Analysis of the features of image processing using the Hamming network on the STM-32 microcontroller

A O Chepkov, V S Klimachev¹, A I Korchagin and A I Vlasov

Bauman Moscow State Technical University, 5, 2-aj Baumanskaya, Moscow, 660041, Russia

¹E-mail: kvs250399@yandex.ru

Abstract. The paper discusses the features of solving a class of problems for pattern recognition using the STM-32 microcontroller. The problem of pattern recognition can be solved on neural networks of different architectures, the main attention is paid to the Hamming neural network model. The features of the implementation of the Hamming network based on the STM-32 microcontroller for the recognition of images entered via the touch screen are analyzed. It is experimentally shown that the network cannot always correctly process the input value and compare it with the reference value of the class for digital test images. This is due to the high degree of similarity of some images and the presence of noise. In conclusion, recommendations on the implementation of neural network algorithms for image processing on microcontrollers are given.

1. Introduction

In the context of the development of digital transformation, there often are data processing tasks that belong to the class of image recognition tasks. Such tasks include recognition of handwritten letters regardless of their scale, angle of rotation and handwriting features, diagnosis of the patient’s disease according to medical indications, determination of the type of target being followed by the known characteristics of the radar signal reflected from the target, assessment of the level of environmental pollution of the reservoir by the characteristics of the reflected laser beam, etc. [1].

In all these examples, the object under study is characterized by a set of features that form the vector \( x = (x_1, x_2, ..., x_M) \). An object can belong to one of the \( K \) known classes. So, when recognizing Russian handwritten letters, 32 classes are set. In this case, the object is most often characterized by a two-dimensional vector of features \( x = (x_1, x_2) \), specified on the set of acceptable values \( x_1 \in [x_1^-, x_1^+] \), \( x_2 \in [x_2^-, x_2^+] \). The feature space is divided into three disjoint regions \( \Omega_1, \Omega_2, \Omega_3 \), each of which combines objects belonging to the same class.

Different mathematical statements of the problem of pattern recognition are usually associated with different amounts and quality of available information about the problem being solved. Statistical data is used in the classical formulation of the image recognition problem regarding feature vectors of objects in different classes and a statistical criterion is introduced that allows assigning an object to one of the classes with certain reliability. This approach requires prior statistical processing of the available data and relies on some assumptions about the types of probability distributions that may be inaccurate.
The complexity of solving the problem of pattern recognition in practical applications is due to the fact that the boundaries between classes are unknown, their equations are not set, and a simple logical check of the current object \( x \) belonging to one of the regions \( \Omega_1, \Omega_2, \Omega_3 \) is impossible.

In recent years, the neural network approach to solving the problem of pattern recognition, which relies exclusively on experimental examples that form a training sample, has become widespread. With the help of a training sample, the parameters of the neural network (synaptic coefficients and displacements of neurons) are adjusted so that when the network is presented with the feature vector of object \( x \), it forms an indication of the object's membership class as a reaction. For example, when configuring the network parameters, you can require that, if there are \( K \) classes, it excites one of the \( K \) output neurons which number matches the number of the object's membership class.

A clear example of the use of handwritten images is the processing of characters entered via touch screens. In this case, the solution to the problem of image classification is usually assigned to the microcontroller. Let's analyze the features and problems of image processing using the computing resources of the STM-32.

2. Literature review

With the increasing adoption of the digital transformation components of the industry and the Internet of Things (IoT), touch screens are widely used [2-5]. They combine an input device and an output device, allowing creating the most convenient interfaces for user interaction with digital systems.

The introduction of methods and tools for automated recognition of input data expands the functionality of digital control systems. Various methods are used to solve the problem of determining a handwritten symbol on a matrix with known dimensions; thus, in [6], a neural network with a single hidden layer is proposed, which imposes requirements for preliminary labeling of data. As the initial set for training and control, the MNIST (Modified National Institute of Standards and Technology) set was used, which contains 60 thousand images for training and 10 thousand for testing. Each image is represented by a string, at the beginning of which there is a marker indicating the real value of the number. The image itself is monochrome (28x28 cells) and is represented as a string of 784 color intensity numbers (from 0 to 255) [7]. Since the sensor matrix will pass the node values as active or inactive, the set has been converted from monochrome to binary. The functionality of creating your own labeled sets was also implemented, which allows getting a new set or supplement an existing one. After successful completion of tests on a set of handwritten numbers, a full set of alphabet characters is added to the system.

The types of sensor arrays are different [2-5]. Their characteristics should be taken into account when implementing an automated character recognition system. For example, infrared matrices are focused on large screens and their control modules are arranged differently. The optical screens in the corners of the matrix have cameras with infrared illumination that track the angular position of the touchpoint. Then the control module calculates its coordinates and transmits them to the PC. There are schemes with both two sensors and four. An increase in the number of cameras leads to an improvement in the accuracy of touch reading, but the polling time increases [6]. The advantage of this technology is the low cost of the device and the high speed of sensor polling, and the disadvantages are poor multi-touch operation and difficult calibration.

3. Methods and models

Various methods and models are used to solve the problem of image recognition [8]. The Hamming neural network is widely used [9-15]. This network is designed to recognize the belonging class of an object defined by the vector \( x \) of bipolar features (possible values of features +1 and -1) of dimension \( M \). If \( M = n_1 \times n_2 \), then a string of bipolar features can be transformed into a rectangular matrix with \( n_1 \) rows and \( n_2 \) columns. Then a graphical interpretation of the object is possible (figure 1), in which the cells of the matrix corresponding to the features with the value +1 are represented in black, and -1 is represented in white. It is assumed that there are \( K \) classes, each of which is characterized by its reference representative-object \( x^{(k)} \), \( k = 1, 2, ..., K \).
The Hamming network accepts bipolar features of an object at \( M \) inputs and, after processing the data, activates one of the \( K \) outputs, which indicates the class of the object presented at the input.

The criterion for assigning an object \( x \) to a class is the square of the distance between the vectors \( x \) and \( x^{(k)} \), \( k = 1, 2, \ldots, K \):

\[
R(x, x^{(k)}) = \sum_{j=1}^{M} (x_j - x_j^{(k)})^2 ,
\]

where \( x_j \) is the \( j \)th bipolar feature, \( j = 1, 2, \ldots, M \).

The network assigns object \( x \) to class \( k^* \) if

\[
\min_{k=1}^{K} R(x, x^{(k)}) = R(x, x^{(k^*)}) .
\]

If the features are bipolar, then for \( R(x, x^{(k)}) \) we get:

\[
R(x, x^{(k)}) = \sum_{j=1}^{M} (x_j^2 - x_j^{(k)})^2 - 2x_j x_j^{(k)} = 2 \sum_{j=1}^{M} (1 - x_j x_j^{(k)}) = 2(M - \sum_{j=1}^{M} x_j x_j^{(k)})
\]

For the maximization criterion of the expression, the maximum is reached at \( k = k^* \) (figure 1):

\[
\arg \max_{k=1,K} I_k = \arg \max_{k=1,K} \left( \sum_{j=1}^{M} x_j x_j^{(k)} + M \right) = k^*. 
\]

![Figure 1. Graphical interpretation of the feature vector of an object.](image)

The value of \( R \in [0, 4M] \), so \( I_k \in [0.2 M] \).

To calculate the values of \( I_k \), \( k = 1, K \), a working layer consisting of \( K \) neurons with the following characteristics is used:

\[
w_{kj} = x_j^{(k)}, b_k = -M, k = 1, K .
\]

The potential \( h_k \) of the \( k \)th neuron is determined as:

\[
h_k = \sum_{j=1}^{M} w_{kj} x_j - b_k = \sum_{j=1}^{M} x_j x_j^{(k)} + M = I_k .
\]

After all the coefficients of \( h \) have been found, it is necessary to find the largest value among them, which will correspond to the defined image. The task of this work is to implement a neural network for
recognizing Arabic numerals from "0" to "9". Then to implement such a network, we will need 10 neurons.

4. Implementation of image processing using the Hamming network on the STM-32 microcontroller

As a microcontroller for implementing algorithms for processing input images, we will use the STM-32 F030R8. The microcontroller is installed on the STM-32 Nucleo debug board [16]. The board has an ST-LINK programmer, which will allow downloading the firmware to the internal memory of the microcontroller, as well as using the debug mode. The 2.8-inch TFT TouchShield touch display is used as the equipment for touch input [17, 18]. The module is a TFT display with a resolution of 320 x 240 pixels, a resistive touch panel, and a connector for connecting a microSD Flash memory card.

For experimental studies of the algorithm, we implement an input module consisting of cells that can be filled in, thereby entering various characters. In addition to the working field, the program will have a window for displaying the recognized image, as well as a reset button and additional labels explaining the operation of the interface.

The size of the cell of the working field will be 20 x 20 pixels since it will be convenient to get into this size with a stylus. We will place the working field in the lower-left corner, leaving a space on the right edge for the output window of 70 pixels, and 100 pixels on top for additional labels. We will also make an indent from the lower-left corner of 5 pixels; thus, it turns out that the size allocated for the working field is horizontally $N_h = 240 - 70 - 5 = 165$ pixels, and $N_h = 320 - 100 - 5 = 215$ pixels vertically.

Calculate the number of cells horizontally and vertically:

$$n_h = \frac{N_h}{20} = \frac{165}{20} = 8.25$$

$$n_v = \frac{N_v}{20} = \frac{215}{20} = 10.75$$

We get that the size of the working field is 8 x 11 cells. The final graphical interface of the program, implemented in the course of work, is shown in figure 2. For example, it shows a handwritten recognizable image of the number "3" in the input window, and in the right window of the program, you can see the recognized image.

![Figure 2. Graphical interface of the touch input module.](image)

![Figure 3. Reference class images.](image)
To configure the microcontroller, we will use the *Cube MX* utility. A series of reference images of class representatives were compiled to train the neural network, corresponding to the Arabic numerals from "0" to "9". For each character, a two-dimensional matrix of values is implemented, where the value "0" corresponds to an empty cell, and the value "1" corresponds to a shaded cell. The size of the working field is 8 x 11 cells. Figure 3 shows all the used reference images. Each of the symbols shown on the screen corresponds to a two-dimensional array implemented in the "C" language (figure 4).

```c
static tp_dev_t s_tTouch;
    uint16_t i, k, j, q, max_value, num_image;
    _Bool table [11][8] = {0};
    _Bool num0 [11][8] = {
        {0,1,1,1,1,1,1,0},
        {1,1,1,1,1,1,1,1},
        {1,1,1,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,0,0,0,1,1,1},
        {1,1,1,0,0,1,1,1},
        {1,1,1,1,1,1,1,0},
    };
    ...... 
    _Bool tp_status1, tp_status2 = 0;
    uint16_t mux_image [88];
    uint16_t sum_image [10];
```

*Figure 4. Listing.*

It should be noted that the values of the cell display mode are inverted vertically. This is due to the peculiarities of building an image on the touch screen.

5. **Discussion of the results**
Experimental studies were conducted on two subjects, the purpose of which was to take turns to enter the processed images on the screen. Some of the total set of recognized images are shown in Figure 5.
During the tests of the neural network, it turned out that for the presented images, the network cannot always correctly process the input value and compare it with the reference value of the class. This is due to the high degree of correlation between some images, and therefore it was decided to modify them. The images of the numbers "6" and "9" were changed. It was with the recognition of these images that the greatest number of errors and mutual substitutions were associated. After the correction, it was possible to achieve the correct recognition of handwritten images for all digits, equal to 94%; however, it is worth noting that specifically for the numbers "3" and "9", the percentage of guessing is much less, and equals to 76%. This is also due to the correlation of the characters with each other. In general, the results of the network performance tests can be called satisfactory.

6. Conclusion
A method was proposed for image processing using the Hamming network on the STM-32 microcontroller in the course of the work. This network belongs to the class of neural networks of direct propagation and has shown high-quality characteristics in the recognition of numbers. The proposed method can be used to implement a simple and relatively cheap device for interactive input of special characters, which are often used when entering formulas, thereby speeding up the process of writing complex mathematical expressions. This solution can also be used for adaptive calligraphy.

Acknowledgements
Some results of the project were obtained with the financial support of the Ministry of science and higher education for the project #0705-2020-0041 "Fundamental research of methods of digital transformation of the component base of micro-and nanosystems".

References
[1] Prudius A A, Karpunin A A and Vlasov A I 2019 Analysis of machine learning methods to improve efficiency of big data processing in industry 4.0 Journal of Physics: Conference Series 1333(3) 032065
[2] Barrett G and Omote R. 2010 Projected capacitive touch technology Information Display 3 16-21
[3] Terent'ev D S, Shakhnov V A and Vlasov A I 2017 Graphene flexible touchscreen with integrated analog-digital converter Russian Microelectronics 46(3) 192-9
[4] Shakhnov V A, Tokarev S V and Vlasov A I 2016 Electrochromic thin-film components for information representation systems *IOP Conference series: materials science and Engineering* **151** 012005

[5] Krivoshein A I, Terent’ev D S, Shakhnov V A and Vlasov A I 2018 Investigation of a capacitor array of a composite capacitive touch panel *Russian Microelectronics* **47(5)** 299-306

[6] Aminev D A, Danilevich V A, Demin A A and Samman A 2021 Recognition of handwritten symbols on a sensory matrix using a neural network *Neurocomputers: development and application* **23(1)** 32-42

[7] Prateek J 2018 Artificial Intelligence with Python (Birmingham, UK: Packt)

[8] Gulyaev Y and Galushkin A 2003 *Neurocomputers in image processing systems* (Moscow, Russia: Radio Engineering Publishing House.)

[9] Hung C A and Lin S F 1999 An incremental learning neural network for pattern classification *International Journal of Pattern Recognition and Artificial Intelligence* **13(6)** 913-28

[10] Bernasconi J and Gustafson K 1998 Contextual quick-learning and generalization by humans and machines *Network: Computation in Neural Systems* **9(1)** 85-106

[11] Du K-L and Swamy M N S 2006 *Neural Networks in a Softcomputing Framework* (Berlin, Germany: Springer)

[12] Barrios D, Carrascal A, Manrique D and Rios Ju 2003 Cooperative binary-real coded genetic algorithms for generating and adapting artificial neural networks *Neural Computing & Applications* **12(2)** 49-60

[13] Cheng-An H and Sheng-Fuu L 1995 Adaptive hamming net: a fast-learning art 1 model without searching *Neural Networks* **8(4)** 605-18

[14] Lin S F 1997 Supervised adaptive hamming net for classification of multiple-valued patterns *International Journal of Neural Systems* **8(2)** 181-200

[15] Swain R K, Kamila N K and Misra BB 2012 Recognition of the state of mind using hamming swarm net *Proceedings of the 2012 World Congress on Information and Communication Technologies, WICT* (Trivandrum, India: IEEE) pp 903-8

[16] Data sheet NUCLEO-F303RE / MCU STM32F303RET6 (ARM Cortex-M4) / ST-LINK/V2-1, https://www.chipdip.ru/product/nucleo-f303re-2, last accessed 02.02.2021

[17] Data sheet 2.8inch TFT Touch Shield, https://www.chipdip.ru/product/2.8inch-tft-touch-shield, last accessed 02.02.2021

[18] Data sheet 2.8inch TFT Touch Shield, www.waveshare.com/wiki/2.8inch_TFT_Touch_Shield, last accessed 02.02.2021

[19] 2014 UM1724. User manual. STM32 Nucleo boards (STMicroelectronics)