Medical Key Area Protection

a Project Report

CSE4019 - Image Processing

by

DHUSHYANTH M K	**- 19BCE1230**
MANOGHN KANDIRAJU	**- 19BCE1621**

under the guidance of

Dr. Geetha S

VIT®

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

May 2022

TABLE OF CONTENTS

ABSTRACT

This project explores ways to reversibly hide data using pixel manipulation methods, and subsequently enable key area protection in the medical industry. Two main methods were explored in this project, both of which are centered around altering pixel values in order to secure important information that can be found on medical images such as x-rays. The first method involves hiding the diagnostic report in a medical scan, then protecting the scan using blurring or distortion algorithms. The second method deals with manually identifying the key area of a medical image, followed by securing the same by embedding a QR Code with diagnostic data, and manipulating pixel values to hide the key area in the embedded image.

INTRODUCTION

Medical images have become an indispensable and effective auxiliary means for modern medical diagnosis. Medical diagnostics like a patient's radiological scan are generally treated as susceptible information. Various methods to protect such information exist, both physically and digitally. Medical Key Area Protection using Image Processing is a powerful and reliable way to store medical diagnostics safely. Information Hiding and Image Encryption are useful techniques to implement the same.

Other integral techniques used in this project include a few pixel manipulation methods. A pixel is the smallest element of an image. Images are generally stored digitally as an array of pixels. Every pixel has a color and the way colors are defined is called the color space. The most used color space is the RGB color space. RGB is a system constituted by three values, one for red, one for green, and one for blue (R,G,B). The color depth of the RGB system is an 8-bit unsigned integer (a range of 0–255) for each part.

Another element of security added was through the means of Quick Response (QR) codes. A QR code is a type of matrix barcode that is machine-readable and optical. It contains information about the item or data to which it is attached, such as data for a locator, identifier, or tracker that points to a website or application. A QR code uses four standardized encoding modes (numeric, alphanumeric, byte/binary, and kanji) to store data efficiently. A QR code is essentially a white background on which black squares are arranged in a square grid, which can be read by an imaging device such as a camera and processed using Reed–Solomon error correction until the image can be appropriately interpreted. The required data is then extracted from patterns that are present in both horizontal and vertical components of the image.

This project makes use of the above mentioned attributes to help protect medical images and their key area in particular.

PROBLEM STATEMENT

To figure out how to effectively hide the key areas of a medical image such as X-Rays using Image Processing.

CHALLENGES

Due to the vast amount of content in datasets, it may be difficult to evaluate every column of the dataset. Even if every column has been evaluated, it is possible that a few columns have not been studied. Correlating the link between likes, dislikes, and view count does provide a relationship, but it is not particularly standard to apply to every video; the relationship changes from video to video, necessitating a reanalysis of the dataset. We represent the most popular genre through the analysis; yet, entering a popular genre has its drawbacks because the market is so densely occupied, which may entail higher risk as there is more competition for the content creator.

DATASET USED

Various MRIs and scans available online are used as the dataset for this project. Any random MRI or a CT scan at the user end can also be used for this project.

LITERATURE SURVEY

J. Li et al[1] presented in their paper an algorithm to protect key regions in medical images. They used the coefficient of variation in order to locate the key/lesion areas of a medical image while other areas are then processed in blocks and analyzed for texture complexity. A reversible data-hiding algorithm is used to embed the extracted contents from these lesion areas into a high-texture area, and Arnold transformation is performed to protect the information from the original lesion. Following up with a similar idea to this, Zelin Zhang et al.[2] used the ciphertext of the basic information about the image and the decryption parameter to generate a Quick Response (QR) code that will in turn replace the original key regions. This ensures that only authorized customers can obtain the encryption key to extract key area information from encrypted images. Other experimental results show that their algorithm can restore the original image without information loss as well as safely transfer the medical image copyright and patient-sensitive information.

In another view of this problem, Kong, Ping, et al.[3] suggested a reversible data hiding technique in order to protect medical DICOM images in particular. They claim that most other existing methods are unsuitable for medical DICOM images, as they do not make the best use of the DICOM image format's features. The recovery accuracy is also low because medical images have large areas with the same pixel values. In order to avoid these weaknesses, they proposed a scheme that segments the image and only embeds data into specific portions of the encrypted image. They exploited the redundancy of pixel cells in DICOM images so that the auxiliary data can be embedded into the image. In addition to this, the hash value of the minimum bounding rectangle of ROI and the feature bit matrix of the rest of the image are calculated to ensure the integrity of the DICOM image. This allows for convenient detection in the event that the image is tampered with and the tampered ROI area can be located on the receiver side.

K. Wang, et al.[4] presented a reversible data hiding (RDH) technique-prediction error histogram (PEH) shifting that is evaluated based on both Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) Index. It involves the two techniques of SSIM based block selection and PEH based optimal expansion bins selection. Block selection involves splitting of the original image into smooth and rough blocks where, smooth blocks are double embedded and rough ones are left unchanged. Optimal expansion bins technique is used to reduce the embedding distortion by selecting a few reasonable pixels for embedding. These new techniques yield higher SSIM and PSNR which exhibits its precedence over other PEH based methods.

The last method explored in this domain was presented by Jiankun Hu, et.al [5], who put forward a pixel based scrambling technique to distribute digital medical images in a safe and efficient manner. The suggested method leverages a basic pixel level XOR operation for image scrambling such that the structural features of the encryption scheme become a part of the cryptographic key. This allows efficient encryption of a huge number of digital medical images. A true random number sequence that is generated from multi-scroll chaotic attractors serves as the cryptographic key. A simulation experiment is used to corroborate the effectiveness of the system.

IMPLEMENTATION

Method 1:

Module 1 - Content Extraction

Extracting textual content from medical diagnoses and reports.

Module 2 - Pixel Manipulation

Extracted content is embedded within the brightest and darkest parts of an x-ray file. A key is generated which stores metadata on the above encrypted data which can be later used to retrieve the original file.

Module 3 - Image Distortion/Blurring

The processed image is then passed through an image distortion or image blurring algorithm in order to secure the file and render it unreadable to the human eye.

Module 4 - Image Restoration/Deblurring

The output file can be reverted by first passing it through an appropriate deblurring algorithm which would return the x-ray with embedded diagnostic text.

Module 5 - Restore original image

The key can be used to extract the embedded text data which would result in the original x-ray file and diagnostic text being outputted separately.

Method 2:

Module 1 - Content Extraction

Extracting textual content from medical diagnoses or reports.

Module 2 - QR Generation

A QR is generated using the extracted textual information from the medical diagnostic file.

Module 3 - Key Area Protection

The generated QR is overlaid on top of the key area of the medical image so as to hide the sensitive information from the human eye.

Module 4 - Pixel Manipulation

Pixel manipulation is used to embed the extracted key area back within the image in the brightest and darkest parts of the x-ray. This results in the distortion of the QR code as well which prevents unauthorized access of the diagnostic data. A key is generated that stores metadata on the above encrypted data which can be later used to retrieve the original file.

Module 5 - Decryption

Use the key to retrieve the key area data from the processed image, after which the diagnostic text can be extracted by reading the embedded QR code.

Module 6 - Restore original image

The retrieved key area can be embedded back into the image over the same area that the QR code is placed in order to restore the original image.

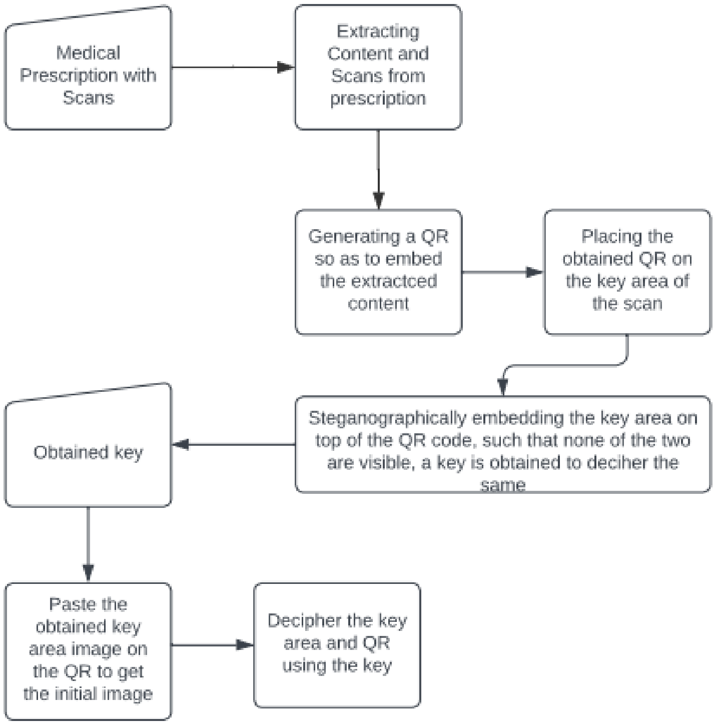

Medical Prescription with Scans
Extracting Content and Scans from prescription
Generating a QR so as to embed the extractced content
Placing the obtained QR on the key area of the scan
Steganographically embedding the key area on top of the QR code, such that none of the two are visible, a key is obtained to deciher the same
Obtained key
Paste the obtained key area image on the QR to get the initial image
Decipher the key area and QR using the key

Fig. 1. Architecture diagram

CONCLUSION AND FUTURE WORK

Through this project, we have understood how integral data protection could be in the medical field. We have discovered and explored various methods to protect the medical images without compressing them or losing any minor information from them. Employing QRs to embed the textual data and steganographically hiding the QR using the key area in the medical image helps us ease the process by limiting our scope. We have manually selected the key area in the medical images, but in future work, we would like to explore automating this process by using various machine learning methods where we could train the model to detect the key areas in the scans or MRIs by using a huge dataset.

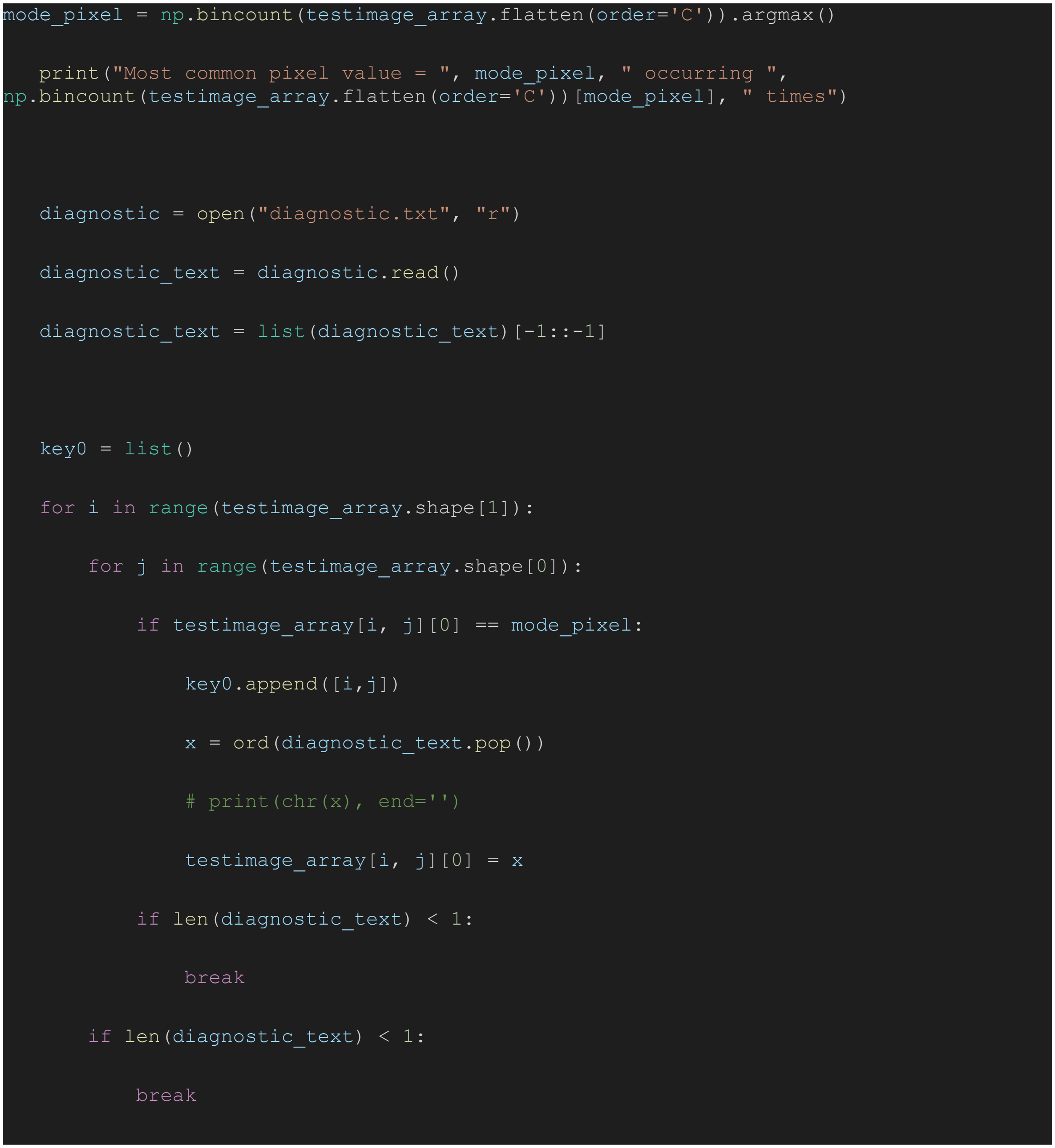
REFERENCES

1. Li, Jian, et al. "Reversible data hiding based key region protection method in medical images." 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2019.
2. Zhang, Zelin, et al. "Medical Image Key Area Protection Scheme Based on QR Code and Reversible Data Hiding." Security and Communication Networks 2021 (2021).
3. Kong, Ping, et al. "Reversible data hiding in encrypted medical DICOM image." Multimedia Systems 27.3 (2021): 303-315.
4. Wang, Kehao, et al. "Reversible data hiding based on structural similarity block selection." IEEE Access 8 (2020): 20375-20385.
5. Hu, Jiankun, and Fengling Han. "A pixel-based scrambling scheme for digital medical images protection." Journal of Network and Computer Applications 32.4 (2009): 788-794.

APPENDIX I

SAMPLE CODE

Replacing most commonly occurring pixel value with ASCII value of text data



```
mode_pixel = np.bincount(testimage_array.flatten(order='C')).argmax()

   print("Most common pixel value = ", mode_pixel, " occurring ",
np.bincount(testimage_array.flatten(order='C'))[mode_pixel], " times")



   diagnostic = open("diagnostic.txt", "r")

   diagnostic_text = diagnostic.read()

   diagnostic_text = list(diagnostic_text)[-1::-1]



   key0 = list()

   for i in range(testimage_array.shape[1]):

       for j in range(testimage_array.shape[0]):

           if testimage_array[i, j][0] == mode_pixel:

               key0.append([i,j])

               x = ord(diagnostic_text.pop())

               # print(chr(x), end='')

               testimage_array[i, j][0] = x

           if len(diagnostic_text) < 1:

               break

       if len(diagnostic_text) < 1:

           break
```

SCREENSHOTS

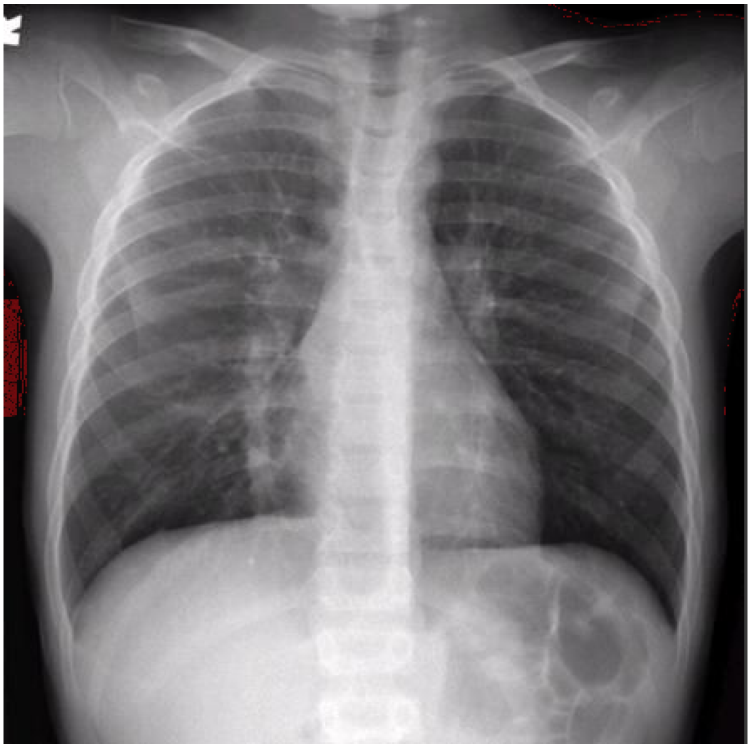


Fig. 2. X-ray embedded with text

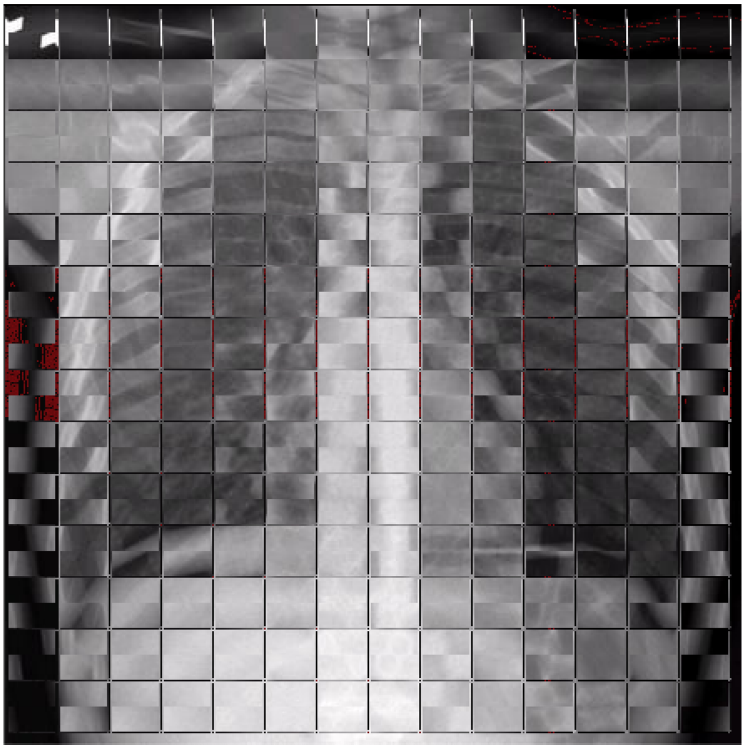


Fig. 3. Distorted X-ray

APPENDIX II

SAMPLE CODE

```
med_img2 = cv2.imread(med_img_file)

    ROI = cv2.selectROI("Select Rois",med_img2)

    print(ROI)



    key_area_loc = [ROI[0] + ROI[2]/2, ROI[1] + ROI[3]/2]

    key_area_loc = [int(key_area_loc[0]-img_qr.size[0]/2),
int(key_area_loc[1]-img_qr.size[1]/2)]

    key_area_loc[0] = 0 if key_area_loc[0] < 0 else key_area_loc[0]

    key_area_loc[0] = med_img.size[0]-img_qr.size[0] if key_area_loc[0] >
med_img.size[0]-img_qr.size[0] else key_area_loc[0]

    key_area_loc[1] = 0 if key_area_loc[1] < 0 else key_area_loc[1]

    key_area_loc[1] = med_img.size[1]-img_qr.size[1] if key_area_loc[1] >
med_img.size[1]-img_qr.size[1] else key_area_loc[1]

    print(key_area_loc)



    key_area = med_img_array[key_area_loc[1]:key_area_loc[1]+img_qr.size[1],
key_area_loc[0]:key_area_loc[0]+img_qr.size[0]]

    Image.fromarray(key_area).show(title="Key Area")

    key_area_str = numpy_to_bytes(key_area)

    med_img.paste(img_qr, key_area_loc)

    med_img_array = np.asarray(med_img)

    Image.fromarray(med_img_array).show()

    key_area_enc = list(str(key_area_str))[-1::-1]

    key = list()
```

```
for mode_pixel in mode_pixels:
    key0 = list()
    for i in range(med_img_array.shape[0]):
        for j in range(med_img_array.shape[1]):

            if len(key_area_enc) > 0 and med_img_array[i, j][0] == mode_pixel:
                key0.append([i,j,0])
                x = ord(key_area_enc.pop())
                med_img_array[i, j][0] = x

            if len(key_area_enc) > 0 and med_img_array[i, j][1] == mode_pixel:
                key0.append([i,j,1])
                x = ord(key_area_enc.pop())
                med_img_array[i, j][1] = x

            if len(key_area_enc) > 0 and med_img_array[i, j][2] == mode_pixel:
                key0.append([i,j,2])
                x = ord(key_area_enc.pop())
                med_img_array[i, j][2] = x

            if len(key_area_enc) < 1:
                break

        if len(key_area_enc) < 1:
            break
    key.append(key0)

    if len(key_area_enc) < 1:

        break

key.append(mode_pixels)

key.append(key_area_loc)

key = np.array(key, dtype=object)

np.save("key.npy", key)
```

SCREENSHOTS

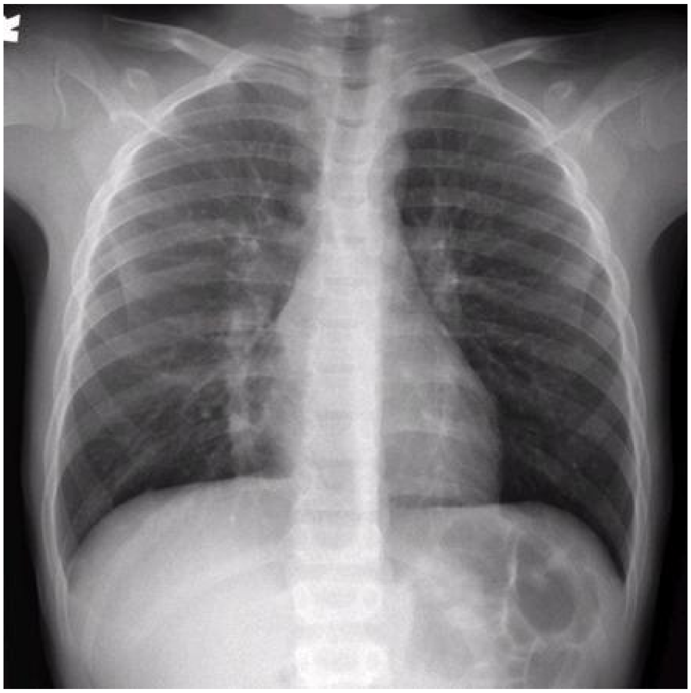


Fig. 4. Original X-ray file

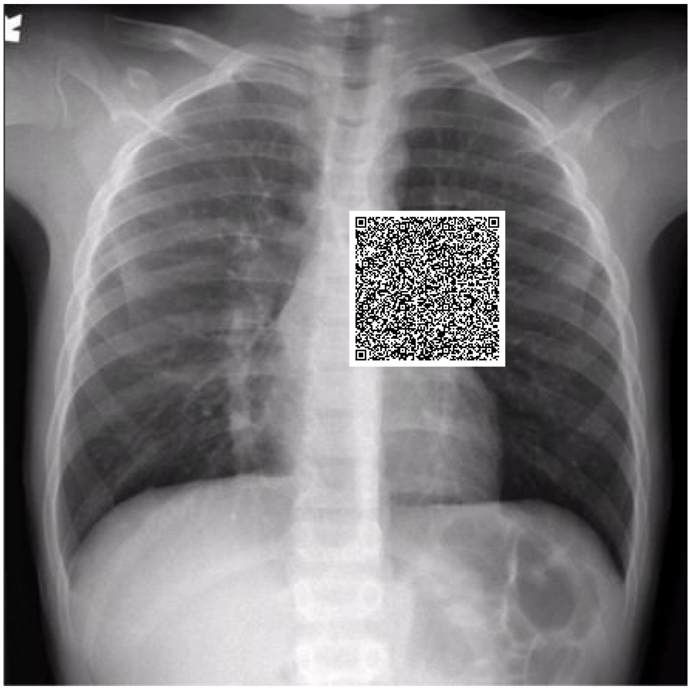


Fig. 5. QR code hiding the key area

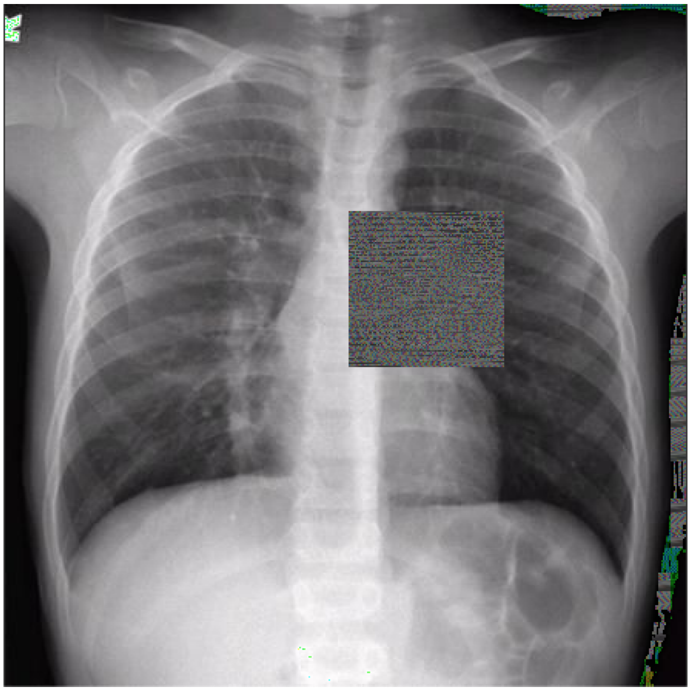


Fig. 6. Key area embedded back into the image