Supplementary information for

Genetically programmable optical random neural networks

Bora Çarpınlıoğlu, Bahrem Serhat Daniş, Uğur Teğin*

Department of Electrical and Electronics Engineering, Koç University, Istanbul, 34450, Türkiye
*utegin@ku.edu.tr

Supplementary Discussion 1: 4f Imaging System and Input Encoding

The beam expander with the Keplerian design applied to the laser beam consists of two lenses placed in parallel at a distance equal to the sum of their focal lengths and its magnification can be calculated by the ratio of the focal lengths of the constituent lenses. The lenses used in this 4f imaging system are two bi-convex lenses with focal lengths of 30 mm and 268 mm. In the 4f imaging system, input beam inversion does not pose a problem for us because the output beam has the same intensity pattern as the input due to the azimuthal symmetry of the input Gaussian beam. As a result, SLM receives a magnified laser beam whose input intensity pattern did not change. Similarly, after reflection from the SLM, we use a 4f imaging system, again using two bi-convex lenses with focal lengths of 268 mm and 50 mm.

In the encoding step, since the size of the samples in the datasets was often smaller than the size of the SLM, we performed image upsampling using bilinear interpolation for each sample such that the illuminated region on the SLM was filled with the sample. In areas where the SLM could not be filled by the rescaled image, we simply performed zero-padding.

Supplementary Discussion 2: Genetic Algorithm Details

Genetic algorithm (GA) employs biology-inspired operations such as mutation, crossover and selection. In order to utilize GA, we formulated a fitness function where higher accuracies were favored, the input parameter being angular position θ specified in terms of the steps of the motor in the interval $[0,4095]$, and the output being the ridge classification accuracy of the randomly-mapped dataset. We specified 12 generations and a population size of 4 for each generation in GA. In the first generation, all genes were selected randomly, then genes with superior classification accuracies were chosen as mating parent genes (2 in our case) and a new generation was created by mutation (random mutation with 1% probability) and crossover (single-point crossover with 80%

probability) operations. We obtained the optimal kernel like so. To ensure a nondecreasing nature for the accuracy, we kept the best gene in a given generation and passed it onto the next generation. So for each generation, 3 new genes were chosen by GA. In total, the dataset was passed through 40 locations.

To calculate the fitness function corresponding to each gene, we used a laptop with Intel Core i7-7600U CPU and 8 GB RAM that sends commands via serial communication to a microcontroller board (Arduino Uno) which processes step sequences to rotate the stepper motor to the orientation specified by genes.

Supplementary Discussion 3: Similarity Between Adjoint Random Projection Kernels

To see the similarity between different random projection kernels, we performed two additional experiments. First, we fixed a reference point on our diffuser (disk with the adhesive tape) and rotated it within a range of 9 steps in either direction, which corresponds to an angular span of $1.58°$, and recorded the resulting image at each angular position. We hypothesized that, by calculating the similarity between any image and the reference image, we could get a measure of the relation between corresponding random projection kernels. Because of its popularity in image processing and deep learning, we used the structural similarity index measure (SSIM). Fig. S1 shows the average SSIM for five trials. As can be seen in the figure, the general trend manifests itself as a negative correlation between angular displacement (in absolute terms) and similarity.

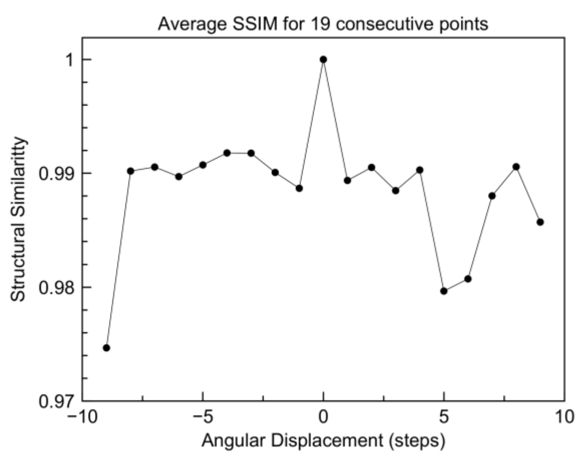

Figure S1: Average SSIM for consecutive points on the diffuser.

Second, we rotated the diffuser for $0°$, $90°$, $180°$ and $270°$ while a sample from the Fashion-MNIST dataset is displayed on the SLM. In Fig. S2 we show captured images and normalized average difference images together with SSIM values when the no rotation case is set as a reference.

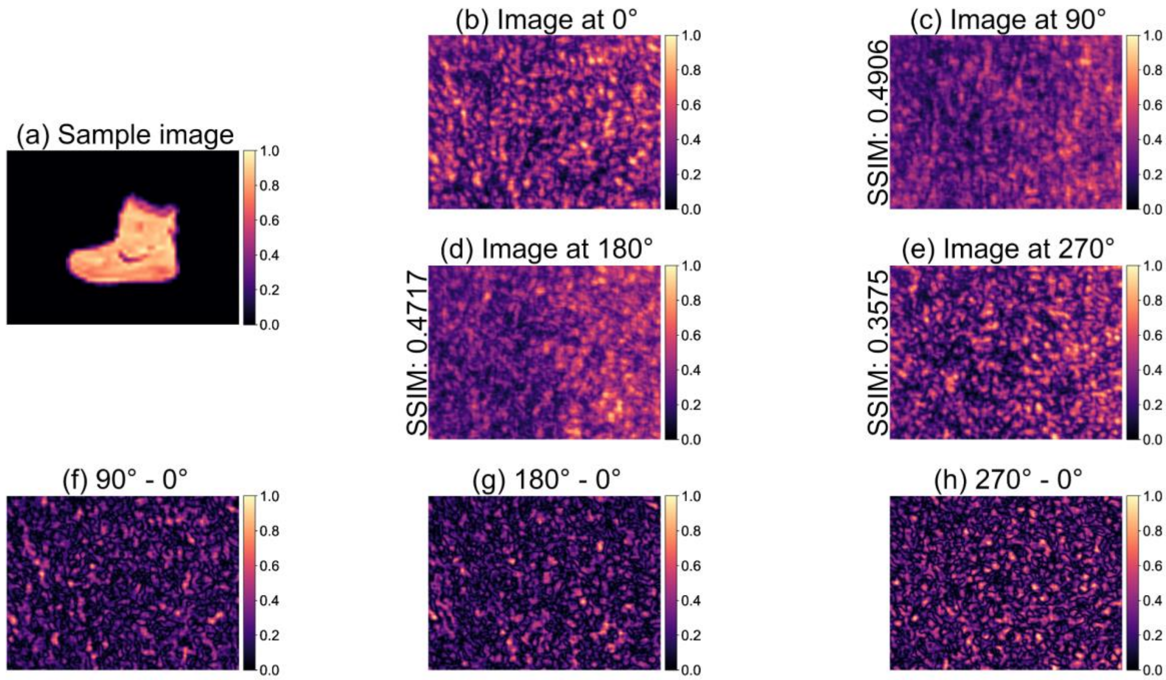

Figure S2: (a) Sample image from the Fashion MNIST dataset. (b-e) Captured images when the diffuser is rotated for $0°$, $90°$, $180°$ and $270°$. SSIM values are given to the left of the images when applicable. (f-h) Difference images when the no rotation case is considered as a reference.

Supplementary Discussion 4: Further Results on the Subset Method

As discussed in Main, accuracy results related to the COVID-19 X-Ray dataset were obtained using a subset method. We also performed GA on the full dataset to be able to assess the effectiveness of the subset method i.e., when all 2481 samples are randomly mapped. For the whole dataset, we obtained an accuracy of 91.15% compared to 90.14% obtained from 300 samples, which shows the approximation capability of the subset method. Fig. S3 shows the confusion matrices and the evolution of the accuracy in this setting.

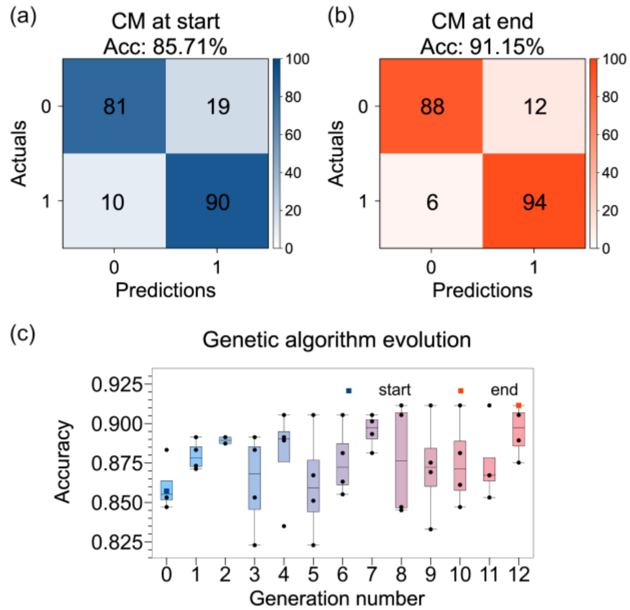

Figure S3: Results for the COVID-19 X-Ray dataset when the full set is used for GA. (a) Confusion matrix corresponding to the first GA iteration. (b) Confusion matrix obtained at the end of GA. (c) Evolution of the accuracy during GA. CM denotes confusion matrix and Acc denotes accuracy.

Supplementary Discussion 5: Detailed Visualization of Clustering Performance

Fig. S4 shows 3D LDA results on the randomly-mapped subset of the Fashion MNIST dataset for the best GA iteration from various elevation and azimuthal angles.

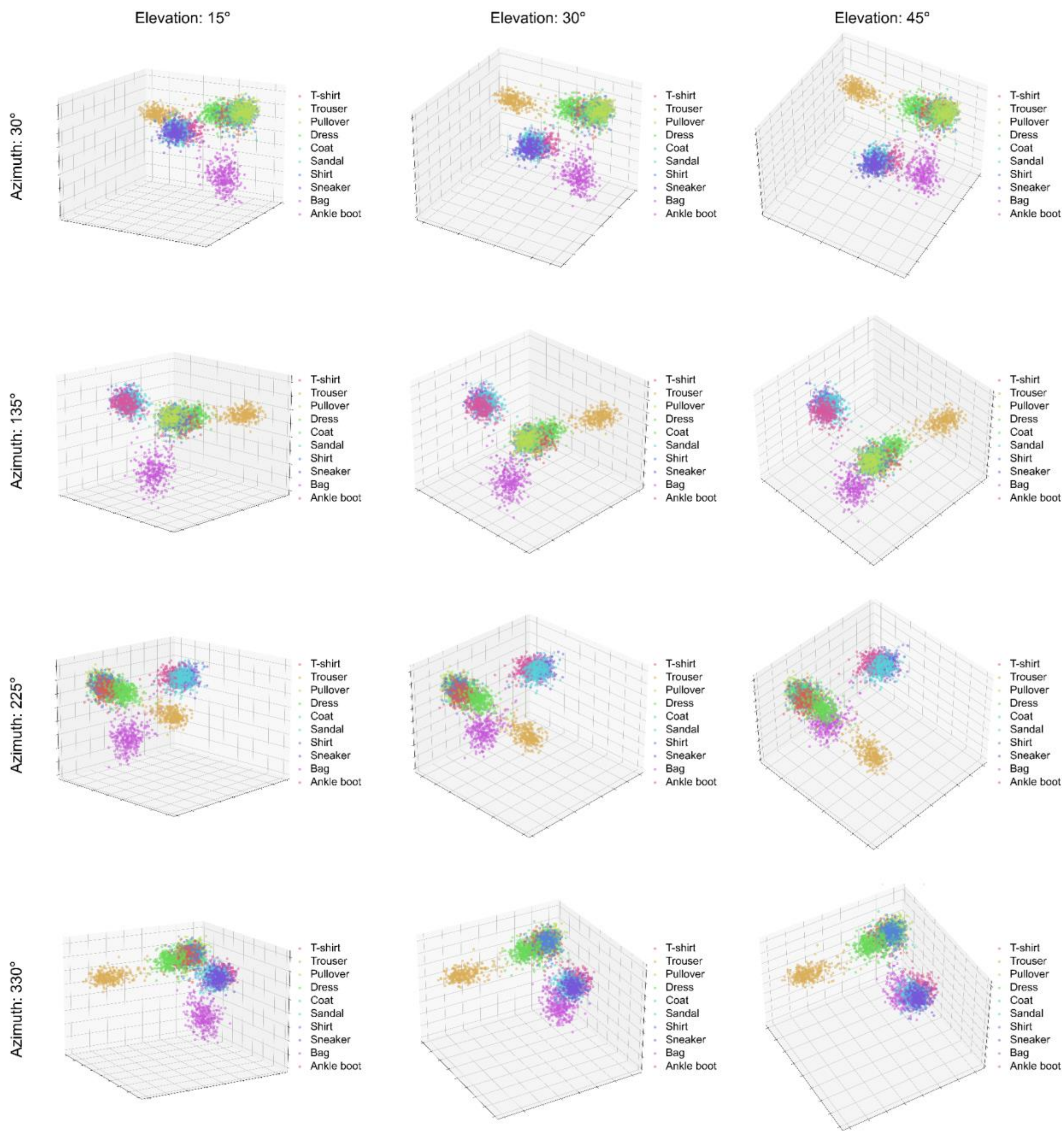

Figure S4: 3D LDA results on the randomly-mapped Fashion MNIST dataset for the best GA iteration for elevation angles of $15°$, $30°$ and $45°$ and azimuthal angles of $30°$, $135°$, $225°$ and $330°$.

Supplementary Discussion 6: Effects of Data Compression on Performance

When applying local averaging to randomly mapped data to reduce the number of pixels used for ridge classification, we fixed a pool size of $(20,16)$. Fig. S5 shows alternative pool sizes and the corresponding number of pixels along with ridge classification accuracies.

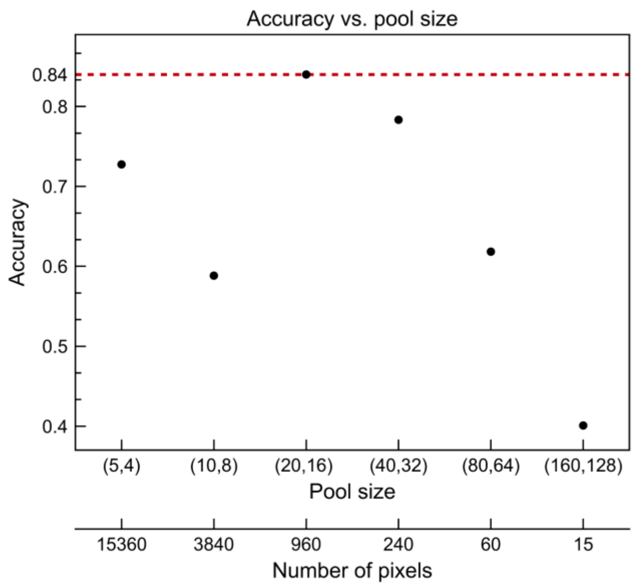

Figure S5: Ridge classification accuracies corresponding to different pool sizes used in local averaging for a randomly mapped subset of the Fashion MNIST dataset consisting of 6000 samples. The maximum accuracy is obtained for the aforementioned pool size of $(20,16)$.

Data compression can also be carried out through reducing the bit depth in captures obtained from randomly mapped inputs. In our experimental setup, we used a camera capable of recording 8-bit grayscale images. Although practically not possible because of the way digital computers process data, we emulated a camera capable of representing images using less bits to test our approach in data-scarce settings by mapping the range $0 - 255$ to a smaller range. If such an operation were possible, faster data rates would be enabled without sacrificing the classification accuracy. Fig. S6 shows GA results for different bit depths. We conclude that with suitable hardware tailored for modern machine vision tasks, our method can offer improvements in classification accuracy.

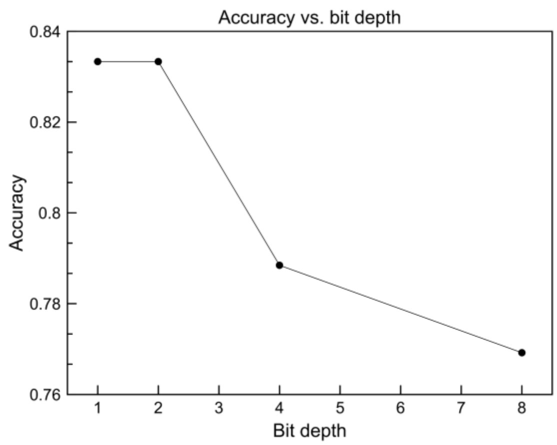

Figure S6: Dependence of ridge classification accuracy on the bit depth of captures in the Breast MNIST dataset. Separate experiments were carried out where 8-bit original captures were normalized and represented with 1, 2, and 4 bits. A maximum accuracy of 83.33% was obtained for bit depths of 1 and 2.

Table S1 demonstrates the effectiveness of our approach by comparing the classification accuracies obtained by (i) processing the randomly-mapped, 8-bit data and (ii) running GA from scratch (Fig. S6) to yield representations using less bits. These cases are denoted by the “accuracy without GA” and “accuracy with GA” rows, respectively. Without GA, after reducing 8-bit captures to 4-, 2-, and 1-bits and applying pooling and ridge classification, we observed small to no changes in classification accuracies, whereas when we let GA search for the optimal random mapping in different quantization schemes, we observed relatively higher improvements. This implies the adaptive nature of our approach to cases where high bitrates are desired when bit depth can be sacrificed. Modern machine vision tasks requiring real-time performance, therefore, could be a future application where we can test our programmable reservoir.

Table S1: Comparison of different quantization schemes in terms of classification accuracy with and without running GA for the Breast MNIST dataset.

	8-bits	**4-bits**	**2-bits**	**1-bit**
Accuracy without GA	76.92%	76.92%	77.56%	76.28%
Accuracy with GA	-	78.85%	83.33%	83.33%

Supplementary Discussion 7: Benchmarking Accuracy Levels

To provide a broader perspective, we compared our approach with other popular approaches on the same datasets, taking into account the number of parameters used in each approach. This comparison is presented in Table S2 below. From the table, one can conclude that programmable optical random neural networks generalize well across different datasets when the number of parameters are constrained.

Table S2: Comparison of different approaches in terms of their classification accuracy and number of parameters for different datasets.

Dataset	Approach	Test accuracy	Number of parameters
Breast MNIST	Our approach	76.92%	961
	ResNet-18[1]	86.3%	~11 million
	Ridge regression	66.67%	784
COVID-19 X-Ray	Our approach	90.14%	961
	ResNet-101[2]	95.58%	~44.5 million
	Ridge regression	75.45%	160,000
Fashion MNIST	Our approach	81.88%	961
	ResNet-18[3]	94.9%	~11 million
	Ridge regression	81.13%	784

References

1. Yang, J. *et al.* MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Sci. Data* **10**, 41 (2023).
2. Covid-19 Binary Classification | ResNet101 | 96%.
3. GitHub - zalandoresearch/fashion-mnist: A MNIST-like fashion product database. Benchmark — github.com.