# DataMatrix Barcode Read Rate Improvement Using Image Enhancement

Vladislavs Svarnovics Department of Computer Vision and Biometrics University of Twente v.svarnovics@student.utwente.nl

Abstract—Almost every product in a supply chain comes with a barcode that can be decoded using special decoding devices. Barcodes are usually classified into one and two-dimensional. One-dimensional barcodes are often used for retail product labeling while two-dimensional barcodes are commonly used for manufacturing, warehousing, and logistics. DataMatrix is a popular type of 2D barcodes that, in the context of this work, was used for post parcel and envelope labeling. It is not always possible to successfully extract the information present in a DataMatrix barcode since the decoding may suffer from various image distortion types including blur, smudge, and deformation. In order to improve the decoding rate, classical binarization methods and modern deep learning enhancement solutions were investigated. More specifically, Otsu, Sauvola, Niblack, and Nick binarizations methods were contrasted against state-of-the-art Unet architectures such as AttUnet and Unet3+. The main research question of this work was to find out to what extent Unet-based architectures outperform binarization methods in terms of the DataMatrix decoding rate. In this paper, 65237 decodable DataMatrix barcode samples were analyzed, where 56580 samples were decoded using the open-source ZXing library. The decoded barcodes served as a training set for the deep learning methods since every decoded barcode could be reconstructed into a distortion-free reference image. The results have revealed that the investigated deep learning methods led to 74% decoding rate improvement, and clearly outperformed binarization methods which achieved a 24% decoding rate on the same test set.

#### I. INTRODUCTION

DataMatrix barcodes are extensively used for automatic identification and data capture [1]. The food industry, logistics, and many other commercial products utilize DataMatrix codes as it allows to store a significant amount of information occupied by a small area. DataMatrix barcodes can be decoded by a scanning device which is substantially faster than entering information into a system by hand. Moreover, it results in fewer errors because decoding machines are more reliable in this task [1].

While DataMatrix barcodes generally offer very good performance, from time to time decoding issues do take place. The most common causes of unreadable barcodes include low contrast, poor printing quality, barcode damage, and various types of distortion [2].

This work explores and compares several enhancement methods that are primarily addressed to improve the DataMatrix barcode decoding rate. Traditionally, binarization methods were used to solve some of the decoding problems, however, in this work, it was attempted to apply deep learning solutions aiming to reconstruct the ground truth barcode image and potentially make a barcode image easily decodable.

A convolutional autoencoder architecture was found suitable for many image denoising, deblurring and resolution enhancement applications [3, 4, 5]. The encoder-decoder structure allows to learn the underlying data patterns, and reconstruct the noiseless output image.

Unet architecture is a more advanced version of the convolutional autoencoder that was initially developed for biomedical semantic segmentation. This paper is mainly focused on the possibilities of barcode enhancement using Unet-based architectures. Subsequently, the main research question of this paper is: "To what extent do Unet-based DataMatrix enhancement solutions outperform classical binarization methods in terms of the decoding rate?". This paper has the following sub-research questions:

- 1) To what extent do Unet-based solutions outperform traditional convolutional autoencoders in terms of the Data-Matrix decoding rate?
- 2) What Unet architecture is best-suited for DataMatrix barcode enhancement?
- 3) What binarization method is best-suited for DataMatrix barcode thresholding?

This work is structured as follows: Section II provides a short overview of DataMatrix barcodes. It also describes the existing barcode decoding tool that was used to conduct the experiments described in this paper. Section III discusses the related works that attempted enhancing 2D barcode images. Section IV provides a detailed overview of the DataMatrix dataset, several binarization methods as well as different Unet architectures. The results of this work are described in Section V. Discussion and conclusion can be found in Sections VI and VII respectively.

#### II. BACKGROUND

# A. DataMatrix

DataMatrix is composed of a sequence of black and white modules (cells) that can be arranged in square or rectangular shapes, and may appear in different sizes.

The size of a DataMatrix code varies from  $10 \times 10$  to  $144 \times 144$  modules [6]. The newest ECC (Error Checking and Correcting) 200 standard allows for the  $10 \times 10$  DataMatrix



Fig. 1: Small DataMatrix Examples



Fig. 2: Large DataMatrix Examples



Fig. 3: DataMatrix Alignment Pattern Examples

version to have a capacity of 3 bytes which is equivalent to six numeric or three alphanumeric symbols while the  $144 \times 144$  DataMatrix can store 1558 bytes corresponding to 3116 numeric or 2335 alphanumeric symbols.

The DataMatrix alignment pattern (see Figure 3) is a solid line of contiguous dark cells abutting a line of alternating dark and light cells. The alignment patterns run horizontally and vertically within the symbols [6]. The alignment pattern line splits a barcode into several data regions. The larger the barcode, the more data regions are present. Small barcodes (see Figure 1) have only one data region while larger barcodes (see Figure 2) have four or more data regions.

DataMatrix codes are designed in such a way that even if a part of the image is severely damaged, the barcode can still be decoded. The ECC 200 standard of DataMatrix uses Reed–Solomon codes for error and erasure recovery. It allows for up to 30% damage while still preserving a decodable barcode [7].

This work is mostly focused on square-shaped barcodes with the number of cells varying from  $18 \times 18$  to  $36 \times 36$  and containing at most four data regions.

# B. ZXing decoder

Decoding a DataMatrix is the process of converting barcode image pixels into an array of bytes that represent the data encoded in the barcode. The first step in decoding a Data-Matrix is to split the data regions into L-shaped blocks (see Figure 4). If the L-shape does not fully fit, then part of it is placed on the other side of the DataMatrix. The cells in every L-shaped block (except for error correction blocks) are numbered according to the powers of two, starting from 128 and ending up with one. For example, in order to obtain an ASCII value of an encoded symbol (not necessarily the first symbol), consider the top left L-shape in Figure 4 (blue color). Numbers 64, 4, and 1 are highlighted since they correspond to black cells of the DataMatrix code. The ASCII value is then a summation of the numbers in the highlighted cells minus one. For this case, this would result in (64 + 4 + 1) - 1 = 68, where 68 corresponds to the character *D*. Note that the *minus one* operation corresponds specifically to ASCII encoding.



Fig. 4: DataMatrix Decoding Pattern Example

Zxing is a third-party open-source library developed by Owen et al. [8], that allows to both detect and decode barcodes and retrieve the data encoded regardless of its rotation angle. ZXing library supports several 1D barcodes (e.g. Code 39, Cobabar, ITF) as well as 2D barcodes (e.g. QR-code, DataMatrix, Aztec, MaxiCode). ZXing is one of the most popular and recognized open-source barcode processing libraries. Table I summarizes the rating of ZXing against other open-source libraries. This rating represents the public recognition of the library. Generally, this rating is composed based on the number of library features and their performance. Despite the fact that amongst open-source barcode decoder projects ZXing is the most successful one, it still does not support complex scanning conditions such as nonuniform illumination, bend, and deformation [9]. For the barcode detection part (i.e. finding barcode region), ZXing uses its own local binarizer call HybridBinarizer, however, after the barcode is located, no further image preprocessing is applied.

# III. RELATED WORK

# A. Classical Methods

The work by Brouwer [13] attempted to improve DataMatrix read rates by using image pre-processing. It involved morphological erosion, region growing for object segmentation, edge analysis, and Fisher's linear discriminant as means for element classification. The entire dataset consisted of 1020 thick dot DataMatrix images, 510 of which were used for training and the remaining 510 exclusively for testing. 2DTG, AYPSYS, and ClearImage were used as software decoding packages. The results demonstrated improved read rates in most test cases. More specifically, the read rate accuracy on the test set was improved from 99.4% to 99.8% for the AYPSYS decoder, and from 90.6% to 98.0% for the ClearImage decoder. It was observed that the read rate accuracy for the 2DTG decoder was not improved after image pre-processing.

TABLE I: Open-source Barcode Decoders Rating Comparison: the rating is based on library public recognition by its users

	ZXing	Efqrcode	Zxinglite	BarcodeScanner
Rating source 1 [10]	28124	4015	1933	-
Rating source 2 [11]	26379	3800	1503	-
Rating source 3 [12]	28108	4015	-	4015

The paper by Ottaviani et al. [14] proposed the entire 2D barcode (incl. DataMatrix) processing framework that involved region of interest detection, barcode pre-processing, and decoding. The pre-processing step included the modified version of the Niblack binarization algorithm to deal with variable lighting conditions. The barcode pixel segmentation was performed by an alternate sequence of region growing and convex hull evaluation. The suggested approach led to high reading performances even under bad lighting conditions and strong perspective deformations.

The paper published by Li et al. [15] aimed to apply a multifeature fusion algorithm to improve QR-code recognition rates. The algorithm incorporated color and texture feature extraction that were further used to classify pixel points using k-means clustering. Final QR-code image optimization was done using mathematical morphology. The proposed method evaluation was performed on 400 QR-code images samples each of size  $653 \times 673$ . Data samples represented QR-codes that were laser ablated on the rough surfaces of aluminum ingots. The proposed methods appeared superior compared to the accepted Otsu binarization method. The read rate after Otsu thresholding was 50%, while the proposed method led to a read rate of nearly 100%.

The work by Chen et al. [16] tackled the problem of QR-code uneven illumination in warehouse automatic sorting systems. The proposed method selected the size of the block window adaptively and then used this window to divide the uneven illumination of the QR code image into several blocks. Afterward, it performed the thresholding on each image block and then combined them in sequence. The experiments were conducted on 80 QR-code samples of size 300×300, where 30 contained weak uneven illumination and the remaining 50 - strong uneven illumination. The suggested approach outperformed other binarization methods such as Niblack, Yao, Di, and Otsu in terms of PNSR and SSIM metrics. Moreover, it surpassed the previously mentioned binarization methods in terms ZXing decoding rate which was boosted from 35% (no binarization) to 88.75% (proposed method). The read rate for the other methods did not exceed 54%.

#### B. Deep Learning

Huo et al. [17] strived to improve QR-code read rate which was negatively influenced by uneven background fluctuations, inadequate illuminations, and geometrical deformations. These distortions were the result of the improper image acquisition method. To increase the read rate, an improved adaptive median filter algorithm was used, followed by the Otsu binarization method. Geometrical deformations and perspective correction were performed by a feed-forward neural network that was trained using ground truth QR-code samples. A total of 300 QR-code samples were acquired by a smartphone camera. The QR-code samples were printed on objects that were prone to wrinkles. 209 samples were decoded by a smartphone barcode scanner. The decoded barcodes were reconstructed into ground truth samples which were used for the neural network training. The read rate on the entire dataset before and after image pre-processing was 69.7% and 83.7% respectively.

# IV. METHODOLOGY

# A. Overview

Figure 5 shows the pipeline of DataMatrix processing. The pipeline involves two steps. The first step is image preprocessing that can be achieved either by binarization or deep learning enhancement. The second step is image decoding using the ZXing library that determines whether the decoding was successful. Image binarization and deep learning enhancement are processed independently from each other.



Fig. 5: DataMatrix Processing Pipiline

#### B. Dataset

For the experimental scope of this work, PrimeVision B.V. [18] provided access to a private DataMatrix dataset. The entire DataMatrix barcode dataset consisted of 65237 unrotated greyscale barcodes that were acquired from various post parcels and envelopes using a very high-resolution camera. All the DataMatrix images were proved to be decodable using a *SwiftDecoder* - commercial state-of-the-art software package developed by Honeywell [19]. The dataset images included barcodes with various types of distortions such as geometrical transformations, partial occlusion, low contrast, low ink, smudge, blur, white noise, and ink merging. Some of the barcode examples are depicted in Figure 6.

The majority of the barcodes (> 99.9%) had a square shape, the remaining ones - rectangular.

The average image size was  $188 \times 188$  pixels while the largest and the smallest -  $403 \times 401$  and  $66 \times 65$  pixels respectively. For neural network training purposes, all images were resized to  $120 \times 120$  pixels. For the binarization methods, the size of all images remained unchanged.



Fig. 6: Dataset Example Samples

1) Training Data: In order to train a neural network, a ground truth image needs to be generated. In fact, the ground truth sample is merely a distortion-free, black and white image that preserves exactly the same barcode content. To obtain the ground truth image, the original barcode has to be decoded first, and the information, encoded in bytes, needs to be extracted. Once the barcode information bytes are extracted, the *clean* version of the barcode can be deterministically reconstructed. Ground truth generation examples are shown in Figure 7.



Fig. 7: Ground Truth Example: original samples (top row), ground truth samples (bottom row)

Not every dataset sample can be decoded due to various distortion problems. A total of 56580 samples out of 65237 were decoded using the ZXing library.

2) Test Data: The remaining 8657 samples form the test set that contains barcodes that are impossible to decode using the ZXing library without any pre-processing (enhancement). The amount of successful barcode decodes on the test set determines the performance of neural network-based and binarization enhancement methods.

3) Data Augmentation: Data augmentation can prevent a neural network from learning irrelevant patterns, essentially boosting overall performance [20]. In this work, every barcode in the training set was rotated three times by 90 degrees, thus increasing the training set size by a factor of four (226320 total training samples). This approach increases the variability in the dataset and helps reduce overfitting. One limitation of data augmentation arises from the data bias. The augmented data distribution can be quite different from the original one. This might lead to suboptimal performance of existing data augmentation methods [21].

#### C. Binarization

Binarization is the method of converting a grayscale image into a black-white image depending on a certain threshold. If the gray value of the pixels is less or equal to the threshold, then those pixels become black, and similarly, if the gray value of the pixels is larger than the threshold, then those pixels become white.

Binarization is an essential preprocessing step for many OCR (Optical Character Recognition) solutions [22]. Regarding barcode binarization, the main objective of the pre-processing phase is to make it as easy as possible for the decoding system (e.g. ZXing) to distinguish a barcode pixel from the background.

1) Global Binarization: Global binarization methods use information from the whole image to find one threshold which will make a binary image. One of the most popular global binarizations is the Otsu method that utilizes the grayscale histogram to find the best separation result [23]. Otsu algorithm exhaustively searches for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes [24]. The mathematical definition of the intra-class variance is shown in (1)

$$\sigma_w^2(t) = w_0(t)\sigma_0^2(t) + w_1(t)\sigma_1^2(t) \tag{1}$$

where weights  $w_0$  and  $w_1$  are the probabilities of the two classes separated by a threshold t and  $\sigma_0^2$  and  $\sigma_1^2$  are the variances of these two classes. The class probability  $w_{0,1}(t)$  is computed from the n bins of histogram of gray-scale values.

$$w_0(t) = \sum_{i=0}^{t-1} p(i)$$
$$w_1(t) = \sum_{i=t}^{n-1} p(i)$$

where p is the histogram value at bin i. The optimal number of bins n can be derived from the Freedman-Diaconis rule [25] which states that the optimal bin width h can be found as shown in (2)

$$h = 2\frac{IQR}{k^{\frac{1}{3}}} \tag{2}$$

where k is the number of observations and IQR is the interquartile range. The number of bins n can then be computed as shown in (3)

$$n = \frac{\max - \min}{h} \tag{3}$$

where  $\max$  and  $\min$  are the largest and the smallest data points in the sample.

The Otsu approach is simple, fast, and effective for many thresholding applications, however, its performance may suffer from nonuniform image lighting.

2) Local Binarization: Local binarization techniques determine a different threshold value for every pixel based on characteristics of their surrounding area [26]. Local binarization algorithms calculate a pixel-wise threshold by sliding a rectangular window w over the image [27]. It was discovered that for many OCR applications as well as for degraded document restoration, local thresholding algorithms have superior performance to the global Otsu thresholding algorithm [28, 29].

Several local binarization methods, that are commonly used in OCR applications [30, 31, 32], were considered in this work.

a) Niblack Algorithm: The computation of threshold T for the Niblack method is based on the local mean m and the standard deviation  $s = \sqrt{\frac{1}{N}\sum(p_i - m)^2}$  of all the pixels in the window w and is given by the (4)

$$T_{Niblack} = m + k \cdot s \tag{4}$$

where N is the number of pixels, m is the average pixel value of the pixels  $p_i$ , and k is the fade control factor [33].

*b) Sauvola Algorithm:* Sauvola algorithm computes the threshold by using the dynamic range of image gray-value standard deviation [27]. Sauvola threshold value can be computed as described in (5)

$$T_{Sauvola} = m(1 - k(1 - \frac{s}{R})) \tag{5}$$

where constant R is set to 128. Sauvola method outperforms Niblack algorithm in images where the pixels have near 0 gray-value and the background pixels have near 255 gray-values. [27]. Sauvola binarization results substantially degrade when background and foreground pixels values are close to each other.

c) Wolf Algorithm: To solve the issues in Sauvola's method, the Wolf algorithm aims to normalize the contrast and the mean gray value of the image and compute the threshold as described in (6) [27]

$$T_{Wolf} = (1 - k)m + kM + k\frac{s}{R}(m - M)$$
(6)

where the value of R is set to the maximum gray-value standard deviation obtained over all the local neighborhoods and the value M is the minimum gray value of the image.

The disadvantage of the Wolf method is that it cannot handle sharp changes in background gray values across the image. This occurs because values M and R are computed based on the entire image.

*d)* Nick Algorithm: Nick binarization algorithm is derived from the Niblack method. Its main advantage is that it substantially improves binarization for light images by shifting down the binarization threshold [27]. The Nick threshold value is depicted in (7)

$$T = m + k\sqrt{\frac{\left(\sum p_i^2 - m^2\right)}{N}} \tag{7}$$

Only for the Nick algorithm, the control parameter k is negative and normally varies between -0.2 and -0.1 [27]. For the remaining binarization algorithms described in this paper, the value of k is positive.

Local binarization algorithms require a user-defined sliding window size w and the fade control parameter k while global binarization methods (e.g. Otsu) can binarize a sample without any additional parameters.

### D. Deep Learning

1) Convolutional Autoencoder: Convolutional autoencoders (see Figure 8) are widely used for many image processing tasks including image denoising [34, 35]. Denoising autoencoders are an important and crucial tool for feature selection [36]. Works by Mao et al. [4] and Bigdeli and Zwicker [5] have successfully applied a convolutional autoencoder architecture to solve image denoising and non-blind deblurring problems. Regarding the barcode enhancement, we attempt to apply the image denoising/deblurring approach to restore the *clean* version of the barcode.



Fig. 8: Convolutional Autoencoder Architecture [37]

Taking into account that the paper by Huo et al. [17] demonstrated that it was possible to correct QR-code geometrical deformations using a BP neural network, it is assumed that DataMatrix geometrical distortions can be corrected using a convolutional autoencoder.

2) Unet: Unet structure (see Figure 9) is similar to the structure of a convolutional autoencoder, however, unlike the autoencoder, Unet supplements plain long skip connections (going from encoder path scale to the same scale at decoder path) that enable high resolution features to be combined in the output layer [38] which results in successfully recovering fine-grained image details [39]. Unet networks showed excellent

performance in various tasks, including image segmentation and denoising [40].



Fig. 9: Unet Architecture [40]

3) Attention Unet: Attention (see Figure 10), in the context of image segmentation, is a way to highlight only the relevant activations during training. This reduces the computational cost and provides the network with better generalization power [41, 42]. There are two types of attention: hard and soft. Hard attention highlights relevant regions by cropping the image or iterative region proposal. Soft attention assigns different weights to different parts of the image. Areas of high relevance are multiplied with larger weights while areas of low relevance are marked with smaller weights. The attention concept can complement the Unet model by implementing soft attention at the plain long skip connections. This actively suppresses activations in irrelevant regions and reduces the number of redundant features being passed on.



Fig. 10: Attention Unet Architecture [42]

4) Unet 3+: Unet3+ architecture (see Figure 11) takes advantage of full-scale skip connections [43] and deep supervisions. The full-scale skip connections incorporate lowlevel details with high-level semantics from feature maps in different scales while the deep supervision learns hierarchical representations from the full-scale aggregated feature maps [44]. The proposed approach maximizes the use of feature maps in full scales which results in accurate segmentation and efficient network architecture with fewer parameters. Unet3+ outperformed both Unet and Attention Unet in biomedical image segmentation.



Fig. 11: Unet3+ Architecture [44]

Table II shows a short summary of the neural network architectures used in this work.

TABLE II: Unet Models Summary

Madal	Number of Features	Number	
Niouei	Maps per Layer	of Parameters	
CAE	[64, 128, 256]	1329153	
Unet	[64, 128, 256]	2066497	
Attention Unet	[64, 128, 256, 512]	7911460	
Unet3+	[64, 128, 256, 512]	7639297	

All the described networks had symmetric encoder-decoder architecture and were trained for 40 epochs with a batch size of 64 using the Adam optimizer, the MSE loss, and the sigmoid activation function in the final layer. During the neural network training, 10% of the training data (i.e. 22632 out of 226320 samples) was used for validation and the remaining 90% for training (i.e. 203688 out of 226320 samples).

#### E. Similarity Metrics

Neural network barcode reconstruction quality can be evaluated using similarity metrics. The predicted image is compared to the ground truth sample. The similarity metrics serve as an intermediate enhancement quality assessment that is decoder (e.g. ZXing) independent.

1) MSE: The MSE (Mean Squared Error) is the most traditional difference estimator [45] that is used to compare how far away the ground truth image's pixels are from the predicted image's pixels. The mean of each pixel's difference is taken and then squared [46]:

$$MSE = \frac{1}{NM} \sum_{i=1}^{M} \sum_{j=1}^{N} (I_{pr}(i,j) - I_{gr}(i,j))^2$$
(8)

where values N and M are the image dimensions,  $I_{pr}$  and  $I_{gr}$  are the predicted and the ground truth gray-scale images respectively. The dissimilarity between the two images is directly proportional to the *mse* value.

2) SSIM: The SSIM (Structural Similarity) that was first introduced by Wang et al. [47], is correlated with the quality and perception of the human visual system [45]. The SSIM models image distortion as a combination of three factors that are luminance l, contrast c, and structure s.

Given two gray-scale images  $I_{pr}$  and  $I_{gr}$  the structural similarity is defined as follows

$$SSIM = l(I_{pr}, I_{gr})^{\alpha} \cdot c(I_{pr}, I_{gr})^{\beta} \cdot s(I_{pr}, I_{gr})^{\gamma}$$
(9)

where

$$l(I_{pr}, I_{gr}) = \frac{2\mu_{I_{pr}}\mu_{I_{gr}} + c_1}{\mu_{I_{pr}}^2 + \mu_{I_{gr}}^2 + c_1}$$
$$c(I_{pr}, I_{gr}) = \frac{2\sigma_{I_{pr}}\sigma_{I_{gr}} + c_2}{\sigma_{I_{pr}}^2 + \sigma_{I_{gr}}^2 + c_2}$$
$$s(I_{pr}, I_{gr}) = \frac{\sigma_{I_{pr}I_{gr}} + c_3}{\sigma_{I_{pr}}\sigma_{I_{ar}} + c_3}$$

The greater the *ssim* value, the greater the similarity between images.

Parameters  $\mu_{I_{pr}}$  and  $\mu_{I_{gr}}$  are the means of the predicted and the ground truth image while  $\sigma_{I_{pr}}$  and  $\sigma_{I_{gr}}$  are their respective standard deviations. The value  $\sigma_{I_{pr}I_{gr}}$  denotes the covariance of the two images. Constants  $c_1$ ,  $c_2$  and  $c_3$  are meant to maintain the stability of the comparison functions [16]. In the original specification for SSIM, the weights  $\alpha$ ,  $\beta$ and  $\gamma$  are set to 1 [48].

#### V. RESULTS

#### A. Binarization

The effect of different binarization methods on the read rate was evaluated on the DataMatrix test set consisting of 8657 samples. Table III shows the read rate percentage on the test set after the binarization. The best two read rates were highlighted and accompanied with a two (sample) standard deviation margin  $2\sigma$ . The standard deviation was calculated on eight non-overlapping subsets (obtained based on a random split), each consisting of 1082 images (except for one consisting of 1083 images). This process was repeated three times, and in the end, the average (out of three) standard deviation value was presented.

For the local binarization methods, several combinations of the sliding square window of the size w and the fade control factor k were used.

The time (in milliseconds) it takes to binarize the entire DataMatrix dataset is shown in Table IV. Nine measurements corresponding to nine different parameter combinations (same as in Table III) were performed. The timing results were obtained using a Linux computer running on a dual-core 1.20GHz CPU.

The visual binarization results are presented in Figure 12. The checkmark ( $\checkmark$ ) on top of the barcode image indicates that the barcode was decoded after binarization, while the cross

mark ( $\boldsymbol{X}$ ) shows that the barcode was not decoded after binarization. The first column of the images in Figure 12 represents the original images from the test set, the second, third, fourth, fifth, and sixth columns correspond to Otsu, Niblack, Sauvola, Wolf, and Nick binarization results respectively. For the local binarization methods (i.e. Niblack, Sauvola, Wolf, Nick) the first two rows correspond to (w = 15, |k| = 0.05), the two middle rows to (w = 17, |k| = 0.1), and the last two rows to (w = 19, |k| = 0.2).

#### B. Deep Learning

The neural network training and validation losses are depicted in Figure 13.

Neural network enhancement results are separated into two types: type 1 (see Figure 14) shows successful enhancement which was determined by the visual barcode quality and the successful decoding. Type 2 (see Figure 15) corresponds to a poor enhancement that did not result in visually correct DataMatrix barcodes which thereupon were not decoded.

The results of the deep learning experiments are summarized in Table V. The first two columns show the average MSE and SSIM metrics evaluated on the training set (226320 samples). The third and the fourth columns represent the read rate on training (226320 samples) and test (8657 samples) sets respectively. Similar to the binarization results (see subsection V-A), the best two read rates on the test sets are highlighted and presented with the double standard deviation.

#### VI. DISCUSSION

#### A. Binarization

Applying binarization as means of DataMatrix preprocessing showed an improved read rate on the test set (see Table III). While the Niblack method had the worst performance, the Nick method corresponded to the best read rate (i.e. 24%) with w = 19 and k = -0.1. The Otsu algorithm with a read rate of 23.35% came very close to the result obtained by the Nick approach.

The visual barcode binarization quality is presented in Figure 12. Regarding the first row, all the samples were decoded after binarization. This can be explained by a relatively low level of distortion in the original sample. In fact, all the binarized images look very similar to each other. The Otsu image, however, appeared to be thicker than the rest, and the Niblack image had surrounding noise around the barcode. In the second row, not a single binarized image was decoded. Only the Otsu image appeared quite clean, while the other binarizers resulted in an image with the surrounding noise which can be explained by a small window size w and strongly uneven background color. It can be suggested that the Otsu image was not decoded due to a small barcode size. Regarding the third row, only the Otsu and Nick images were decoded, where only the Otsu image did not have any surrounding noise. In fact, it cannot be fully determined why the Nick image was decoded since it appeared quite similar to Niblack, Sauvola, and Wolf images. It can be argued that the Nick image had slightly less surrounding noise around the barcode which was the deciding

TABLE III: Binarization Read Rate Results: best two results are highlighted and provided with two standard deviation margin

	Parameters (w, k )								
Binarization method	(15,0.05)	(15,0.1)	(15,0.2)	(17,0.05)	(17,0.1)	(17,0.2)	(19,0.05)	(19,0.1)	(19,0.2)
Otsu	23.35% ± 0.026%								
Niblack	10.45%	8.65%	4.14%	12.75%	10.72%	6.98%	14.05%	12.27%	9.22%
Sauvola	20.05%	21.08%	16.73%	20.37%	22.04%	18.69%	20.28%	22.15%	20.52%
Wolf	17.68%	19.71%	21.83%	18.49%	20.00%	22.31%	18.00%	19.68%	22.07%
Nick	21.67%	20.50%	12.58%	22.31%	23.02%	14.00%	22.31%	$24.00\%\pm0.017\%$	16.29%

TABLE IV: Binarization Time [ms] of 65237 DataMatrix Barcodes

Binarization method	Otsu	Niblack	Sauvola	Wolf	Nick
Mean	2	120	73	656	502
Standard deviation	0.6	20	24	238	46



Fig. 12: Binarization Results. Column order from right to left: Original, Otsu, Niblack, Sauvola, Wolf, Nick. Local binarization methods w and |k| parameters for the first two rows: (15, 0.05), middle two rows: (17, 0.1), last two rows (19, 0.2).  $\checkmark$ - barcode was decoded after binarization,  $\varkappa$ - barcode was not decoded after binarization

factor for the ZXing decoder. The fourth row demonstrates an example of uneven background illumination, where only the

Niblack image was decoded. The other binarized images had a large amount of white space in the middle of the barcode

TABLE V: Deep Learning Results Summary: best two results are highlighted and provided with two standard deviation margin

Architecture	Train Data MSE	Train Data SSIM	Train Data Read Rate	Test Data Read Rate
Cae	41.03	0.9653	82.96%	55.60%
Unet	10.65	1.0285	97.08%	70.72%
Attention Unet	11.32	1.0326	98.86%	$72.22\% \pm 0.114\%$
Unet3+	10.77	1.0298	97.27%	$74.40\% \pm 0.132\%$



Fig. 13: Neural Networks Training/Validation Losses



Fig. 14: Neural Network Type 1 Enhancement (barcodes were decoded after enhancement): column left to right order: Original, Cae, Unet, Attention Unet, Unet3+



Fig. 15: Neural Network Type 2 Enhancement (barcodes were not decoded after enhancement): column left to right order: Original, Cae, Unet, Attention Unet, Unet3+

which could have prevented them from being decoded. In the fifth row, both Otsu and Niblack images were not decoded, which can be explained by incorrectly separating background and foreground. The Sauvola, Wolf, and Nick images appeared very similar, however, the Nick image was a bit thinner. It cannot be concluded why the Sauvola image was not decoded. In the very last row, all the binarized images were decoded except for the Niblack image which contained a large amount of surrounding noise. The only explanation behind the decoded Niblack image in the first row and not decoded Niblack image in the last row might be related to the decoding algorithm specifics of the ZXing decoder.

Taking into account a significant transcendence of the Otsu method in terms of execution time (see Table IV), and very high read rate results (compared with other binarization methods in this work), it can be concluded that it is the most suited method for DataMatrix thresholding.

One way to potentially improve local binarization methods (e.g. Nick) is to select the window w adaptively depending on the image size. This might help with dealing with uneven background barcode image lighting.

As mentioned in related works [13, 15], the barcode decoding rate can be improved by applying morphological operations such a dilation and erosion. The experimental scope of this work has shown that in some cases, the above-mentioned morphological operations can improve the DataMatrix decoding rate. More specifically, the Wolf image in the fifth row of Figure 12, was decoded after applying dilation operation on the initially undecoded binarized image.

Another way to improve the local binarization method was suggested by works published by Shafait et al. [49] and Chan [50] that proposed several optimizations to the existing algorithms that led to substantial binarization speed improvement that came close to the Otsu algorithm performance and were independent of the sliding window size.

While this paper covered some of the most popular binarization methods, other approaches such as Gatos [51], Improved Sauvola [52] and WAN [53] were found successful in historical and degraded document binarization, and, as a part of future work could be incorporated in DataMatrix binarization.

#### B. Deep Learning

Four different neural network architectures were trained to enhance a DataMatrix image, and subsequently improve the barcode read rate. The after-enhancement read rate results are summarized in Table V.

The last column of Table V clearly shows that Unet-based architectures lead to notably higher read rates in comparison with convolutional autoencoders. Another observation that points to Unet-based solution superior performance is the approximately 200 times lower training and validation losses depicted in Figure 13. Accurately reconstructing (enhancing) a DataMatrix barcode is a challenging task due to a large amount of image content (barcode cells). It is assumed that Unetbased solutions are far more effective in enhancing DataMatrix barcodes because they incorporate long skip connections that allow to combine high-resolution features in the output layer. Long skip connections help to recover spatial information lost during downsampling which results in a more accurate output image reconstruction [54].

Image enhancement quality can be directly accessed using similarity scores. The mse and ssim values together with the read rates on the training set can be found in Table V. The training set read rates are directly proportional to both mse and ssim scores which shows that the more accurate the predicted image, (in comparison with the ground truth) the higher the read rate. This, however, does not hold for the read rates on the test set where Unet3+ model had the best performance (74.40%), and CAE the worst (55.60%). Decoding most of the test samples (that are potentially harder to decode) demonstrates neural network generalization ability. The reason behind Unet3+ winning performance on the test set can be explained by its full-scale skip connections. Both Unet and Attention Unet incorporate only plain skip connections that fail to sufficiently explore DataMatrix image information from full scales which might lead to an inaccurate barcode enhancement.

The examples of barcode enhancement shown in Figure 14 prove that Unet-based deep learning solutions can almost perfectly reconstruct a barcode image. Minor side effects such as occasional grey-scale spots, do take place, however,

they do not affect the overall image quality and do not prevent a barcode from being decoded. The deep learning barcode enhancement approach allows to restore barcodes (see Figure 14) from poor contrast (row 1), blur (row 3), geometrical transformation such as shearing, and barcode modules narrowing (row 4).

The ineffective barcode enhancement samples that could not be decoded are shown in Figure 15, where the majority of the barcode content was either smudged or covered with grey-scale paint. The reason behind the failed enhancements is explained by severe barcode deformations that appear quite rarely in the training dataset, and thus do not allow a neural network to sufficiently learn its specific patterns.

Recent publications made by Chen et al. [55] and Cao et al. [56] presented Unet-like architectures (TransUnet and SwinUnet respectively) based on a visual transformer model. These architectures were predominantly addressed to improve medical segmentation quality. Both TransUnet and SwinUnet showed excellent performance and generalization ability.

Within the experimental scope of this work, the SwinUnet model (1.8M parameters) was trained for 10 epochs and resulted in a 48% read rate on the test set. Unlike the Unet models described in this work, SwinUnet has a wider choice of hyper-parameters which, as a part of future work, creates room for various architectural decisions.

Another potential improvement to this work is training Unet-based models on a considerably larger DataMatrix dataset including more barcodes with large distortions. In case of the unavailability of a larger dataset, a possible solution could be training Unet-based networks on augmented data incorporating barcode geometrical distortions (at this moment, the Unet-based network cannot accurately enhance barcodes with large geometrical distortions (see Figure 15)). Such an approach might lead to a better barcode enhancement, which subsequently might result in a better DataMatrix decoding rate.

### C. Binarization vs Deep Learning

The read rate results have identified the Unet-based approach far more superior than image binarization in terms of barcode enhancement. This might be explained by the fact that Unet-based neural networks were trained to reconstruct the *ideal* barcode image, overcoming various types of distortion, and, therefore, making it as easy as possible, for the ZXing library to decode the enhanced samples. Binarization, on the other hand, is a rather limited tool, that can only solve image contrast problems and uneven illumination issues. Binarization is not able to correct barcode image geometrical transformations.

The advantage of the binarization methods is that, unlike neural networks, they do not require any training and thus do not depend on a particular dataset (e.g. DataMatrix, Aztec, QR-code) which makes it a more generic barcode preprocessing tool.

# D. Deep Learning vs State-of-the-art Swift Decoder

The experiments in this work have shown that using Unet3+ architecture for DataMatrix enhancement resulted in a 74%

read rate on the test set which accounts for 6406 decoded samples out of 8657 test samples. This indicates the total amount of decoded images from the DataMatrix dataset (65237 samples) is equal to 62986 (56580 decoded without any enhancement plus 6406 decoded using Unet3+) which is 96.55% of the barcode samples decoded by the state-of-the-art Swift decoder.

# VII. CONCLUSION

In this work, two fundamentally different DataMatrix enhancement ways were proposed intending to increase the barcode read rate. Traditional image binarization was contrasted with state-of-the-art deep learning solutions. The results identified the Unet architecture as a more superior tool for DataMatrix image pre-processing that essentially was able to solve not only the binarization problem but also correct various barcode geometrical transformations. Regarding the main research question, the Unet-based architecture performance was several times better than the performance of the binarization methods. More specifically, the Unet3+ architecture demonstrated the best performance - 74% read rate improvement on the test set, while the Nick binarization method resulted only in 24% read rate improvement.

#### REFERENCES

- [1] Natalie Gwenner. Data matrix code: a barcode with special skills. https://www.weber-marking.com/blog /data-matrix-code-a-barcode-with-special-skills/, 2019.
- [2] Assosication for Advanced Automation. The most common causes of unreadable barcodes. https://www.automa te.org/tech-papers/the-most-common-causes-of-unreada ble-barcodes, 2015.
- [3] Pawan Kumar and Nikita Goel. Image resolution enhancement using convolutional autoencoders. In *Pre*sented at the 7th Electronic Conference on Sensors and Applications, volume 15, page 30, 2020.
- [4] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using convolutional auto-encoders with symmetric skip connections. arXiv preprint arXiv:1606.08921, 2016.
- [5] Siavash Arjomand Bigdeli and Matthias Zwicker. Image restoration using autoencoding priors. arXiv preprint arXiv:1703.09964, 2017.
- [6] ISO 16022:2000(E). Information technology International symbology specification — Data matrix. Standard, International Organization for Standardization, Geneva, CH, May 2000.
- [7] Wikipedia. Data Matrix Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Data% 20Matrix&oldid=1032955386, 2021. [Online; accessed 04-August-2021].
- [8] Sean Owen et al. Zxing. https://github.com/zxing/zxing, 2013.
- [9] RChancha. Comparison between huawei scan kit and zxing. https://rchancha.medium.com/comparison-betwe en-huawei-scan-kit-and-zxing-871f0fca72b5, 2020.

- [10] The top 28 barcode scanner open source projects. https: //awesomeopensource.com/projects/barcode-scanner, .
- [11] 55 open source barcode scanner software projects. https: //opensourcelibs.com/libs/barcode-scanner, .
- [12] Top 8 barcode-scanner open-source projects. https://ww w.libhunt.com/t/barcode-scanner, .
- [13] Nathan P Brouwer. Image pre-processing to improve data matrix barcode read rates. PhD thesis, University of New Hampshire, 2013.
- [14] Ennio Ottaviani, A Pavan, M Bottazzi, E Brunelli, F Caselli, and M Guerrero. A common image processing framework for 2-d barcode reading. In *IEE conference publication*, number 465, pages 652–655, 1999.
- [15] Jianhua Li, Zhi Shen, Chaoning Yan, Nan Dong, and Haimin Liang. A method of image processing with qr code ablated on rough and highly reflective metal surface by laser. In *MATEC Web of Conferences*, volume 232, page 02024. EDP Sciences, 2018.
- [16] Rongjun Chen, Yongxing Yu, Xiansheng Xu, Leijun Wang, Huimin Zhao, and Hong-Zhou Tan. Adaptive binarization of qr code images for fast automatic sorting in warehouse systems. *Sensors*, 19(24):5466, 2019.
- [17] Lina Huo, Jianxing Zhu, Pradeep Kumar Singh, and Pljonkin Anton Pavlovich. Research on qr image code recognition system based on artificial intelligence algorithm. *Journal of Intelligent Systems*, 30(1):855–867, 2021. doi: doi:10.1515/jisys-2020-0143. URL https://doi.org/10.1515/jisys-2020-0143.
- [18] Prime vision. https://www.primevision.com/.
- [19] Swift decoder. https://sps.honeywell.com/us/en/products /sensing-and-iot/barcode-scan-engines-modules-and-de coding-software/swiftdecoder-barcode-decoding-softwa re.
- [20] Arun Gandhi. Data augmentation how to use deep learning when you have limited data—part 2. https: //nanonets.com/blog/data-augmentation-how-to-use-dee p-learning-when-you-have-limited-data-part-2/, 2021.
- [21] Yi Xu, Asaf Noy, Ming Lin, Qi Qian, Hao Li, and Rong Jin. Wemix: How to better utilize data augmentation. arXiv preprint arXiv:2010.01267, 2020.
- [22] Archana A Shinde and DG Chougule. Text preprocessing and text segmentation for ocr. *International Journal of Computer Science Engineering and Technol*ogy, 2(1):810–812, 2012.
- [23] HBY coding academic. Otsu thresholding mathematical secrets behind image binarization. https://hbyacade mic.medium.com/mathematical-secrets-behind-image-b inarization-otsu-thresholding-25edf8d7cb60, 2021.
- [24] Wikipedia. Otsu's method Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=O tsu's%20method&oldid=1020699671, 2021. [Online; accessed 03-August-2021].
- [25] Brian Pollack, Saptaparna Bhattacharya, and Michael Schmitt. Bayesian block histogramming for high energy physics. arXiv preprint arXiv:1708.00810, 2017.
- [26] THE CRAFT OF CODING. Image binarization: What

about local thresholding algorithms? https://craftofcodin g.wordpress.com/2017/04/05/image-binarization-7-what -about-local-thresholding-algorithms/.

- [27] Khurram Khurshid, Imran Siddiqi, Claudie Faure, and Nicole Vincent. Comparison of niblack inspired binarization methods for ancient documents. In *Document Recognition and Retrieval XVI*, volume 7247, page 72470U. International Society for Optics and Photonics, 2009.
- [28] Jingyu He, QDM Do, Andy C Downton, and JinHyung Kim. A comparison of binarization methods for historical archive documents. In *Eighth International Conference* on Document Analysis and Recognition (ICDAR'05), pages 538–542. IEEE, 2005.
- [29] T Romen Singh, Sudipta Roy, O Imocha Singh, Tejmani Sinam, Kh Singh, et al. A new local adaptive thresholding technique in binarization. *arXiv preprint arXiv:1201.5227*, 2012.
- [30] Wayne Niblack. An introduction to digital image processing. Strandberg Publishing Company, 1985.
- [31] Jaakko Sauvola, Tapio Seppanen, Sami Haapakoski, and Matti Pietikainen. Adaptive document binarization. In Proceedings of the Fourth International Conference on Document Analysis and Recognition, volume 1, pages 147–152. IEEE, 1997.
- [32] Christian Wolf and J-M Jolion. Extraction and recognition of artificial text in multimedia documents. *Formal Pattern Analysis & Applications*, 6(4):309–326, 2004.
- [33] Bilal Bataineh, Siti NHS Abdullah, Khairuddin Omar, and M Faidzul. Adaptive thresholding methods for documents image binarization. In *Mexican Conference* on Pattern Recognition, pages 230–239. Springer, 2011.
- [34] Dr. Dataman. Convolutional autoencoders for image noise reduction. https://towardsdatascience.com/con volutional-autoencoders-for-image-noise-reduction-32fc e9fc1763, 2019.
- [35] Mizuho Nishio, Chihiro Nagashima, Saori Hirabayashi, Akinori Ohnishi, Kaori Sasaki, Tomoyuki Sagawa, Masayuki Hamada, and Tatsuo Yamashita. Convolutional auto-encoder for image denoising of ultra-low-dose ct. *Heliyon*, 3(8):e00393, 2017.
- [36] Dominic Monn. Denoising autoencoders explained. http s://towardsdatascience.com/denoising-autoencoders-expl ained-dbb82467fc2, 2017.
- [37] Sefik Ilikin Serengil. Convolutional autoencoder: Clustering images with neural networks. https://sefiks.com/ 2018/03/23/convolutional-autoencoder-clustering-imag es-with-neural-networks, 2018.
- [38] Donghwi Hwang, Kyeong Yun Kim, Seung Kwan Kang, Seongho Seo, Jin Chul Paeng, Dong Soo Lee, and Jae Sung Lee. Improving the accuracy of simultaneously reconstructed activity and attenuation maps using deep learning. *Journal of Nuclear Medicine*, 59(10):1624– 1629, 2018.
- [39] Nikolas Adaloglou. Intuitive explanation of skip connections in deep learning. https://theaisummer.com/skip-c onnections/, 2020.

- [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [41] Robin Vinod. A detailed explanation of the attention unet. https://towardsdatascience.com/a-detailed-explanat ion-of-the-attention-u-net-b371a5590831, 2020.
- [42] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, et al. Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999, 2018.
- [43] Sik-Ho Tsang. Reading: Unet 3+ a full-scale connected unet (medical image segmentation). https://sh-tsa ng.medium.com/reading-unet-3-a-full-scale-connectedunet-medical-image-segmentation-ebb5e7f53caa.
- [44] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059. IEEE, 2020.
- [45] Data Monsters. A quick overview of methods to measure the similarity between images. https://medium.com/@d atamonsters/a-quick-overview-of-methods-to-measure-t he-similarity-between-images-f907166694ee, 2020.
- [46] Christopher Thomas. Deep learning image enhancement insights on loss function engineering. https://towardsdat ascience.com/deep-learning-image-enhancement-insight s-on-loss-function-engineering-f57ccbb585d7, 2020.
- [47] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [48] David M Rouse and Sheila S Hemami. Understanding and simplifying the structural similarity metric. In 2008 15th IEEE international conference on image processing, pages 1188–1191. IEEE, 2008.
- [49] Faisal Shafait, Daniel Keysers, and Thomas M Breuel. Efficient implementation of local adaptive thresholding techniques using integral images. In *Document recognition and retrieval XV*, volume 6815, page 681510. International Society for Optics and Photonics, 2008.
- [50] Chungkwong Chan. Memory-efficient and fast implementation of local adaptive binarization methods. arXiv preprint arXiv:1905.13038, 2019.
- [51] Basilios Gatos, Ioannis Pratikakis, and Stavros J Perantonis. Adaptive degraded document image binarization. *Pattern recognition*, 39(3):317–327, 2006.
- [52] Zineb Hadjadj, Abdelkrim Meziane, Yazid Cherfa, Mohamed Cheriet, and Insaf Setitra. Isauvola: Improved sauvola's algorithm for document image binarization. In *International Conference on Image Analysis and Recognition*, pages 737–745. Springer, 2016.

- [53] Wan Azani Mustafa and Mohamed Mydin M Abdul Kader. Binarization of document image using optimum threshold modification. In *Journal of Physics: Conference Series*, volume 1019, page 012022. IOP Publishing, 2018.
- [54] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *Deep learning and data labeling for medical applications*, pages 179–187. Springer, 2016.
- [55] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.
- [56] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swinunet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021.