# Real time robust Invisible Hyperlinks in Physical Photographs based on embodied AI platform

Kang Chen
*School of Optical-Electrical and Computer Engineering*
*University of Shanghai for Science and Technology*
*Shanghai, China*
*1712440708@st.usst.edu.cn*

Fuwang Yi, Jun Jia, Guangtao Zhai*
*Institute of Image Communication and Network Engineering*
*Shanghai Jiao Tong University*
*Shanghai, China*
*{ yifuwang, jiajun0302, zhaiguangtao}@sjtu.edu.cn*

## Abstract

*Quick Response (QR) Code plays an important role in connecting the physical world where we live in and the digital world that contains information. However, the pictures made up of black and white color blocks really affect people's moods. This paper designs a visual appealing two-dimensional code to help people access information from offline to online. A Deep Neural Network (DNN) based encoder hides information into natural images and a DNN based decoder recovers it. A novel finder pattern is designed to help the decoding device to locate our codes quickly. We design a decoding device. It consists of an AI chip, camera, and LCD screen. The information is invisible in our generated hidden images but detectable by our device. With the help of powerful edge computing, the device can recover the hidden information from the pictures in real time.*

**Keywords:** Information Hiding and Recovery, Generative Adversarial Network, Edge Computing

## 1. Introduction

Nowadays, Quick Response (QR) code plays the role of transmitting information from the digital world to our physical world. However, QR code not only looks ugly but also takes up places. Thus, it is desperately to develop a state-of-the-art information medium to replace QR code.

The imagination of this technology is not only another QR code. In the future, in the 5G era, everything will be connected, AIOT devices will be everywhere, and VR / AR will no longer be out of reach. Imagine that you are wearing mixed reality glasses, which are equipped with powerful edge computing capabilities and high-speed 5G capabilities. On the street, advertisements are scrolled on the billboards on the side of the road. The advertisements look the same as usual, but the hidden information is embedded in the advertisement's screen with our technology. When your head turns to it, the mixed reality glasses scan it and upload it to the cloud. After the system responds,you will know about recent promotions from this merchant and other fun places around. As for merchants, they can personally recommend products you like based on how often and how often you see different ads.

The novel medium is supposed to have both visual quality and decoding robustness. This paper implements a complete system to generate an information code like a natural image. To maintain visual quality, we use the model proposed by Jia et al.[1] to hide information into natural images. The model of Jia et al. consists of an encoder to generate the codes and a decoder to recover the information. We reproduce this model runs on an AI chip-based device that is portable, low power, and real-time image responding. An effective finder pattern is designed to help our device to detect the location of the codes quickly.

## 2. Methods

The core device of our demo system is an NVIDIA Nano artificial intelligence development board which consists of a camera, a 4.3-inch LCD screen, and an efficient edge computing core. We reproduce the model of [1] on the development board to build our offline-to-online inference model. With the help of the beforementioned model, we can generate natural images with hidden information. In addition, we design an effective finder pattern to locate the generated images quickly.

### 2.1 Neural Network Model

In the offline learning process, we built a Generative adversarial network model [2] to generate pictures with

978-1-7281-4272-2/20/$31.00 ©2020 IEEE
DOI 10.1109/MIPR49039.2020.00063

hidden information. It contains three parts, encoder, decoder, and Discriminator.[1] Discriminator is used to improve the quality of pictures with hidden information, as a part of GAN. The encoder is used to generate a residual image, which will be added to the raw picture, and become the picture with hidden information. To improve the quality of pictures, JND [3] is added to losses on the process of training model. The decoder is used to translate hidden information behind pictures. It designed in two parts, One is STN net and the other is the net with deep convolution layers and dense layers.

In order to make the model recognition effect more robust, we added image distortion including random noise, blur, light reflection rendering, JPEG compression, and 3D rotation and perspective projection.[1]

Our model training is based on the Tensorflow deep Learning framework. During the training process, the Adam optimizer for the encoder and decoder has a learning rate of 0.0001 and the discriminator in GAN uses the RMS optimizer with 0.00001 learning rate. At the beginning of training, the losses do not include the quality loss of image generation and the implementation of methods such as rotation, contrast change, and blurring to improve the robustness of the picture, to ensure that in the beginning stage, the decoder can stably and efficiently decode the hidden information of the picture. At first, the adversarial generation network will not be paralyzed. Later, as the number of training increases, the levels of image quality loss, picture rotation, and brightness change are gradually added to ensure that the entire training is performed in an orderly and controllable state.

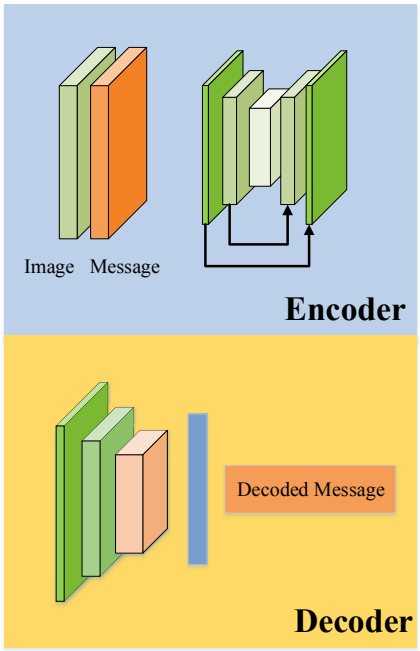


**Figure 1. Encoder and decoder**

## 2.2 Image localization Methods

We designed a new location scheme. The code is designed to have a black boundary around it, further surrounded by a white boundary and a black boundary. The width of the boundaries is equal. The location algorithm is based on a premise: the center of the camera is aimed at the center of the code. That is, the center of the captured image is within the boundary. This is reasonable because it is in line with human behavior to roughly align the code center when scanning the code at a short distance[4].


**Figure 2. The picture with location signs**

First of all, we take the center of the image as the coordinate origin and transform the image from Cartesian coordinate system to a polar coordinate system.

$$\begin{cases} r=\log\left(\sqrt{(x-x_0)^2+(y-y_0)^2}\right) \\ \varphi=\operatorname{atan2}\left(\dfrac{(y-y_0)}{(x-x_0)}\right) \end{cases} \quad (1)$$

As shown in the figure below.

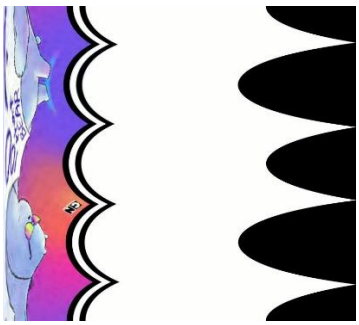

**Figure 3. Picture in polar coordinate system**

Our location scheme is based on the following features:

1. The boundary in the original image is a continuous curve in the polar coordinate system. And the φ is from π to -π.

2.Black boundary: White boundary: Black boundary = 1:1:1

3. If the boundary in the polar coordinate system is regarded as a function of r about φ, namely r = f (φ), four corners of the boundary corresponding to four local maximum of f (φ).

And then we do contour detection. Remove the contours with length less than 2π (based on the feature (1)). Scan along the r-axis and find the place with a black and white boundary ratio of 1:1:1 (based on the feature (2)). Traverse the contour points. If the r value of a point is greater than the r value of the point in its left and right neighborhood, the point is considered as the corner of the boundary (based on the feature (3)). Finally, the coordinates of corners are transformed from the polar coordinate system to Cartesian coordinate system.

### 2.3 Real-time identification device

The real-time identification device consists of three parts, they are a display module, a high-definition camera module, and a high-performance computing module.

The high-performance computing module is power by Nvidia Jetson Nano, an AI platform. Jetson Nano has 472GFLOP computing power (128 Cuda care). With the Tensorflow neural network framework adapted on Jetson Nano, it can quickly run AI algorithms.

At the same time, it has powerful graphics processing capabilities and can input high-resolution cameras. While realizing high-speed AI calculations, its power consumption is only 5 to 10 watts, and its volume is also very small, only 69.6mm * 45mm, which is convenient to carry. On the development platform, we use SanDisk 64G MicroSD (write speed: 90MB / s, read speed: 170MB / s) to store the application system. we use the MIPI CSI-2 channel to connect the camera to the AI platform, (up to 1.5Gps Transmission speed can be achieved). The resolution of the camera is up to 3280 * 2464. Through the connection with the HDMI interface and the USB interface on the platform and a 4.3-inch capacitive LCD to achieve image display and touch interaction.

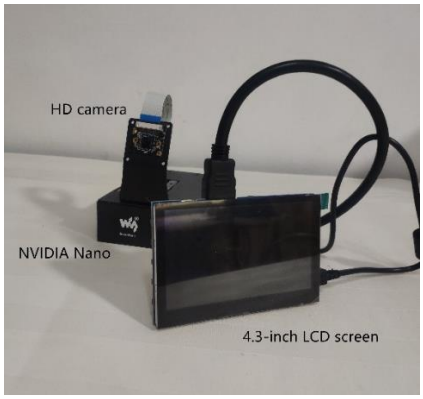


**Figure 4. Composition of the device**

## 3. Experiment

### 3.1 Image quality

StegaStamp is currently the only model that can still very robust work under different distortions. In the experiments, we compared the quality of the images generated by StegaStamp with the quality of the images generated by our model.

To evaluate the quality of images generated by encoder, We compare with StegaStamp[5] in SSIM[6] and PSNR. It shows that our image quality is better than the images generated by StegaStamp. This shows that it is effective to add JND as a loss to the neural network training process.

**Table 1. The quality of image reconstruction**

| Method | SSIM | PSNR |
|---|---|---|
| StegaStamp | 0.9233 | 27.24 |
| Proposed | **0.9362** | **28.60** |

### 3.2 Image position

It's important for our system to locate the image with hidden information efficiently. In this experiment, we randomly place the picture in the real environment, and use our positioning program to achieve the positioning of the marker. This picture shows how the image position system works in real environments.

Raw image

Polar binary image

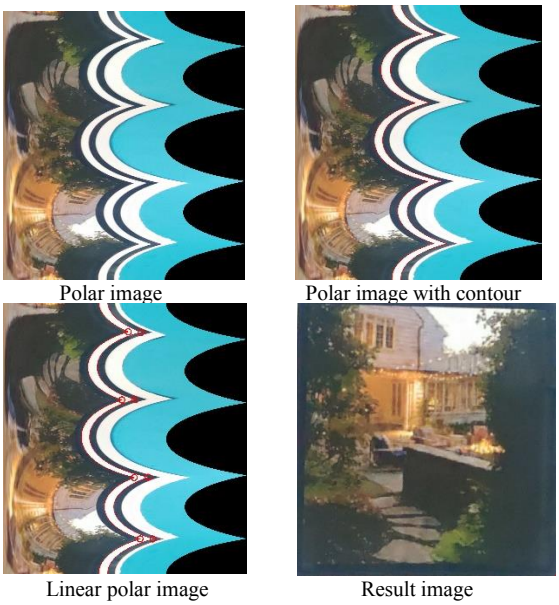


**Figure 5. Process of locating pictures**

### 3.3 Performance on AI chips

We use high-definition cameras on the AI platform to collect pictures and pass the pictures through our positioning program to locate the position of the picture with the recognition frame in the pictures with different backgrounds. After positioning, we perform cropping, perspective transformation operations, and input to the decoding network parsing the hidden information behind the picture. In this process, the camera information is input to the CPU through MIPI CSI-2 DPHY 1.1. The CPU is a quad-core ARM Cortex-A57 MPCare processor. A positioning program is also running on this CPU, and the output of the picture by the positioning program will be handed over to the GPU to complete the parsing information. The GPU is equipped with 128 NVIDIA CUDA cores and a processor based on the NVIDIA Maxwell architecture.

We evaluate the time consumption of recognizing a single image and the accuracy of model recognition on the AI chip. To evaluatethe the robustness of our model, we test our system in different scenes including indoor, outdoor, light and wrap. The test images were presented in paper and screen. We capture 20-50 photos under different spots. The speed result is 86ms per picture or 11.62Fps.

This experiment shows our model is quite robust, whether it is an indoor or outdoor environment, the angle is distorted, even under the condition of light, The recognition effect is still very well. At the same time based on hardware acceleration, the recognition speed is relatively ideal. In the future, with the continuous optimization of software and hardware, our recognition speed and accuracy will be further improved, and the application scenarios will be expanded.

**Table 2. Test results in real environments**

| Printed on Paper | | | |
|---|---|---|---|
| indoor | outdoor | warp | mean |
| 100% | 100% | 72.73% | 94.59% |
| Displayed on Screen | | | |
| indoor | warp | light | mean |
| 100% | 100% | 92.86% | 96.77% |

## 4. Conclusion

We designed a special locator to locate the position of the picture with hidden information in the field of vision and a removable device that accelerates in software and hardware to achieve real-time interfacing pictures. In the future, we will add more information to one picture. At the same time we will reduce the device's energy consumption and make it smaller, more imperceptible just like our glasses.

## 5. References

[1] Jun Jia, Zhongpai Gao, Kang Chen and Menghan Hu. Robust Invisible Hyperlinks in Physical Photographs Based on 3D Rendering Attacks. arXiv preprint arXiv: 1912.01224

[2] D. Volkhonskiy, I. Nazarov, B. Borisenko, and E. Burnaev. Steganographic generative adversarial networks. arXiv preprint arXiv:1703.05502, 2017

[3] J. Wu, L. Li, W. Dong, G. Shi, W. Lin and C. -. J. Kuo, "Enhanced Just Noticeable Difference Model for Images With Pattern Complexity," in IEEE Transactions on Image Processing, vol. 26, no. 6, pp. 2682-2693, June 2017.

[4] D. Parikh and G. Jancke, "Localization and Segmentation of A 2D High Capacity Color Barcode," 2008 IEEE Workshop on Applications of Computer Vision, Copper Mountain, CO, 2008, pp. 1-6.

[5] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp::Invisible Hyperlinks in Physical Photographs. arXiv preprint arXiv:1904.05343, 2019. 1, 2, 3, 5, 6, 7, 8

[6] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, April 2004.