HIDING IMAGES WITH DEEP STEGANOGRAPHY

Kazim Sanlav
Bogazici University
34342 Bebek/Istanbul, Turkey
kazimsanlav@gmail.com

ABSTRACT

Steganography is the art of covered or hidden writing which dates back to the 15th century, when messages were physically hidden. wik (2020) With technological development, people applied it on images, sounds and videos. Traditional stegonagraphy applications have been using deterministic algorithms and Information Theory heavily. With recent development in Deep Learning, neural networks increases the success of the steganography methods. Baluja (2017) In this work, a neural network based algorithm is developed to hide secret image into a cover image. Source code is available at: https://github.com/kazimsanlav/Hiding-Images-with-Deep-Steganography .

1 INTRODUCTION

Steganography is the art of covered or hidden writing which dates back to the 15th century, when messages were physically hidden. wik (2020) With technological development, people applied it on images, sounds and videos. Traditional stegonagraphy applications have been using deterministic algorithms and Information Theory heavily. With recent development in Deep Learning, neural networks increases the success of the steganography methods. Baluja (2017)

Success of a steganography algorithm is based on mainly two factors: ratio of the amount of secret data bits to cover data bits and the structure of secret data. In image based steganography, ratio of bits in secret image to cover image can be measured with bits-per-pixel(bpp). On the other hand, if the cover image has a very simple structure, like blank image, it will be very hard to cover any secret image into it.

In this work both secret and cover images are obtained from MS COCO dataset and the bpp is equal to 1. Lin et al. (2014)

2 RELATED WORK

Baluja successfully demonstrated how neural networks can be used in the field of steganography in 2017. Baluja (2017) His model consist of Prep Network, Hiding Network and Revealing Network. Prep Network used for fixing the secret image size to cover image and preparing the secret image to be hidden by extracting useful features such as edges, textures and high frequency regions. Output of the Prep Network is used as input for the Hiding Net together with Cover Image. Hiding Net outputs Container Image and later this image inputted to Revealing Network to get back the secret image. Proposed network aims to minimize the total loss which consist of hiding loss and revealing loss. Hiding loss measures the pixel differences between cover image and container image while revealing loss measures the pixel differences between secret image and revealed image. Baluja shows the example outputs of model and analyse the results further by examining where the secret image are encoded, Figure 1.

Figure 1: Examples from the work of Baluja

Zhang et al. showed how GAN based models can be used in steganography in their more recent paper. Zhang et al. (2019) They use a novel architecture which is suitable to hide arbitrary binary data into an image, Figure 2. They experimented with upto 4.4 bpp ratio. As they used GAN based model they do not use real secret images, instead they used random noise as a secret image and use real cover images. Their model consist of Encoder and Decoder networks. Encoder is responsible for generating container image which encodes generated secret image into cover image. They experimented with three different architectures: Basic, Residual and Dense Encoder. Decoder takes the output of the Encoder and reveals the secret image from it.

Their loss term consist of 3 parts: decoding loss, similarity loss, realness loss. Also, their model is able to run with variable size of secret data (generated).

Figure 2: Examples from the work of Zhang et al.

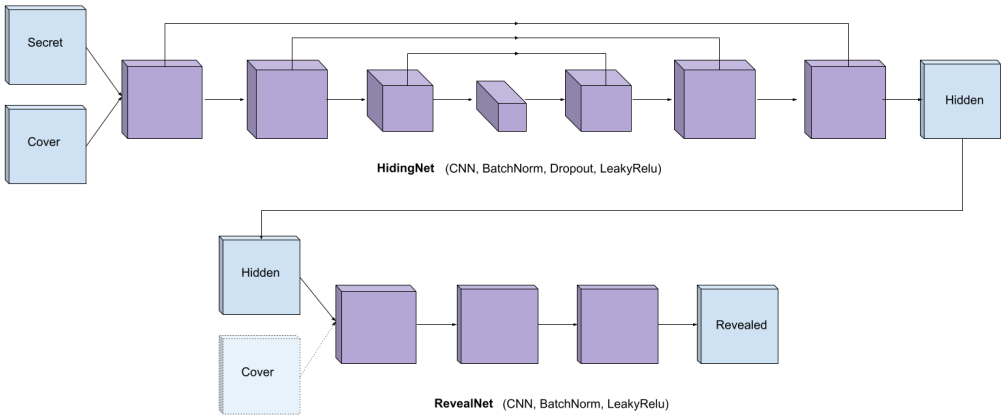
3 METHOD

In this work, encoder-decoder architecture is used which is similar to an autoencoder. Model consist of Hiding Net and Revealing Net, Figure 3. Hiding Net consist of Convolutional Blocks and skip connections. Skip connections help model to preserve objects' structures. U-Net like architecture is used with a bottle neck in the middle. Hiding Net inputs both secret image and cover image both are $3 \times 256 \times 256$ in size. It outputs Hidden Image which encodes both secret image and cover image. Hidden Image then fed into the Revealing Net together with Cover Image to get back Revealed Image. Model loss consist of revealing loss and hiding loss, similar to Baluja (2017).

MS COCO data set is used for training with approximately 40000 images and testing with 5000 images. Lin et al. (2014)

Various experiments have been done with different model architectures and proposed architecture was chosen as it was successful.

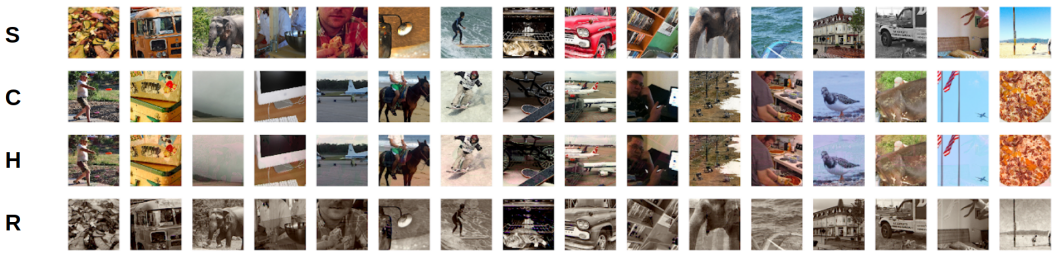
Figure 3: Model Structure



4 EXPERIMENTS

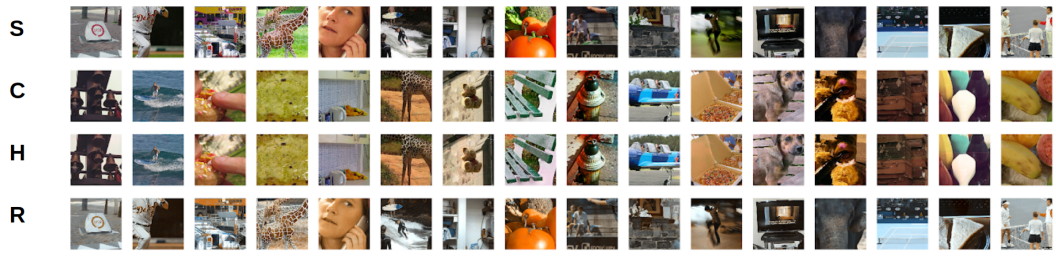
Up to some point in the training, revealed net was not able to reveal back all the color channels of the secret image, Figure 4.

Figure 4: Early Training: Secret, Cover, Hidden, Revealed



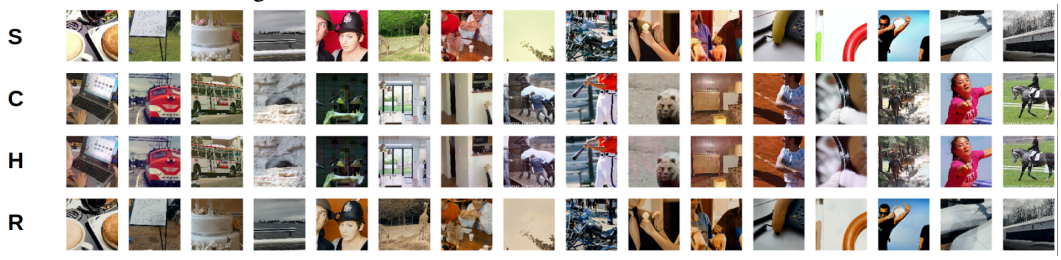
Then model able to encode all the color channels successfully, Figure 5

Figure 5: End of Training: Secret, Cover, Hidden, Revealed



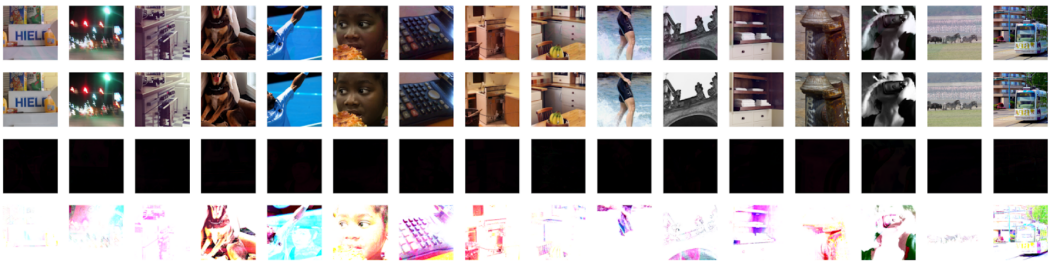
Test result was also successful as well, Figure 6

Figure 6: Test Dataset: Secret, Cover, Hidden, Revealed



To analyse the results further pixel differences between hidden and cover images are plotted, Figure 7. If model was not successfully hide the secret, it would be possible to recognize the secret image from the residuals. It can be seen from the results that the model was successful and residuals was not reveal the secret image. When the residuals are magnified by 10, they also does not reveal back the secret image in all cases except the 5th column where we can see the secret image in the form of a boy's head in the bottom right corner without color.

Figure 7: Hidden-Cover: Hidden, Cover, Residuals, 10×Residuals



Several metrics are used to quantify the model performance. Peak signal to noise ratio is used in measure image distortions. Given two images X and Y of size(W,H) and a scaling factor sc which represents the maximum possible difference in the numerical representation of each pixel, the PSNR is defined as a function of the mean squared error (MSE):

$$MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(X_{i,j} - Y_{i,j})^2$$

$$PSNR = 20\log_{10}(\text{sc}) - 10\log_{10}(MSE)$$

In our case, as images are represented as floating point numbers in [-1.0 , 1.0], then sc = 2.0 which is the maximum possible difference between two pixels.

Other useful metric is Structural Similarity Index between cover image and the revealed image. Given two images X and Y, the SSIM can be computed using the means, μ_X and μ_Y, variances, σ_X^2 and σ_Y^2, and covariance σ_{XY}^2 of the images as:

$$SSIM = \frac{(2\mu_X\mu_Y + k_1R)(\sigma_{XY}^2 + k_2R)}{(\mu_X^2\mu_Y^2 + k_1R)(\sigma_X^2\sigma_Y^2 + k_2R)}$$

The default configuration for SSIM uses k_1 = 0.01 and k_2 = 0.03 and returns values in the range [-1.0 , 1.0] where 1.0 indicates the images are identical.

Table 1: Test Dataset Results

MSE_hiding	8e-4
MSE_revealing	6.2e-6
MSE_total	4e-4
PSNR	58.1
mean SSIM_secret	90
mean SSIM_cover	95.3

5 Discussion & Future work

5.1 Discussion

In conclusion, deep learning can be applied to steganography. However, to use this kind of techniques in the real world, one should experiment and check all sort of boundary conditions. At the end, model learns it's objective function from the data supplied to it. Using more diverse datasets and testing the model against possible attacks would be beneficial.

5.2 Future Work

For the future work, I would like to investigate the model further, find out it's weak sides and try to improve it further.

References

Steganography, Jul 2020. URL `https://en.wikipedia.org/wiki/Steganography`.

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

T Lin, Michael Maire, Serge J Belongie, Lubomir D Bourdev, Ross B Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. arxiv 2014. *arXiv preprint arXiv:1405.0312*, 2014.

Kevin Alex Zhang, Alfredo Cuesta-Infante, Lei Xu, and Kalyan Veeramachaneni. Steganogan: High capacity image steganography with gans. *arXiv preprint arXiv:1901.03892*, 2019.