CNN Neural Network Compression using SVD Decomposition

Antoni Dulewicz

07.11.2025

1 Introduction

The aim of this report is to become familiar with the possibilities of sampling using the compressive sensing method and to check how random projection performs as a feature extraction method (image -> vector) in representation classification.

1.1 Relationship between reconstruction quality and the number of samples M in a set of N points

In order to evaluate the impact of the number of samples M on image reconstruction quality, experiments were conducted for two types of data: *smooth* (smooth image) and *textured* (textured image). For each configuration, mean squared errors (MSE) and mean absolute errors (MAE) were calculated. $N = 64$ was assumed.

Table 1: Reconstruction error values for different numbers of samples M.

Image type	Number of samples M	MSE	MAE
smooth	50	0.073211	0.206209
smooth	100	0.037500	0.146185
smooth	200	0.003654	0.046571
smooth	300	0.000423	0.015975
smooth	400	0.000141	0.009167
smooth	500	0.000070	0.006400
smooth	600	0.000050	0.005382
textured	50	0.017756	0.105444
textured	100	0.014255	0.095968
textured	200	0.002000	0.035190
textured	300	0.002278	0.038092
textured	400	0.001605	0.031892
textured	500	0.001512	0.031116
textured	600	0.001401	0.029822

The presented data shows that an increase in the number of samples M leads to a clear improvement in reconstruction quality for both types of images. In the case of smooth data, a rapid decrease in error is observed even with a small increase in the number of samples, while for textured images the improvement is slower, indicating greater structural complexity and more difficult reconstructability.

For values $M \geq 400$, the MSE and MAE errors reach very low values, suggesting convergence of the reconstruction to the original. The differences between image types confirm that texture characteristics significantly affect the efficiency of the recovery process.

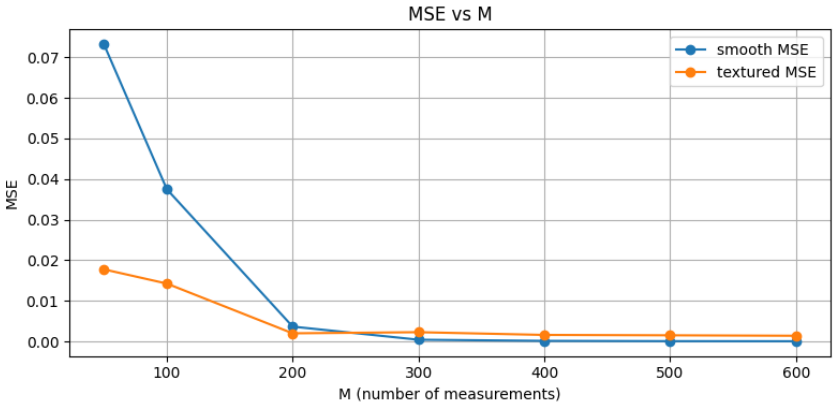

Figure 1: Dependence of MSE on the number of samples M for *smooth* and *textured* image types.

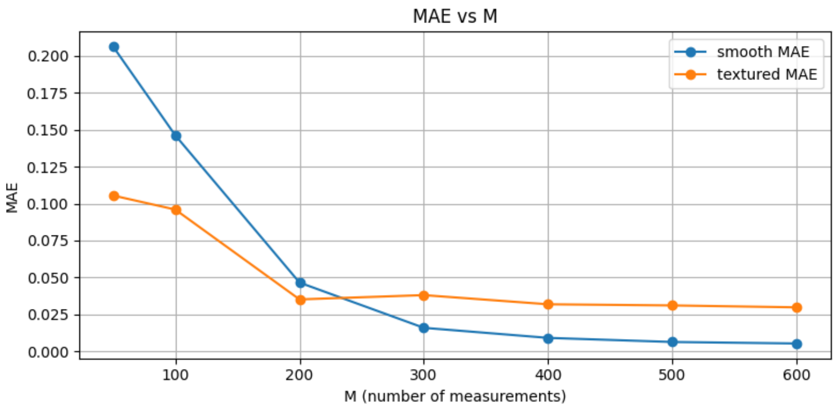

Figure 2: Dependence of MAE on the number of samples M for *smooth* and *textured* image types.

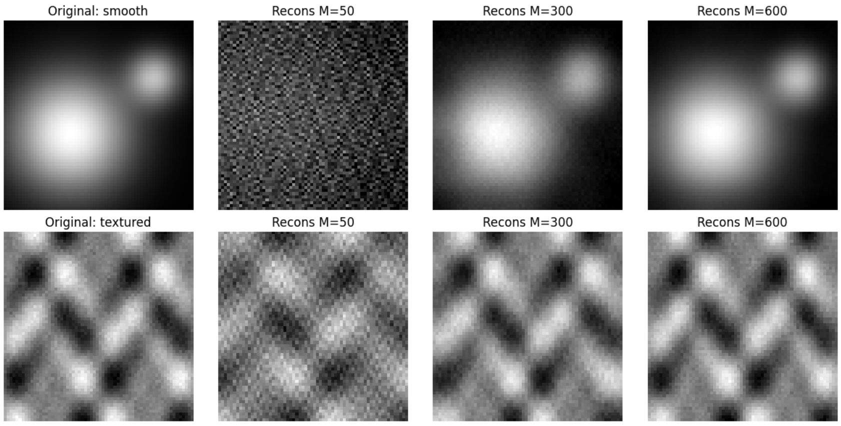

Figure 3: Comparison of selected reconstructions for different values of M.

2 Experiments with image classification using CNN and MLP networks

2.1 Datasets and preprocessing

Two popular datasets were used in the experiment:

- **CIFAR-10** – containing 60 000 color images (32×32 pixels) divided into 10 object classes (e.g., airplanes, dogs, cars). Due to the complexity of textures and colors, this dataset is considered more difficult to classify.
- **Fashion-MNIST (FMNIST)** – consisting of 70 000 images of clothing (28×28 pixels, grayscale) also divided into 10 classes (e.g., shirt, shoe, bag). This data is simpler, with less complex structure.

In both cases, images were converted to grayscale to standardize input data and reduce computational complexity. Training and test sets were then separated in a proportion adapted to hardware capabilities.

2.2 Training convolutional neural networks (CNN)

For classification of original images, a simple convolutional neural network (CNN) with the following structure was used:

```
Conv2D(32, (3,3), relu) -> MaxPooling(2,2)
Conv2D(64, (3,3), relu) -> MaxPooling(2,2)
Conv2D(128, (3,3), relu) -> MaxPooling(2,2)
Conv2D(128, (3,3), relu)
Flatten -> Dense(128, relu) -> Dropout(0.5)
Dense(10, softmax)
```

The model was compiled using the *Adam* optimizer and *sparse categorical crossentropy* loss function. The network was trained with cross-validation to obtain a reliable generalization estimate.

Table 2: Cross-validation results for CNN network.

Dataset	CNN CV Mean	Standard deviation
CIFAR-10	0.3678	$\pm$ 0.0503
FMNIST	0.8663	$\pm$ 0.0048

The results confirm a large difference in dataset difficulty: the network achieved high accuracy on simple FMNIST images, while in the case of CIFAR-10 the effectiveness was significantly lower, which results from the greater complexity of the data.

2.3 Transformation to vector space and training MLP network

The next step was to transform the image data into an M-dimensional vector space. For this purpose, a random projection matrix was used, which allows each image to be mapped as an M-dimensional vector. As a result, two datasets were created:

- M**CIFAR10** – vectors obtained from projecting the CIFAR-10 dataset,
- M**FMNIST** – vectors obtained from projecting the FMNIST dataset.

On such prepared data, a multilayer perceptron (MLP) with the following structure was trained:

```
Input(M) -> Dense(512, relu) -> Dropout(0.3)
Dense(256, relu) -> Dropout(0.3)
Dense(128, relu) -> Dropout(0.2)
Dense(10, softmax)
```

The model was compiled analogously to CNN, and the training process was also evaluated using cross-validation.

Table 3: Cross-validation results for MLP networks trained on transformed datasets.

Dataset	MLP CV Mean	Standard deviation
MCIFAR10	0.2412	$\pm$ 0.0177
MFMNIST	0.8308	$\pm$ 0.0102

3 Analysis of results and conclusions

Analysis of the obtained results indicates that data complexity is crucial for the effectiveness of models trained on different representations. The CNN network, using local convolutional filters, performs better with data having spatial structure – such as images in CIFAR-10. Despite lower absolute accuracy, this model can recognize dependencies between pixels, which an MLP trained on random projections cannot do.

In the case of simpler FMNIST images, the differences between CNN and MLP are much smaller. Transformation to vector space does not cause significant information loss, allowing the MLP to achieve results close to CNN (0.83 vs 0.86). This means that for data with simple structure and limited hardware resources, dimensionality reduction and replacing CNN with MLP can be a beneficial compromise.

Conclusions:

- Transformation to M-dimensional space is cost-effective for simple, low-complexity data (e.g., FMNIST), when the goal is to reduce computational costs.
- For complex data (e.g., CIFAR-10), the loss of spatial information causes a significant decrease in classification quality.
- MLP networks can be an efficient alternative to CNN only in situations where data has low structural complexity or when hardware limitations require simpler models.