

Using AI for Designing ID cards Embedded with Invisible QR code

Chirag Jain ^a, Dhruva Mhatre ^b, Prem Vispute ^c, Kiran Deshpande ^d, Kaushiki Upadhyaya ^e

^a A.P. Shah Institute of Technology, Thane, India, chirag3635@gmail.com

^b A.P. Shah Institute of Technology, Thane, India, dhruvamhatre@apsit.edu.in

^c A.P. Shah Institute of Technology, Thane, India, premvispute@apsit.edu.in

^d A.P. Shah Institute of Technology, Thane, India, kbdeshpande@apsit.edu.in

^e A.P. Shah Institute of Technology, Thane, India, ksupadhyaya@apsit.edu.in

Abstract

Organizations strongly encourage the usage of Identification cards (ID cards) as a measure of security. ID cards are used to reliably identify a person and, as a result, serve to improve the safeness of an organization by preventing unauthorized individuals from gaining access to the premises. However, these ID cards contain the user's personal information, which must be protected in today's world. This paper proposes an approach for obtaining a data hiding mechanism for personal information on ID cards by using user data to develop a QR (Quick Response) code and embedding that QR code into the user image, making the QR code invisible to the naked eye. QR codes are well-known for their error-correcting mechanism and are particularly good in hiding arbitrary information. Convolution neural networks (CNN) are a class of Deep neural networks that helps us analyze visual imagery and recognize patterns. The objective of this paper is to use the CNN mechanism to hide a QR code in a user image. The proposed model is comprised of two CNNs (encoder CNN and decoder CNN). The encoder CNN's function will embed the QR code into the user image and generate an output image that appears just like the user image. The decoder CNN's role will be to take the encoder CNN's output as input and generate the embedded QR code image as output. By ensuring enhanced security and hiding critical information, our approach prohibits the duplication of the general QR code, preventing its misuse.

Keywords

Convolution neural network; Deep Learning; QR code; Steganography; Data hiding; Security; Artificial Intelligence.

1. Introduction

Identification cards or ID cards are the most important credential for any organization, and it helps uniquely identify the person or verify them. However, these ID cards may contain some personal information like the mobile number or the person's address. Such crucial information, if fallen into the wrong hands, could be misused. Hence, here we are proposing a system where all the information of the user will be used to generate a QR code and that QR will be embedded in the user's image such that it will be invisible to the naked eyes. The main aim of our project is to develop a system that will embed the QR code (containing arbitrary user information) into a coloured image (User photograph), such that the QR image will be hidden in the background and the primary visible image will be the user image [1]. We are using the methodology of convolution neural networks and Image Steganography for developing this system. Convolution neural networks (known as ConvNet or CNN) is a branch of Deep learning that is commonly applied to analyzing visual imagery. In the fields of image classification and recognition, CNN has been proven to be efficient. A CNN model works by extracting necessary features from images in the form of pixels, which helps in reducing the image size and provides us with data that is important in image recognition. CNN has multiple layers, including the convolutional, non-linearity, pooling, and fully-connected layers [2]. When a CNN trained model receives an image as input, the first layer that acts on it is the convolution layer. This layer consists of a filter/feature detector (of a specified size) which moves all over the image matrix, multiplying its values with the original pixel values. After every convolution layer, a non-linearity layer is added. The pooling layer is generally used to further downsize the image by extracting more important pixels but using certain functions like Max pooling, Min pooling, etc. This helps reduce the network's training time by reducing the number of computations required. Steganography is a method for hiding information (text, image, etc.) inside a cover image such that the very existence of the secret information is unknown. In our model, we are aiming to use Convolution neural networks to achieve Image Steganography which will consist of two CNNs; encoder CNN and decoder CNN. The role of encoder CNN will be to take an input of the QR code image and colour image and generate an output of

coloured image embedded with the QR code image such that the QR image will be undetected by the naked eyes. The role of decoder CNN is to recover the original QR code image from the coloured image [10].

1.1. Literature Survey

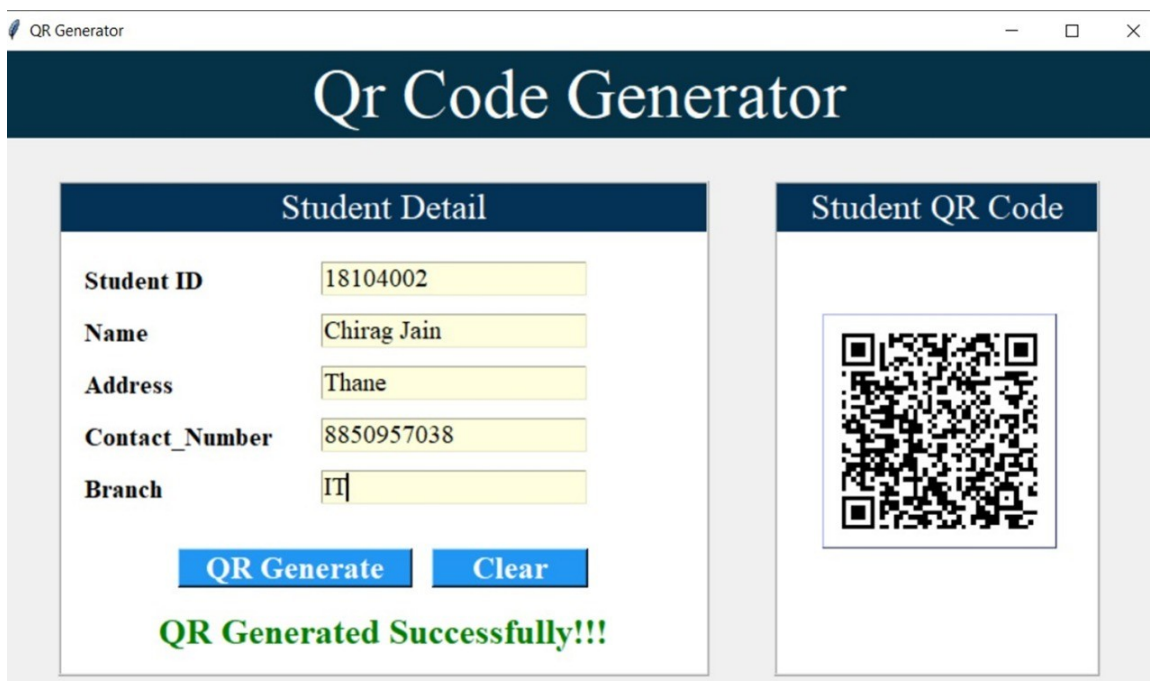
The most popular form of Steganography used nowadays is using images as a source of hiding data. These methods involve hiding messages by altering “noisy” areas by using methods like a least-significant bit or LSB, which directly alters the image pixels value [3]. This image steganography method limits the capacity of data that can be hidden in the carrier image; if more data is added, it may distort the image leading to a significant difference between the original image and the stego-image. To overcome this drawback, the authors of [4] proposed a method for image steganography using Fully Convolutional Dense Network (FC-DenseNet). Convolution neural networks are easier to train if they have shorter connections and perhaps produce more accurate and efficient results. Dense convolutional neural networks (hereafter referred to as DenseNet) connect each layer to every other layer in a feed-forward fashion [5]. DenseNets can be considered as an extension of Residual Networks (ResNets) [6], where each layer obtains additional inputs from all preceding layers and passes on its feature-maps to all subsequent layers, wherein Resnet we combine features through Summation, whereas in DenseNet, features are combined through the method of concatenation before passing into another layer [5]. FC-DenseNets is mainly used for semantic segmentation. It uses an upsampling layer for recovering the spatial resolution of the input or output layer. The information from additional dense blocks of the same resolution is combined in the upsampled dense blocks, and the higher resolution information is obtained via the standard skip link between the upsampling and downsampling channels [7]. We are using FC-DenseNets for hiding the image, as shown in [4]. Here the authors are hiding information (secret image) into a carrier image in such a way that the stego-image will be visually similar to the carrier image. In our project, we use this FC-DenseNet method to hide a QR code image into a carrier image, as shown in [1]. We are generating a QR code consisting of user data (such as phone no., address, email, etc.) and are embedding the QR code image into the user image. The goal of this system will be to provide data hiding through Cryptography and Steganography, such that the user data will be hidden through QR generation and hiding the QR code through Steganography so that there will be no knowledge of the existence of the QR in the first place

1.2. SYSTEM ARCHITECTURE

Our proposed system will be developed in order to achieve steganographic mechanisms by hiding the QR image (consisting of user data) into the user's photograph (carrier image) such that the QR image will be invisible to the naked eyes. The entire system will mainly consist of a QR code generator for generating the QR code image consisting of arbitrary user information, and the CNN network will mainly consist of three sections which are preparation network, hiding network, and reveal network. All these networks will collectively form an end-to-end system for encoding as well as decoding the image [8]. Figure 2 illustrates the suggested system's workflow.

1.2.1 QR Code Generator

The QR Code is a two-dimensional matrix code that transmits information by arranging dark and light elements, known as "modules," in columns and rows and are widely known and used for their error-correcting mechanism [9]. Our system's QR code generator model is responsible for generating a QR code by taking user information as input. We have used the pyqrcode library of python for developing a GUI based QR generator where user data such as name, number, address, etc. will be provided to the system as input, and as an output, the system will generate a QR code encrypting the given information as shown in the figure 1.



QR Generator

Qr Code Generator

Student Detail	
Student ID	18104002
Name	Chirag Jain
Address	Thane
Contact_Number	8850957038
Branch	IT

QR Generate **Clear**

QR Generated Successfully!!!

Student QR Code




Fig. 1. Demonstration of QR code generator

1.2.2. Encoder CNN

The mechanism of encoder CNN is to embed the QR image generated in section A. into the user image such that the QR image will not be visible to the naked eyes, and the output image of encoder CNN will be visually similar to the user image. The encoder CNN comprises of two networks which are the Prep-network and the Hiding Network. The Prep-Network is responsible for preparing the secret image to be hidden. It serves two purposes; if the secret image is smaller than the cover image, the Prep-Network will increase the size of the secret image progressively in order to distribute the secret image's bits across the pixels of the cover image. The more important purpose of the Prep-network is to extract more useful features such as textures or edges for succinctly encoding the image. The Prep-Network comprises 2 convolution layers of 65 filters (50 3x3 filters, 10 4x4 filters, and 5 5x5 filters). In the Hidden Network, the output of the Prepnetwork along with the cover image will be given as the input, and it will generate a container image as the output to it. This network is responsible for concatenating the convoluted RGB channels of the cover image and the extracted channels of the secret image and generating a container image that will be visually similar to the carrier image. The best-suited architecture for this network consists of 5 convolution layers that have 65 filters (50 3x3 filters, 10 4x4 filters, and 5 5x5 filters).

1.2.3. Decoder CNN

The mechanism of decoder CNN is to extract the QR code image from the output image of the encoder CNN. It takes the output image of the encoder CNN as input and provides us with the original QR image which was embedded in the previous user image. The decoder CNN comprises of the Reveal Network whose role is to reveal the original secret image from the container image (output of encoder CNN). This network only takes the container image as an input and provides us with a decoded secret image. The architecture for this network is the same as for the hiding network, i.e., it consists of 65 filters (50 3x3 filters, 10 4x4 filters, and 5 5x5 filters).

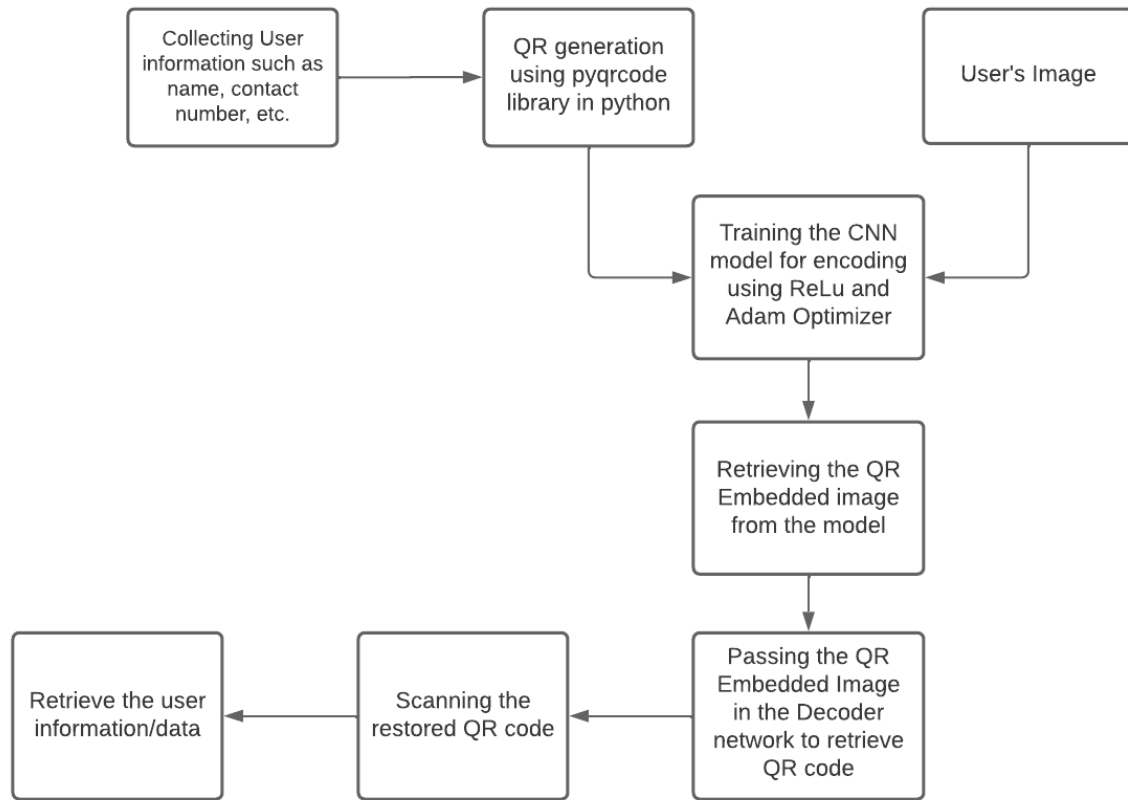


Fig. 2. The Proposed architecture of CNN Model

1.3. TRAINING THE CNN MODEL

For training the CNN model, we are using the Tiny ImageNet dataset, which consists of 2000 images of dimension $64 \times 64 \times 3$. Firstly, we are training the model for a randomly picked pair of images from the tiny ImageNet dataset, such that half of the images are used as cover images, and the other half is used as the secret image. In the first phase, the dataset is processed using the Keras image library and the Adam optimizer. The features of the secret image are extracted in the Prep network of the Encoder CNN, and then, the cover image and the output of Prep-Network are passed as input in the Hidden network of the Encoder CNN. At this time, the image size of both images is converted to 64×64 , the colour image is normalized, and pixel values are input to CNN in the range of 0 to 255. We get a container image that will contain the embedded secret image into the cover image such that the container image will be visually similar to the cover image. In the second phase, we give the outputted container image as input to the Decoder CNN and restore the secret image from it. The encoder and decoder CNN are trained simultaneously during training, and the weights are updated simultaneously.

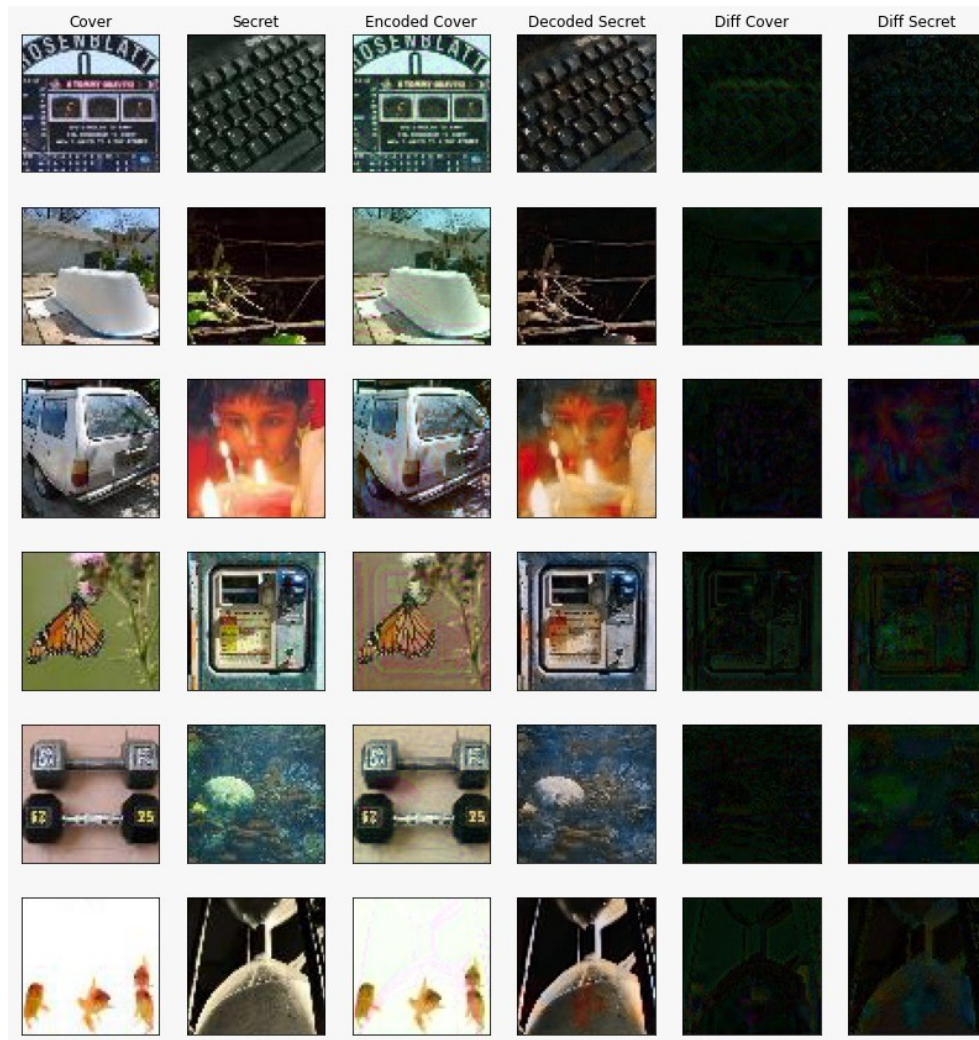


Fig. 3. Result of Training Tiny ImageNet dataset

We have trained the model with a batch size of 32 and performed learning with epoch 300, and after every epoch 40, we generate an output, as shown in Figure 3. We have used the optimization algorithm Adam, weights 0.001, and a learning rate of 0.001. The Rectifier Linear Unit (ReLU) is used as an activation function to train this layered network, which reduces the model training computing needs by addressing the vanishing gradient problem. We used a sequential model with multiple layers consisting of Conv2D, padding, strides, and Batch-normalization blocks.

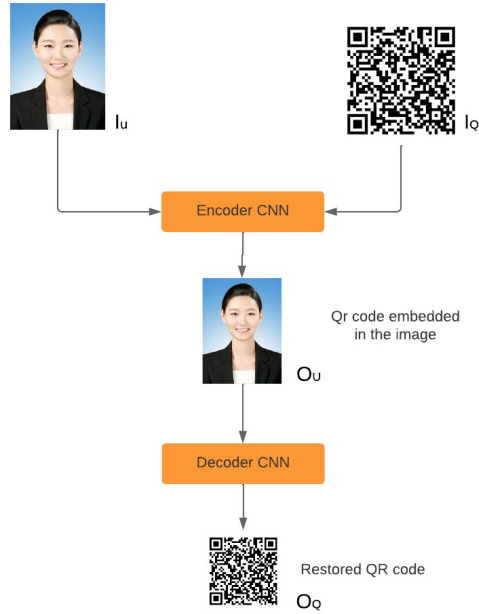


Fig. 4. Proposed Encoder-Decoder CNN Model

After the model training is completed, we will use this model for embedding QR codes into colour images. For the cover image, we will be using the Tiny ImageNet dataset, and for the secret image, we will use the QR-DN1.0 dataset for importing the QR code. We will train the CNN model using these datasets as we did in the previous phases. Once this CNN model has undergone training, we will use it to embed our QR code image I_q into the User Image I_u and restore the embedded QR image O_q from container image O_u , as shown in figure 4. Firstly, provide the User Image I_u and the QR code image I_q as the input to the Encoder CNN; embedded image O_u will be the output to this CNN. Then this generated embedded image O_u will then be inputted to the decoder CNN, generating the restored image O_q .

2. Conclusion

The emerging technological advancements have increased the risk of data breaching, thus making data security and privacy our primary responsibility. Identification cards [ID-Cards] are essential for any organization and are used as proof of identification and verification. These ID cards contain personal information about the user, such as its name, number, address, etc. This information is essential and can be misused if fallen into the wrong hands. The model proposed in this paper provides an adequate data hiding mechanism for protecting this sensitive data. Here we are achieving data security by generating a QR code consisting of the user's information, and then we will embed the generated QR are into the user's image such that the very existence of the QR will be unknown, i.e., the QR code

will be invisible to the naked eyes. The concepts of Convolution neural networks are used to achieve the steganographic mechanism where the entire system consists of two CNNs (Encoder CNN and Decoder CNN) for embedding and decoding the QR code. The proposed system can be beneficial in providing data security and thus can prevent a data breach. Further, we can even add developments in the system for managing and maintaining the attendance system. Including web development mechanisms, we can develop a system for marking the attendance of students/employees simply by scanning the ID cards. This can make attendance marking quick and effective processes and avoid the chances of proxy marking or false attendance, thus increasing the organization's efficiency. We would research and work upon adding these developments to the proposed systems in the future.

3. REFERENCES

- [1] K. Yamauchi and H. Kobayashi, "Invisible QR Code Generator Using Convolutional Neural Network," IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, 2020, pp. 4009-4014, doi: 10.1109/IECON43393.2020.9254709
- [2] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.
- [3] Krenn, Robert. "Steganography and steganalysis." (2004): 2007.
- [4] Xintao, Duan, and Liu Nao. "Hide the image in fc-densenets to another image." arXiv preprint arXiv:1910.08341 (2019).
- [5] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
- [6] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [7] Jegou, Simon, et al. "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017.
- [8] Baluja, Shumeet. "Hiding images in plain sight: Deep steganography." Advances in neural information processing systems 30 (2017).
- [9] Gao, Zhongpai, Guangtao Zhai, and Chunjia Hu. "The invisible qr code." Proceedings of the 23rd ACM international conference on Multimedia. 2015. [10] Yamauchi, Kohei, and Hiroyuki Kobayashi. "A CNN based invisible QR code generator for human living space." IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society. Vol. 1. IEEE, 201