Recognition of the MNIST-dataset with skeletonized images

A N Shevlyakov¹, V P Lobanov²

¹Sobolev Institute of Mathematics, 13 Pevtsova str. Omsk 644099, Russia
²Omsk State University, 55 Mira str. Omsk 644077, Russia

E-mail : art.shevlyakov@gmail.com

Abstract. For a long time there have been many neural networks that are able to recognize handwritten numbers. In our work, we will investigate a neural network according to a specific feature, indicating the number in the picture. We will try to achieve an important image skeleton for recognition.

1. Neural network preparation

1.1. Description of data

MNIST is a dataset developed by Yann LeCune, Corinna Cortez, and Christopher Burgess to evaluate machine learning models for the handwritten digit classification problem.

The dataset was compiled from several scanned document datasets available from the National Institute of Standards and Technology (NIST). Hence the name of the dataset, such as modified NIST dataset or MNIST.

Digit images were taken from various scanned documents, normalized in size and centered. This makes it a great dataset for evaluating models, allowing the developer to focus on machine learning with minimal data cleaning or preparation.

Each image is a 28 x 28 pixel square (784 pixels in total) (Figure 1). A standard dataset is used for model evaluation and comparison, where 60,000 images are used to train the model, and a separate set of 10,000 images is used to validate it.
So there are 10 digits (0 to 9) or 10 classes to predict.
At the same time, the classes are balanced (Figure 2). This ensures that the neural network learns without any problems.

Each pixel of the image is encoded with a number from 0 to 255 (from black to white).
To recognize images from the MNIST library, we will use a convolutional neural network of the following architecture (Figure 3):
2. Skeleton Algorithms
A skeleton in computer graphics is a set of points equidistant from the boundaries of a figure. The skeleton emphasizes the geometric and topological properties of a figure, such as its connections, connectivity, length, direction, width. In fact, the skeleton is a representation of the shape of the figure, simplifying its further analysis.

Obtaining the skeleton of the region has a visual explanation, figuratively called "fire in the steppe". Consider the image area as a steppe evenly covered with dry grass, and suppose that its entire border lights up at the same time. The fire front spreads into the area, everywhere at a uniform constant speed. The result of skeletonization will be many points where the front of fire reaches simultaneously from more than one direction.

Skeleton reduces blobs to representations that are 1 pixel wide. This can be useful for feature extraction and/or representation of an object's topology.

The first skeletonization methods were developed when solving the problem of recognizing letters and text in an image, for example, on a scanned sheet of paper.

The most popular algorithms for skeletonizing a binary image include the wave algorithm, the region thinning algorithm, and the template method.

Next, we will consider two groups of skeletonization methods that are used in our work.

The skeletonize algorithm of the Scikit-image library works on the basis of the “Fast Parallel Algorithm for Decimating Digital Patterns by T. Yu. Zhang and K. Yu. Suen”

Examples of the algorithm's work can be seen in Table. 1.
We will also use the following algorithms of the ndimage library.

The distance_transform_edt method calculates the distance from non-zero (i.e., non-background) points to the nearest zero (i.e., background) point.

Calculating the transformation to the Euclidean distance of the image will give us a ridge in the center of the image, which will be the bones of our skeleton.

In our case, the background and picture are inverted. That is, the background is the image (takes up most of the canvas), and the image itself is the background (takes up less of the canvas).

Now let's move on to other transformations that will be carried out on the basis of the obtained Euclidean distances (Figure 4).

**Table 1.** Comparison of non-deformed images and images obtained using the skeletonize algorithm.

| Undeformed images | The result of skeletonize |
|-------------------|---------------------------|
| ![](undeformed_6.png) | ![](skeletonize_6.png) |
| ![](undeformed_0.png) | ![](skeletonize_0.png) |

**Figure 4.** Images obtained by calculating Euclidean distances.
The morphological_laplace method calculates the morphological Laplacian, which will make the ridge from the distance transform really pop out, which is a kind of edge detection.

After obtaining Laplacians based on Euclidean distances (Figure 5), you need to select the main ridge. We will replace each pixel in the image 0 with black or 1 with white, depending on the condition: is the pixel value on the Laplacian map less than a third of the minimum Laplacian value on the map.

![Figure 5. Images obtained by computing Laplacians.](image)

It should be noted that the resulting image (Figure 6) does not fit the definition of the image skeleton, because it is not continuous. Selecting a skeleton in a connected set, the result should be continuous.

![Figure 6. Images after applying Euclidean distances, Laplacians and replacing pixels.](image)

But the resulting images will help us in our research, although they do not correspond to the exact definition of the skeleton of the image.

### 3. Binarization

Images in the MNIST library are 28 x 28 pixels in size. Each pixel is encoded with a number from 0 to 255. Black is encoded with 0, white with 255. The rest of the numbers are encoded with shades of gray. The smaller the number that encodes the pixel, the “blacker” the pixel. The higher the number, the whiter the pixel.

By binarization of the image we mean the reduction of the picture to the form when any pixel is encoded 0 - black or 1 - white (Figure 7).
We need a constant greater than 0 and less than 255, depending on which binarization will take place. Pixels, the value of which is greater than the value of the proposed constant, will be replaced with knowledge 1, the rest with the value 0.

It should be taken into account that the value of our constant will determine how many pixels of the gradient (gray pixels encoded with values not equal to 0 or 255) will be absorbed (which pixels will become part of the newest image) or discarded (which pixels will become part of the background of the new image) binarization.

3.1. Applying ndimage methods on binarized images.

Binarization affects the result that we can obtain in the above presented way (Euclidean distances, Laplacian). Since binarization turns some non-zero pixels into background (zero) ones, then the map of Euclidean distances, and therefore the map of Laplacians, will differ from those maps that could be obtained from transformations of non-binarized images.

Thus, we got another set of data, even less in some individual cases similar to skeletons, but still retaining some features of the skeleton. This gives us a reason to consider these data worthy of our research. For comparison see Table 2.

**Table 2.** Comparison of undeformed images, non-binarized images on which algorithms from ndimage are applied and binarized images on which algorithms from ndimage are applied.

| Original images | ndimage (non binarized) | ndimage (binarized) |
|-----------------|-------------------------|---------------------|
| ![Original Image](image1.png) | ![Non-Binarized Image](image2.png) | ![Binarized Image](image3.png) |
4. The results of the thesis

4.1. Skimage skeletons (skeletonize)
Testing with a dataset obtained using the skimage library.
This is the best result available. The drop in accuracy was only 6%.
Final validation Accuracy: 92.54999542236328%.

4.2. Ndimage images
Testing with a dataset obtained using the ndimage library on non-binarized images.
This is not to say that the result is good. But this was to be expected, since the deformation carried out by this method leaves a very small part of the image.
Final validation Accuracy: 52.04999923706055%.

4.3. Ndimage images (binarized)
Testing with a dataset obtained using the ndimage library on binarized images.
This result is interesting because the recognition accuracy of this dataset is less than the recognition accuracy of the dataset obtained using skimage. Although, in most cases, skimage images will be subsets of images obtained using the ndimage library on binarized images. This can be explained by the fact that when the image is binarized, rather weak pixels (with a numerical value close to 100 and more than 100) are transformed into a part of the image. While for skeletons with skimage, such a pixel is more likely to move to the background.
Final validation Accuracy: 90.93000030517578%.

4.4. Skeleton training
Images obtained with the skimage library were highlighted as the most successful pre-skeleton deformations. In this regard, it was decided to train the same model, but on skeletonized datasets.
Final validation Accuracy 97.54999542236328%.
As expected, the recognition accuracy dropped. This is explained by the fact that this neural network is worse designed for skeletal images than for ordinary ones.

4.5. Reverse analysis
Below are the results of recognition of original images on a neural network trained on skeletons.
Final validation Accuracy 89.29000091552734%.
It can be argued that, at least on the proposed model, a neural network trained on skeletons recognizes original images worse than a neural network trained on original images recognizes image skeletons.

5. Conclusion
We found out that the skeleton of the image is an important and integral part of it. We have gained confidence that the neural network recognizes the very image of the digit, and not the background, gradient or something else.

The neural network can recognize images only by their skeleton with good enough accuracy. This result gives reason to think about whether it is possible to improve the existing accuracy. After all, if we manage to learn to work well with skeletons, then we will get a big gain from memory, because skeletons take up less disk space than full-fledged images. This means that there is a reason to further investigate skeletal recognition.

Acknowledgments
The research of the first author was funded in accordance with the state task of the IM SB RAS, project FWNF-2022-0003.

References
[1] Nikolenko S, Kadurin A, Arkhangelskaya E 2018 Deep learning. (SPb.: Peter) p 480