Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Industriales

Máster en Automática y Robótica

Applied Artificial Intelligence

Assignment 1.3: Printing Digits

Josep María Barberá Civera (17048)

February 4, 2024

ASSIGNMENT 1.3

Printing Digits

Given a (784 × 10000) matrix, where each column represents a (28 × 28) image of handwritten digits. We iterate until finding two different images of the same number, considering four different numbers. This is done thanks to the `labels` vector, which has the corresponding label in the same order as in the matrix.

1.1 Methodology

The developed code creates two arrays: one named `already_seen` and the other named `my_numbers`. The former is designed to store the four distinct numbers initially found in the `labels` vector, while the latter holds the positional indices corresponding to the selected numbers from the `labels` vector.

The program functions as follows: initially, the two aforementioned vectors are initialized. `already_seen` is assigned the first element of the `labels` vector, and `my_numbers` is initialized with the number '1', representing the index of the first number in the `labels` vector. Subsequently, the program iterates through the `labels` vector, searching for occurrences of the same number as the one stored in `already_seen`. Upon finding a match, the index of this position is saved in the `my_numbers` array.

Following this, the program identifies the first number in the `labels` vector that differs from the one already searched. Once found, this number is stored in `already_seen`, and the loop restarts. Care is taken to manage indices, ensuring the search begins at the appropriate position in the `labels` vector.

1.2 Results

The four pairs of numbers found can be seen in the Figure 1.

1.3 Discussions and Results

The careful use of iteration indices as well as auxiliary vectors is essential in the visualization of clustered or condensed information in matrices. Again, we believe that Matlab is a suitable tool for this type of algorithms. On the other hand, the use of Python could also be suitable and could be an alternative through the use of the appropriate libraries, which, underneath, present optimizations for the handling of vectorial and computationally heavy structures.

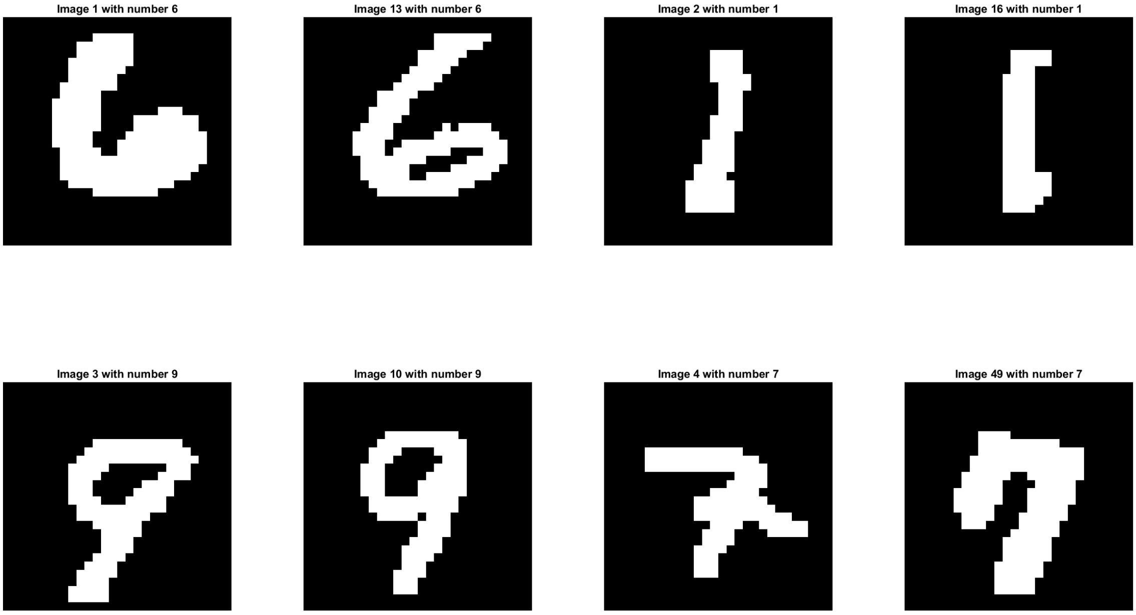

Figure 1: *Four pairs of handwritten numbers plotted from a initial matrix.*

1.4 Relevant Code

```
already_seen(1) = labels(1);
my_numbers(1) = 1;

k = 2;
l = 1;
for i = 1:times
    for j = k:length(labels)
        if labels(j) == already_seen(i)
            my_numbers(2*i) = j;
            break;
        end
    end
    for k = l:length(labels)
        if already_seen(i) == labels(k)
            continue;
        else
            break;
        end
    end
    if i == 4
        break;
    end
    already_seen(i+1) = labels(k);
    my_numbers(2*i+1) = k;
    k = k + 1;
    l = k;
end
```

The following code allows you to select the set of bits corresponding to the image within the array of numbers. It has been provided by the teacher and has been essential in this exercise.

```matlab
for a = 1:length(my_numbers)
    subplot(2,4,a);
    for i = 1:28
        for j = 1:28
            digit(i,j) = image((i-1)*28+j,my_numbers(a));
        end
    end
    imshow(digit);
    title(['Image ' num2str(my_numbers(a)) ' with number ' num2str(labels(my_numbers(a)))]);
end
```