Optics-free imaging of complex, non-sparse and color QR-codes with deep neural networks: supplement

SOREN NELSON, EVAN ScULLION, AND RAJESH MENON*

Department of Electrical & Computer Engineering, University of Utah, Salt Lake City, UT 84112, USA

*menon@eng.utah.edu

This supplement published with The Optical Society on 25 August 2020 by The Authors under the terms of the Creative Commons Attribution 4.0 License in the format provided by the authors and unedited. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

Supplement DOI: https://doi.org/10.6084/m9.figshare.12816200

Parent Article DOI: https://doi.org/10.1364/OSAC.403295
Optics-free imaging of complex, non-sparse and color QR-codes with Deep Neural Networks: Supplementary Information

SOREN NELSON,1 EVAN SCULLION,1 AND RAJESH MENON1,*

1Department of Electrical & Computer Engineering, University of Utah, Salt Lake City, UT 84112, USA
*rmeson@eng.utah.edu

1. Black QR codes

We gathered 30,000 training, 5,000 validation, and 1,000 test images of black QR codes on a white background with the sensor a distance of z=1, 5, and 10mm away from the display. The QR codes are 25x25 pixels with a 4 pixel white border on each side. Each QR code is made with a random 10 character encoding. The neural network architecture used is a variation of the UNet++ architecture (https://arxiv.org/pdf/1807.10165.pdf), as shown in figure S1, which uses convolution layers along the skip connections that “aim at reducing the semantic gap between the feature maps of the encoder and decoder sub-networks.” The last layer uses the sigmoid activation to get the values in the range (0,1) and outputs a 29x29x1 array. Three networks are trained for each distance z away from the display for 100 epochs with a Binary Cross Entropy loss alongside the Adam optimizer with a learning rate of 0.001. The models were able to achieve 96.7%, 93%, and 91% accuracy for the distances z=1mm, 5mm, and 10mm respectively. The training and validation accuracy along with examples for the 1mm network are shown in figure S2, 5mm network in figure S3, and 10mm network in figure S4. Figure S5 shows a combined accuracy plot of all networks.
Figure S1

Figure S2 - Accuracy for black QR codes captured 1mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.
Figure S3 - Accuracy for black QR codes captured 5mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.

Figure S4 - Accuracy for black QR codes captured 10mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.
2. Augmented Black QR codes

Using the above dataset we synthetically rotated all training, validation, and test images randomly in the range [-5,5] degrees. The same UNet++ was trained on the rotated datasets and achieved a 96.5% on the 1mm dataset, 92.7% on the 5mm, and 91.8 on the 10mm. Figures S6, S7, and S8 show the plots as well as some examples.

Updated results:
1mm network, z = 2mm: 90%
1mm network, z = 3mm: 88%
1mm network, z = 4mm: 88%
1mm network, z = 5mm: 88%
1mm network, z = 6mm: 88%
1mm network, z = 7mm: 88%
1mm network, z = 8mm: 87%
1mm network, z = 9mm: 88%
1mm network, z = 10mm: 88%

5mm network, z = 1mm: 80%
5mm network, z = 2mm: 83%
5mm network, z = 3mm: 84%
5mm network, z = 4mm: 86%
5mm network, z = 6mm: 89%
5mm network, z = 7mm: 89%
5mm network, z = 8mm: 89%
5mm network, z = 9mm: 89%
5mm network, z = 10mm: 85%

10mm network, z = 1mm: 75%
10mm network, z = 2mm: 78%
10mm network, z = 3mm: 80%
10mm network, z = 4mm: 83%
10mm network, z = 5mm: 86%
10mm network, z = 6mm: 86%
10mm network, z = 7mm: 86%
10mm network, z = 8mm: 86%
10mm network, z = 9mm: 88%

1mm translation x=1, y=1: 88%
1mm translation x=1, y=0: 88%
1mm translation x=1, y=-1: 88%
1mm translation x=0, y=1: 88%
1mm translation x=0, y=-1: 88%
1mm translation x=-1, y=1: 88%
1mm translation x=-1, y=0: 88%
1mm translation x=-1, y=-1: 88%

1mm rotation @ z by +1 degree: 88%
1mm rotation @ z by -1 degree: 88%
1mm rotation @ x by +1 degree: 87%
1mm rotation @ x by -1 degree: 87%

5mm translation x=1, y=1: 87%
5mm translation x=1, y=0: 87%
5mm translation x=1, y=-1: 88%
5mm translation x=0, y=1: 87%
5mm translation x=0, y=-1: 89%
5mm translation x=-1, y=1: 86%
5mm translation x=-1, y=0: 87%
5mm translation x=-1, y=-1: 87%

5mm rotation @ z by +1 degree: 89%
5mm rotation @ z by -1 degree: 87%
5mm rotation @ x by +1 degree: 87%
5mm rotation @ x by -1 degree: 88%
10mm rotation @ z by +1 degree: 87%
10mm rotation @ z by -1 degree: 87%
10mm rotation @ x by +1 degree: 87%
10mm rotation @ x by -1 degree: 87%

10mm translation x=1, y=1: 88%
10mm translation x=1, y=0: 89%
10mm translation x=1, y=-1: 89%
10mm translation x=0, y=1: 88%
10mm translation x=0, y=-1: 88%
10mm translation x=-1, y=1: 87%
10mm translation x=-1, y=0: 87%
10mm translation x=-1, y=-1: 88%

Figure S6 - Accuracy for augmented black QR codes captured 1mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.

Figure S7 - Accuracy for augmented black QR codes captured 5mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.
### 3. Color QR codes

Again, we gathered 30,000 training, 5,000 validation, and 1,000 test images of QR codes on a white background with the sensor a distance of z=1, 5, and 10mm away from the display. In this experiment ⅓ of the QR codes were red, ⅓ green, and ⅓ blue. The same network architecture was used as above but the output is a 29x29x3 array instead of the 29x29x1. Once again 3 networks were trained for each distance over 100 epochs using the MSE loss with the Adam optimizer and a learning rate of 0.001. The networks received 96.1%, 90.7%, and 80.5% accuracy for the distances z=1mm, 5mm, and 10mm, respectively. The training and validation accuracy along with examples for the 1mm network are shown in figure S9, 5mm network in figure S10, and 10mm network in figure S11.

---

**Figure S8** - Accuracy for augmented black QR codes captured 10mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.

**Figure S9** - Accuracy for colored QR codes captured 1mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.
Figure S10 - Accuracy for colored QR codes captured 5mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.

Figure S11 - Accuracy for colored QR codes captured 10mm away from the display and 3 input, ground truth, and prediction examples (left to right). The entropy for each target example is displayed beneath the QR code.

4. Fourier Transform of MNIST vs Black QR code
For the fourier transform 2 images were used (shown below). The MNIST image was resized from 28x28 to 29x29 and multiplied by 1/255 to get values in [0,1] same as the QR code. The MNIST image was thresholded by setting all values $\geq 0.5$ to 1 and $< 0.5$ to 0. The following lines of python code were used.

```python
from scipy.fft import fftshift, fftfreq
import imagesc as imagesc
image = image - np.mean(np.mean(image))
asbs_fft = np.abs(fftshift(np.fft.fft2(image)))
imagesc.plot(abs_fft)
```
Fig. S12: MNIST.
Fig. S13: QR code.

5. Impact of noise

To study the impact of noise, we added Gaussian noise with 0 mean and varying standard deviation to the monochrome QR code images (total of 100 experimental sensor images were used for testing). An example is shown in Fig. S14. Then, we used ANN₁ to reconstruct the images and plotted accuracy as a function of the standard deviation in Fig. S15. Even with 10% noise, the accuracy is maintained at 90% or higher, indicating that our approach is relatively robust to noise.
Fig. S14: Example image showing addition of Gaussian noise.

Fig. S15: ANN_1 reconstruction accuracy vs standard deviation of Gaussian noise.