A CNN based invisible QR code generator for human living space

Kohei Yamauchi
Graduate School of Robotics and Design
Osaka Institute of Technology
Osaka, Japan
m1m19r36@oit.ac.jp

Hiroyuki Kobayashi
Graduate School of Robotics and Design
Osaka Institute of Technology
Osaka, Japan
hiroyuki.kobayashi@oit.ac.jp

***Abstract*—In this paper, the authors aim at embedding and restoring arbitrary information in an arbitrary image using CNN. In order to achieve this objective, we propose a model composed of two CNNs of different roles. In the proposed method, the QR code is used as the medium of information to be embedded. Thanks to the error correction nature of QR code, embedded information is expected to be restored without errors. We hope to use this technology to integrate QR codes into human living space,thereby enabling us to hide arbitrary information without damaging the atmosphere of it. The proposed CNN model was trained to test whether it is possible to embed a QR code image in a color image. As a result, it was possible to embed the QR code without deteriorating the image quality of the input color image. However, it has been found that embedding a QR code into a sharp color image can not restore the QR code correctly. From this result, it turns out that the proposed method can only adapt to blurry images at present, and we will aim to solve this problem in future research.**

***Index Terms*—Deep Learning, Convolutional Neural Network, Steganography**

I. Introduction

In recent years, method using a Convolutional Neural Network (hereinafter referred to as CNN) have been successful in image classification problems. For example, in the field of image generation, models for generating images using CNN, such as GAN [1] and DCGAN [2], and algorithms for converting the style of images [3] have been proposed. Here, an algorithm for converting the style of the image will be described. When an image is input to CNN trained for object recognition, in the hidden layer, the information on the color and texture of the object in the image weakens, and the information on the shape of the object becomes stronger. Therefore, by replacing the weakened part of the information with the style information of another image, it is possible to generate an image in which only the style has changed. Now, We thought that if it was possible to replace part of the information in the input image with other arbitrary information using the above method, it would be possible to embed arbitrary information in part of the image. We also thought that it would be possible to restore the embedded information by extracting any information embedded in the image as a feature.

Here, as a technique for embedding other information in an image, there is one called steganography. Steganography is a technology that embeds important information in information media such as images and sounds so secretly that watchers or listeners would not notice the modification. Conventional methods of steganography include a method of directly changing pixel values such as luminance information and color information of an image in order to embed information in the image, and a method of changing frequency components (amplitude, phase) in the image [4]. In this paper, we use CNN to embed arbitrary information in a color image, and to restore the embedded information, we use the QR code as a medium of the information to be embedded in the image. By using the CNN model proposed in this paper, it is possible to embed arbitrary information without waste, and to restore an image that can correctly read QR code information.

We hope to use this technology to integrate QR codes into human living space and thereby becomes possible to hide arbitrary information without damaging the atmosphere of it. The rest of the paper is organized as follows. Section II gives details on CNN model. Section III gives the details of the experiment. Section IV gives the discussion from the results of the experiment. Section V gives a summary and the future work,and concludes this paper.

II. Methods

Fig. 1 shows that the configuration of the proposed method. As shown in Fig. 1, proposed method is composed of two pairs of CNNs with different roles, an encoder CNN for embedding a QR code image into a color image, and a decoder CNN for recovering the embedded QR code image. We aim to correspond to various kinds of color images by using CNN. The design of these CNNs is based on the paper [5] of Rafia Rahim et al.The remainder of this section describes the details of two CNNs with different roles.

A. Encoder CNN

The role of the encoder CNN is to generate an embedded image O_e from the color image I_h and the QR code image I_g.The encoder CNN is configured as Fig. 2. Also, the details of the encoder CNN is shown in the table Fig. 3.

978-1-7281-4878-6/19/$31.00 ©2019 IEEE

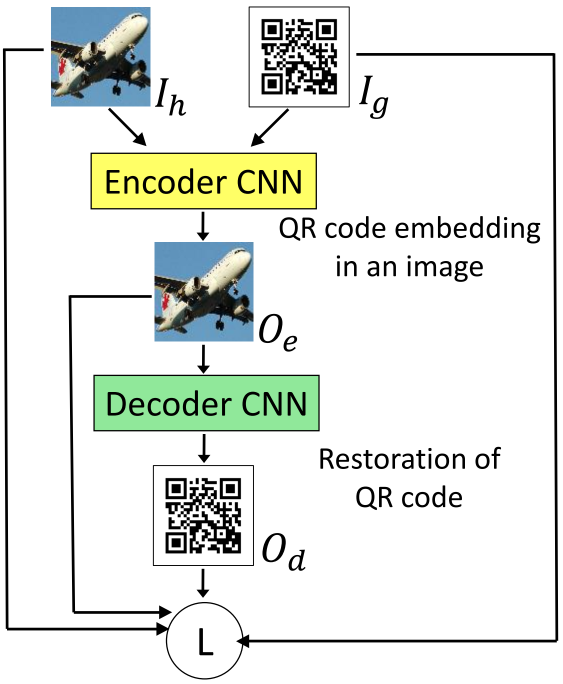

Fig. 1. Proposed CNN Model

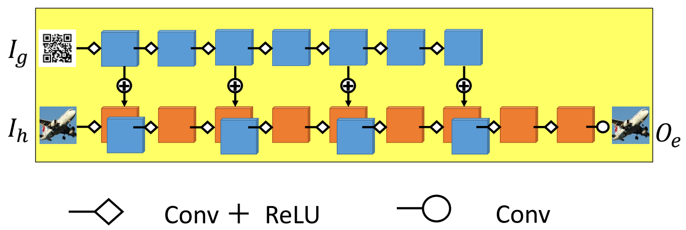

Fig. 2. Encoder CNN

	Input	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv
I_h	100 × 100 × 3	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16			
		concat		concat		concat		concat	1 × 1 × 16	1 × 1 × 8	1 × 1 × 3
I_g	100 × 100 × 1	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16	3 × 3 × 16			

Fig. 3. Details of Encoder CNN

The encoder CNN has an input layer for color images and an input layer for QR code images, and the color image and the QR code image are processed respectively. On the QR code image side, low level (edge, color, etc.) and high level features are extracted from the QR code image I_g using the convolution layer and the activation function ReLU (ramp function).On the color image side, features are extracted using a convolution layer and an activation function. Thereafter, the feature extracted from the QR code image and the feature extracted from the color image are combined by performing processing with the convolution layer and the activation function.This procedure is repeated many times, and then the feature extracted on the QR code image side is completely synthesized with the feature extracted from the color image, and the convolution process of the QR code image is ended.Thereafter, the process using the convolution and the activation number is repeated several times to generate the embedded image O_e. After processing of the convolutional layer in which O_e is generated, processing with the activation function is not performed.

B. Decoder CNN

The role of the decoder CNN is to restore the embedded QR code image from the embedded image O_e.The decoder CNN is configured as Fig. 4. Also, the details of the decoder CNN is shown in the table Fig. 3.

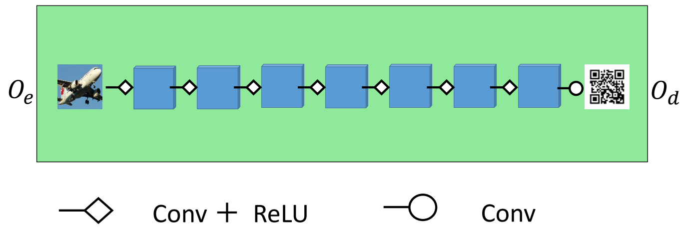

Fig. 4. Decoder CNN

	Input	lambda	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv+Relu	Conv
O_e	100 × 100 × 3	100 × 100 × 1	3 × 3 × 16	3 × 3 × 16	3 × 3 × 8	3 × 3 × 8	3 × 3 × 3	3 × 3 × 3	3 × 3 × 1

Fig. 5. Details of Decoder CNN

The decoder CNN receives the embedded image O_e generated by the encoder CNN as an input, and uses the convolution layer and the activation function ReLU (except for the final layer not processed with the activation function) to obtain Oe The embedded QR code image is restored to generate a restored image O_d.The embedded image O_e input to the decoder CNN is converted from the 3-channel color image to a gray-scale image in the lambda layer of Fig. 5 before processing in the convolution layer. The conversion equation from color image to gray scale image is as follows. R, G and B are respectively the R component, G component and B component of the color image.

$$Gray = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B \quad (1)$$

When training a CNN model, concatenate two CNNs and use the loss function L to update the weights of the convolutional layer.

$$L(I_g, I_h) = ||I_h - O_e||^2 + ||I_g - O_d||^2 + \lambda(||W_e||^2 + ||W_d||^2) \quad (2)$$

Here, λ represents the regularization parameter. W_e and W_d represent the learned weights of the encoder and decoder CNN, respectively. By simultaneously updating the filter weights of the encoder CNN and the decoder CNN, it is possible to generate an embedded image O_e in which the features of the embedded QR code image remain without impairing the image quality of the color image I_h. During training, two pairs of CNNs are linked, but when actually embedding and restoring a QR code image, CNNs corresponding to each can be used alone.

III. EXPERIMENT

In this section, we report our experiments. We trained a model consisting of two CNNs, which is a proposed method. And it experimented whether it was possible to embed and restore a QR code image to a color image in a model after learning. In the experiment, three different color images were used to investigate the difference in the result by the image. We also conducted experiments to evaluate the quality of the embedded image and read the restored QR code. Details of each experiment are described below.

A. Training CNNs

First, We train CNNs. We input the color image I_h and the QR code image I_g into the encoder CNN. At this time, the image size of both images is converted to 100×100, the color image is not normalized, and pixel values are input to CNN in the range of 0 to 255. In addition, We process the QR code image which is a binary image so that the pixel is 0 or 1 and input this. Then, the output embedded image O_e is input to the decoder CNN, and the restored image O_d is output. During training, as shown in Fig. 6, the output of encoder CNN and the input of decoder CNN are connected, and the weights are updated simultaneously. In training, color images use an image dataset called CIFAR-10. Also, the QR code image is generated using python's "qrcode" library. We prepared 10000 test images for use in calculation of learning accuracy for each 50000 input images for training. Perform learning with Epoch 500, batch size 32, weights 0.0001, optimization algorithm Adam [6], and a learning rate of 0.0001.

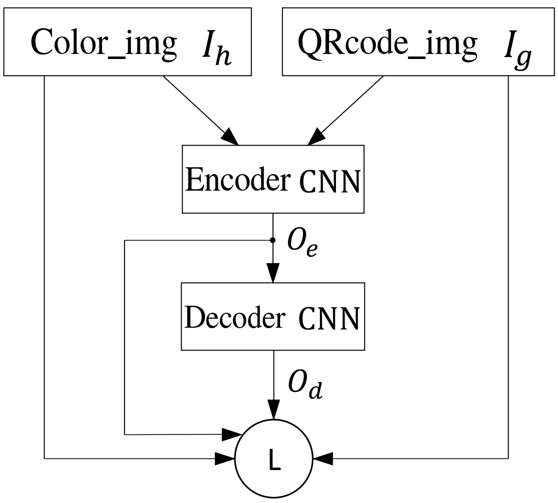

Fig. 6. CNN Model at Training

B. Experiment 1:QR code image embedding for color image

Embedding the QR code image I_g into the color image I_h using the CNNs that has undergone training. Also, restore embedded QR code image from embedded image O_e and create image O_d. First, enter the color image I_h and the QR code image I_g into the encoder CNN. The input image uses the test image used to learn the image embedded CNN. Second, the output embedded image O_e is PNG imaged using OpenCV and compared with the input color image I_h. Third, the data of the embedded image O_e outputted from the encoder CNN is inputted as it is to the decoder CNN, and a restored image O_d is generated. Final, the restored image O_d is PNG imaged using OpenCV and compared to the input QR code image I_g.

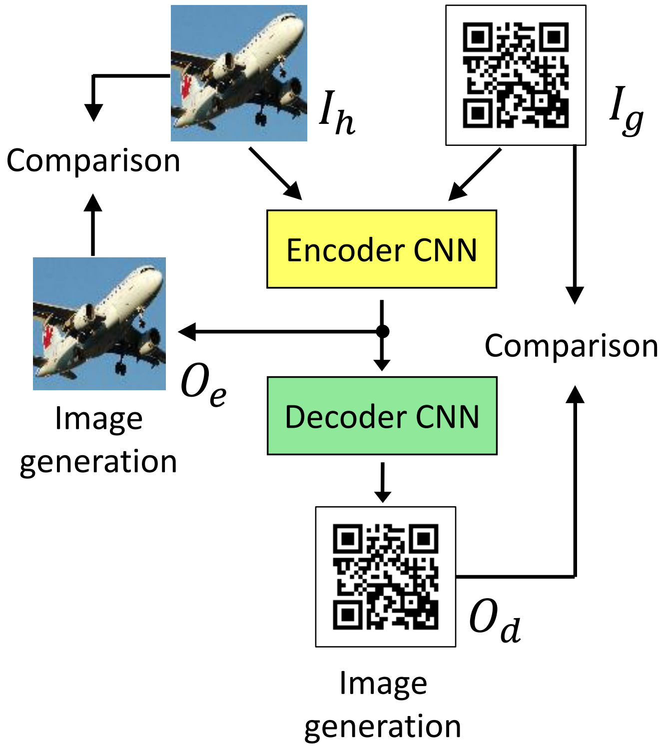

Fig. 7. Outline of Experiment 1

C. Experiment 2:Embedding and restoring images with high sharpness

The color image used to embed the QR code in Experiment 1 is strongly blurred because the images in CIFAR-10 dataset were enlarged from the original size of 32x32 pixels. So, in this experiment, embedding and restoration of the QR code image were performed to what changed the image size of the image whose original resolution is larger than 100x100. An ImageNet [7] image was used as a color image to be input to the encoder CNN. The image of ImageNet was resized to 100×100 and used as a color image I_h for input.First, as in Experiment 1, enter the color image I_h and the QR code image I_g into the encoder CNN.Second, the output embedded image O_e is PNG imaged using OpenCV and compared with the input color image I_h. Third, the data of the embedded

image O_e outputted from the encoder CNN is inputted as it is to the decoder CNN, and a restored image O_d is generated. Final, the restored image O_d is PNG imaged using OpenCV and compared to the input QR code image I_g.
In addition, an image was created in which the contour of the CIFAR-10 image was enhanced using a sharpening filter, and experiments were conducted to see if the image could be embedded and restored with QR code. The image of CIFAR-10 resized to 100 × 100 was processed with the Fig.9 sharpening filter to generate an image with enhanced outline. This image is input as a color image I_h to the encoder CNN to output the embedded image O_e, and the decoder CNN is restored to compare the embedded QR code with the result of Experiment 1.

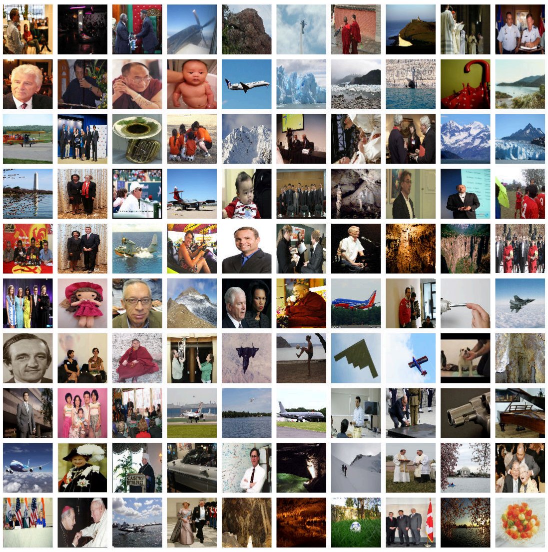

Fig. 8. ImageNet Dataset

0	−1	0
−1	5	−1
0	−1	0

Fig. 9. Sharpening Filter

D. Experiment 3:Evaluation of embedded and restored images

Evaluate the image quality of the embedded image O_e. The peak signal-to-noise ratio (PSNR) and SSIM [8] were used as evaluation indicators. Calculating PSNR and SSIM use Scikit-Image which is a library of image processing algorithms. The embedded image is generated using the test image, and the image quality is evaluated using two evaluation indices. Next, Input embedded image O_e to decoder CNN and recover the embedded QR code image O_d. To read the QR code, use ”Zbar”, an open source software for reading barcodes.In the experiment, experiments were conducted in three cases of CIFAR-10, ImageNet, and sharpened CIFAR-10. In each case, 10000 color images and QR code images were prepared and experimented.

E. Results

We embedded the QR code image in the color image with CNNs after training and restored the embedded QR code image. Fig. 10 shows the input image and the output image as PNG images when the image of CIFAR 10 is used as a color image.

Fig. 10. Input Image and Output Image (CIFAR-10)

Next, the result of embedding and restoring a QR code using an image of ImageNet as a color image as an input is a Fig. 11.

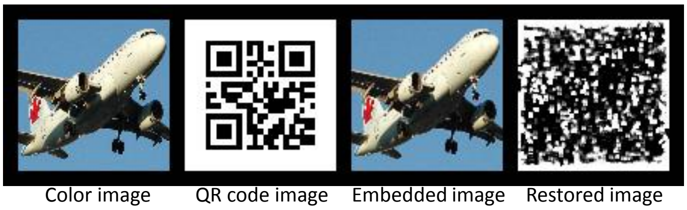

Fig. 11. Input Image and Output Image (ImageNet)

Next, the image of CIFAR-10 processed by the sharpening filter is embedded as a color image with a QR code embedded. The images before and after the sharpening filter processing are shown in the Fig. 12. The result of restoration is Fig. 13.

Fig. 12. Sharpening Filter Processing

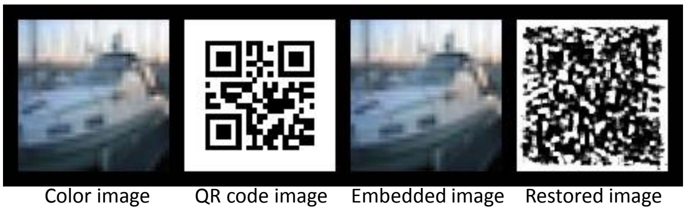

Fig. 13. Input Image and Output Image (CIFAR-10 with Sharpening Filter Processing)

Also, TABLE I shows the evaluation of the image quality of the embedded image and the reading result of the restored image.

TABLE I
EVALUATION RESULTS

	PSNR(dB)	SSIM(%)	Reading rate(%)
CIFAR-10	46.52	99.47	99.25
ImageNet	28.78	90.85	0
CIFAR-10 (Sharpening)	42.73	99.30	0

IV. DISCUSSION

A. Experiment 1

Looking at the results of Experiment 1, The image output by the encoder CNN does not show any change in appearance compared to the input color image. Also, the input QR code image and the restored QR code image are compared. The restored QR code image output by the decoder CNN has some parts where the image is broken, but the rough shape of the input QR code image is reproduced.

B. Experiment 2

Looking at the results of Experiment 2. The image output by the encoder CNN when using an ImageNet image for the input color image shows some degradation as compared to the input color image. Also, the input QR code image and the restored image are compared. The original QR code image did not remain in the image output by the decoder CNN. When using an image obtained by sharpening the image of CIFAR-10 for the input color image, the image output by the encoder CNN does not show any change compared to the input color image. However, the original QR code image did not remain in the image output by the decoder CNN as in the case of using the ImageNet image.

C. Experiment 3

Looking at the results of Experiment 3, it can be said that high quality is realized when the PSNR value is 35 dB or more and the SSIM value is 0.98 or more as a standard. When CIFAR-10 is used for color images, comparing these values with the experimental results, both values exceed the values that are said to be high quality. With ImageNet images, both are lower than what is called high quality. In addition, when using an image obtained by processing the image of CIFAR-10 with a sharpening filter, both PSNR and SSIM values exceed the values that are said to be of high quality. However, the embedded QR code could not be restored.

D. Whole discussion

As a result of experiment, the result was different depending on the color image to embed the QR code. From the results of Experiment 1 and Experiment 3, when the CIFAR-10 image was used, the QR code image could be embedded without losing the image quality of the input color image, and the embedded QR code image could be restored. According to the results of Experiment 2 and Experiment 3, when an image of ImageNet is used, when a QR code image is embedded in a color image, the output image is significantly degraded compared to the input color image. Also, when trying to restore a QR code image from an embedded image, an image without the original QR code prototype was being output. In addition, embedding a QR code image into a CIFAR-10 image processed with a sharpening filter, the quality of the embedded image was good, but the embedded QR code image could not be restored. From this result, it can be considered that the current proposed method can not cope with the rapid change of color tone. If sharp color images are used from Experiment 3, the deterioration of the image quality of the embedded image is large, so it may be considered that embedding of the QR code image with the encoder CNN is not successful.
Also, in Experiment 1 and Experiment 2, the output of the encoder CNN is inputted to the decoder CNN as it is. Under the present circumstances, it is not possible to restore the embedded QR code image even if the embedded image that has been reread after imaging the output of the encoder CNN is input to the decoder CNN.

V. CONCLUSIONS

A. Summary

In this paper, we aimed to embed arbitrary information in an image, and proposed a model composed of two pairs of CNNs with different roles to do that. By using the proposed method, CNN was able to embed and restore arbitrary information in the image. By using the proposed method, CNN was able to embed and restore arbitrary information in the image. However, under the present circumstances, if the QR code is embedded in an image in which color density changes rapidly in a color image such as an image in which an edge is emphasized, the embedded QR code can not be restored correctly.

B. Future work

When the proposed method is used in human living space, the current proposed method can only adapt to blurred images. If it is used in the human living space in this state, there is a risk of destroying the atmosphere of the space. In addition, since the QR code can not be restored if the output of the encoder CNN is not input as it is, in the future, research will be conducted with the goal of solving these problems.

By solving these problems we hope to be able to cover information in the human living space and make people's lives more comfortable.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[4] R. Krenn, "Steganography and steganalysis," *Retrieved September*, vol. 8, p. 2007, 2004.

[5] A. ur Rehman, R. Rahim, M. S. Nadeem, and S. ul Hussain, "End-to-end trained cnn encode-decoder networks for image steganography," 2017.

[6] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.