial domain based methods are not robust against the image processing attacks, geometric attacks, compression attacks.

In order to improve the disadvantage of spatial based methods, the transform domain based steganography is used to embed the secret information into the area of the cover image that are less uncover to the image attacks. Discrete cosine transformation technique (DCT) [1] and Discrete Wavelet transformation technique (DWT) [10] are two methods that are mostly used as a transformation domain technique. This kind of methods has more usable for real applications.

C. Convolutional Neural Network

Convolutional Neural Network [17] (CNN) is a variant of a feed-forward neural network. Since its appearance when AlexNet [2] won the championship of ImageNet competition [24], it has quickly become a trending technology in deep learning. This method proved particularly effective in the computer vision field. Several popular tasks has successfully been established like object detection [21], image classification [17], image segmentation [23], etc. The efficient of CNN is its ability to find hidden structures inside the data via optimizing the process of the neural network. The traditional methods were difficult to do this or must require experiences of experts. In addition to the advantages of the CNN itself, our work also benefits a lot from some factors such as the GPU hardware, the modern optimizer as Adam [16] and especially the abundance of data. Also, selecting the appropriate network architecture, tuning the model parameters, and pre-processing data are some important factors in our experiments.

D. Deep learning for image steganography

Recently there are some researches on applying deep learning techniques for image steganography [28] [26] [5]. Unlike traditional methods, neural network models will use the entire pixel of the image to make the message invisible. The selection of model parameters depends on the optimization process. So these are like a black box for the end users. That means the attacking to find the rule to hide information will become much more difficult. Baluja [5] is the first to propose using neural networks to hide image inside another image. However, using three networks makes training and encoding time nearly twice as much as that of our architecture. Another method is Stegnet [28], but with their network architecture, there are still some noise at non-texture areas in container image. This is also the motivation for us to propose a new neural network model which aims to mitigate these disadvantages in above methods.

III. Proposed method

A. Model architecture

As mentioned in the previous section, we build our model using two neural networks with different architectures. These networks are responsible to encode the secret image into the container image and decode the container image to get the original secret image. The whole processing pipeline is shown in Figure 1. This structure is similar to the architecture that appear in auto encoder [20], generative adversarial networks (GAN) [11], etc. However, the structure of the two networks in the encoding and decoding phase is different. This architecture promotes the advantages of both two sub-network, thereby helping to improve the accuracy of the entire model.

In our proposed method, the use of two different architectures leads to difficulty in the optimization process because each network has effect to the general loss function as in other method such as Stegnet [28], and Baluja [5]. To solve this problem, we propose two different optimization strategies for each network. These two optimization processes are independent, making optimization simpler and easier to reach the extrema. We implement this idea with the ReduceLROnPlateau function in the Pytorch framework to reduce learning rate when a metric has stopped being improved. This scheduler monitors the metric and if no improvement is seen after some epochs, the learning rate is, then, reduced. Specifically, we use Adam [16] optimizer with two different settings for each network.

B. Unet-based hiding network

In this paper, we only use two neural networks namely **H-net** for hiding image and **R-net** for retrieving original image. For the hiding network, we use a convolutional neural network (CNN) architecture based on the U-net [22]. The cover image and secret image are concatenated as the input of this network. It is presented by a 6 channels image in our implementation. U-Net is a popular architecture used in semantic segmentation problems [19]. The H-net consists of two approaches: an encoder and decoder approach. The encode approach has various convolution layers. Depend on that, the decoder approach also has various deconvolution layers. When the features are reduced in dimensions, it is upsampled again to the image size by deconvolution. Deconvolution learns

Layer (type)	Output Shape
Conv2d-1	[-1, 50, 112, 112]
LeakyReLU-2	[-1, 50, 112, 112]
Conv2d-3	[-1, 100, 56, 56]
BatchNorm2d-4	[-1, 100, 56, 56]
LeakyReLU-5	[-1, 100, 56, 56]
Conv2d-6	[-1, 200, 28, 28]
BatchNorm2d-7	[-1, 200, 28, 28]
LeakyReLU-8	[-1, 200, 28, 28]
Conv2d-9	[-1, 400, 14, 14]
BatchNorm2d-10	[-1, 400, 14, 14]
LeakyReLU-11	[-1, 400, 14, 14]
Conv2d-12	[-1, 400, 7, 7]
ReLU-13	[-1, 400, 7, 7]
ConvTranspose2d-14	[-1, 400, 14, 14]
BatchNorm2d-15	[-1, 400, 14, 14]
SkipConnection-16	[-1, 800, 14, 14]
ReLU-17	[-1, 800, 14, 14]
ConvTranspose2d-18	[-1, 200, 28, 28]
BatchNorm2d-19	[-1, 200, 28, 28]
SkipConnection-20	[-1, 400, 28, 28]
ReLU-21	[-1, 400, 28, 28]
ConvTranspose2d-22	[-1, 100, 56, 56]
BatchNorm2d-23	[-1, 100, 56, 56]
SkipConnection-24	[-1, 200, 56, 56]
ReLU-25	[-1, 200, 56, 56]
ConvTranspose2d-26	[-1, 50, 112, 112]
BatchNorm2d-27	[-1, 50, 112, 112]
SkipConnection-28	[-1, 100, 112, 112]
ReLU-29	[-1, 100, 112, 112]
ConvTranspose2d-30	[-1, 3, 224, 224]
Sigmoid-31	[-1, 3, 224, 224]
SkipConnection-32	[-1, 3, 224, 224]

Fig. 2. The detail of H-net architecture with 224×224 input size

Layer (type)	Output Shape
Conv2d-1	[-1, 64, 224, 224]
BatchNorm2d-2	[-1, 64, 224, 224]
ReLU-3	[-1, 64, 224, 224]
Conv2d-4	[-1, 128, 224, 224]
BatchNorm2d-5	[-1, 128, 224, 224]
ReLU-6	[-1, 128, 224, 224]
Conv2d-7	[-1, 256, 224, 224]
BatchNorm2d-8	[-1, 256, 224, 224]
ReLU-9	[-1, 256, 224, 224]
Conv2d-10	[-1, 128, 224, 224]
BatchNorm2d-11	[-1, 128, 224, 224]
ReLU-12	[-1, 128, 224, 224]
Conv2d-13	[-1, 64, 224, 224]
BatchNorm2d-14	[-1, 64, 224, 224]
ReLU-15	[-1, 64, 224, 224]
Conv2d-16	[-1, 3, 224, 224]
Sigmoid-17	[-1, 3, 224, 224]

Fig. 3. The detail of R-net architecture with 224×224 input size

the parameters for upsampling. For the encode phase in H-net, we used five continuous convolution layers with batch normalization [13] and Leaky-ReLU activation [29]. In the decode phase, we add the skip connection layer and the ReLU activation after each deconvolution layer. Assume that the input size $224 \times 224 \times 3$ of both cover and secret image, the detail of H-net is describe in **Figure 2**.

C. CNN decoding network

After the pair of cover and secret images processed via H-net, the output of this network is a container image. This image is used as the input of the R-net directly. The container image generated by H-net should be much similar to the cover image. However, it should be able to recover the secret image from the container image using R-net. In order to do that, we use 6 convolution layers with 3×3 kernel size. Each layer is followed by a Batch Normalization layer and ReLU activation except the last one. With output of H-net described in Figure 2, the parameters of R-net is shown in **Figure 3**.

D. Error measurement

As mentioned above, the main purpose of our method is to make two neural networks for hiding and recovering the secret image. The goal of H-net is to hide the secret image (S) in the cover image (C). Therefore, after hiding S, only the cover image is visible, called container image or hidden image (H). It can be passed to the R-net to get the secret image back to output image (O), called revealed image. To compare the similarity of the two images, we used two method that are structural similarity index (SSIM) [6] and mean square error (MSE) [7]. For the optimization process, a loss function is defined to optimize the model parameters. The neural networks are trained by reducing the following error function.

$$L_{total}(C,H,S,O) = L(C,H) + \lambda L(S,O), \tag{1}$$

where L can be MSE or SSIM as mentioned above. In fact, the experiment showed that the MSE loss function is better than SSIM. Therefore, MSE is used in the training phase.

IV. Dataset and System Setup

A. Dataset

In any machine learning and deep learning tasks, data plays an important role to the accuracy of the whole system. Thus, the data preparation phase must be carried out carefully. Specifically, the well-known ImageNet dataset [24] is used in the training phase, and the Holiday dataset [9] is used for validation. The ImageNet is one of the largest visual databases which contains up to 14 million images of different categories. The Holiday dataset is a set of 1491 images which are mainly holiday photos. In this experiment, only $15K$ randomly chosen images from the ImageNet dataset are used for training. Then, all of the images from the Holiday dataset are used for validation.

B. System setup

Our experiment is conducted on a computer with Intel Core i5-7500 CPU @3.4GHz, 32GB of RAM, GPU GeForce GTX 1080 Ti, and 1TB SSD Harddisk. Both the hiding and revealing network are implemented with the well-known PyTorch framework [15].

V. Experimental Results and Evaluation

A. Accuracy with MSE

To evaluate the accuracy of the model, we used two measurement method are MSE and Average Pixel-wise Difference (APD) of two image. The MSE is the cumulative squared error between two image. The mathematical formula for the MSE error is presented as follows.

$$\frac{1}{M*N}\sum_{y=1}^{M}\sum_{x=1}^{N}(I_{x,y} - I'_{x,y})^2, \tag{2}$$

where $I(x,y)$ is a pixel in original image and $I'(x,y)$ is the corresponding pixel in the output image of our model. M, N are the dimensions of the images.

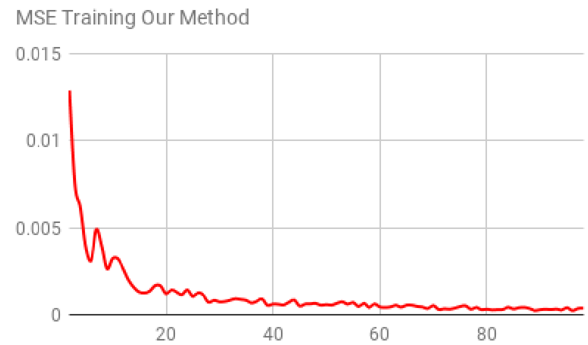

Fig. 4. Training loss of our method in 100 epochs

TABLE I
MSE error compare with other method in same dataset

λ	Our method	Baluja method [5]
	Imagenet Dataset	
0.5	**0.00024**	0.00244
0.75	**0.00035**	0.00423
1	**0.00043**	0.00829
	Holiday Dataset	
0.5	**0.01333**	0.06354
0.75	0.02135	0.09476
1	**0.02546**	0.10638

A lower value for MSE means lesser errors, meaning that the quality of the neural network model is higher. We perform the calculation of MSE errors directly when training of neural networks. The value of the loss function is evaluated on the train and the test dataset after each epoch. This error is calculated based on the formula given in (1). With $\lambda = 0.5$, the smallest loss value of our method in the training dataset is **0.00024** and that of value in testing dataset is **0.00042**. This result is better than other methods implemented on the same dataset. The detailed results are described in Table I. Our method achieved the best results in the Imagenet dataset. The value of loss function is superior to other methods. However, in Holiday dataset, the difference is trivial. This can be explained because the characteristics of two datasets are different.

B. Accuracy with averaged pixel-wise difference

In this subsection, we evaluate the accuracy of the proposed approach by another criterion: the averaged pixel-wise difference (APD). The source code used in our experiment are available at this repository[1] on Github

$$APD = \frac{\sum_{c=channel}^{RGB}\sum_{y=1}^{M}\sum_{x=1}^{N}(I_{x,y,c} - I'_{x,y,c})^2}{255\times 3}, \tag{3}$$

where $I_{x,y,c}$ is the value of pixel (x,y) in color channel c, same for $I'_{x,y,c}$

C. Training and predicting time

Our network architecture not only improves the accuracy of the steganography but also reduces the training and predicting

[1]https://github.com/nicolashahn/diffimg

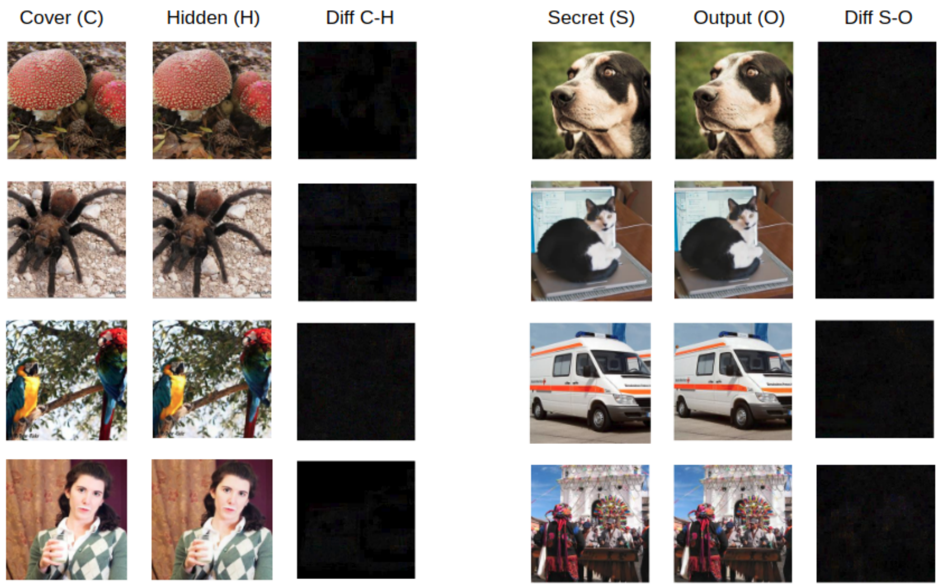

Fig. 5. Some result of our method with images in Imagenet dataset

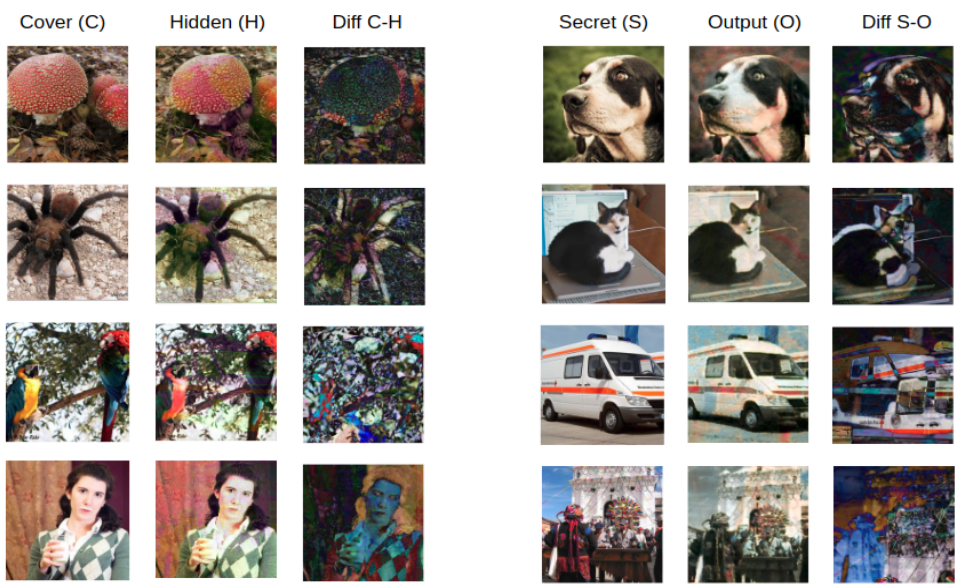

Fig. 6. Some result of Baluja [5] method with images in Imagenet dataset

TABLE II
COMPARISON OF AVERAGE PIXEL WISE IN DIFFERENCE DATASET

	Our method		Baluja method [5]	
	C - H	S - O	C - H	S - O
Training Imagenet	**0.0116**	**0.0123**	0.1434	0.1242
Training Holiday	**0.3123**	**0.4532**	1.2523	1.1822
Testing Imagenet	**0.0142**	**0.0168**	0.1952	0.2286
Testing Holiday	**0.3675**	**0.5675**	1.8620	2.1522

TABLE III
TRAINING TIME WITH 15000 IMAGES IN IMAGENET DATASET ON GPU GEFORCE GTX 1080 TI

Batch size	Our method (s)	Buluja method (s) [5]
16	**890**	1123
32	**564**	876
64	**391**	543

time since using only two neural networks. The Table III and Table IV show that the time of training and testing had been improved significantly. Especially testing time is about 1.8 times faster than Baluja's method [5].

TABLE IV
TESTING TIME WITH 1 IMAGE IN IMAGENET DATASET ON GPU GEFORCE GTX 1080 TI

Device type	A - Our method(ms)	B - Buluja method(ms) [5]	B / A
GPU	**48**	86	**1.79**
CPU	**934**	1656	**1.81**

VI. CONCLUSION

In this paper, we have proposed a deep learning based method for efficient image steganography. Two different deep convolutional neural networks are designed to work in pair for hiding the secret image inside the cover image and revealing the original secret image from the container image. Also, a new training scheme with some modification in error back-propagation is introduced to speed up the training phase. The difference between pair of cover - container images and pair of secret - revealed images is hardly noticeable by human's eyes. Through extensive experiments, it is verified that our proposed method can achieve lower mean square error of pixel difference than both traditional and recently proposed methods. In term of training time, the proposed training scheme can cut the training time by half whereas the accuracy is not sacrificed.

ACKNOWLEDGMENT

This work is partially supported by ***Sun-Asterisk Inc***. We would like to thank our colleagues at ***Sun-Asterisk Inc*** for their advice and expertise. Without their support, this experiment would not have been accomplished.

REFERENCES

[1] High-performance jpeg steganography using quantization index modulation in dct domain. *Pattern Recognition Letters*, 27(5):455 – 461, 2006.

[2] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Mahmudul Hasan, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.

[3] Dimple Anandpara and Amit Kothari. Article: Working and comparative analysis of various spatial based image steganography techniques. *International Journal of Computer Applications*, 113(12):8–12, March 2015. Full text available.

[4] Ross J Anderson and Fabien AP Petitcolas. On the limits of steganography. *IEEE Journal on selected areas in communications*, 16(4):474–481, 1998.

[5] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems*, pages 2069–2079, 2017.

[6] Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2012.

[7] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

[8] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. *Signal processing*, 90(3):727–752, 2010.

[9] Matthijs Douze, Adrien Gaidon, Herve Jegou, Marcin Marszałek, and Cordelia Schmid. Inria-lears video copy detection system. In *TREC Video Retrieval Evaluation (TRECVID Workshop)*, 2008.

[10] Emir Ganic and Ahmet M. Eskicioglu. Robust dwt-svd domain image watermarking: Embedding data in all frequencies. In *Proceedings of the 2004 Workshop on Multimedia and Security*, MM&Sec '04, pages 166–174. ACM, 2004.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[12] Shailender Gupta, Ankur Goyal, and Bharat Bhushan. Information hiding using least significant bit steganography and cryptography. *International Journal of Modern Education and Computer Science*, 4(6):27, 2012.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[14] Eiji Kawaguchi and Richard O Eason. Principles and applications of bpcs steganography. In *Multimedia Systems and Applications*, volume 3528, pages 464–474. International Society for Optics and Photonics, 1999.

[15] Nikhil Ketkar. Introduction to pytorch. In *Deep learning with python*, pages 195–208. Springer, 2017.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[18] Bin Li, Junhui He, Jiwu Huang, and Yun Qing Shi. A survey on image steganography and steganalysis. *Journal of Information Hiding and Multimedia Signal Processing*, 2(2):142–172, 2011.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[20] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[25] V. Sabeti, S. Samavi, M. Mahdavi, and S. Shirani. Steganalysis of pixel-value differencing steganographic method. In *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 292–295. IEEE, 2007.

[26] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Adversarial generative nets: Neural network attacks on state-of-the-art face recognition. *arXiv preprint arXiv:1801.00349*, 2017.

[27] Mei-Yi Wu, Yu-Kun Ho, and Jia-Hong Lee. An iterative method of palette-based image steganography. *Pattern Recognition Letters*, 25(3):301 – 309, 2004.

[28] Pin Wu, Yang Yang, and Xiaoqiang Li. Stegnet: Mega image steganography capacity with deep convolutional network. *Future Internet*, 10(6):54, 2018.

[29] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.