REALIZATION OF THE GESTURE INTERFACE BY MULTIFINGERED ROBOT HAND

Pavlovsky Vladimir[1], Stepanova Elizaveta[2]
[1]Keldysh Institute of Applied Mathematics, Moscow, Russia

vlpavl@mail.ru,
[2]Moscow State University, Moscow, Russia

pashastepanova@gmail.com

Summary. The paper considers theoretical mechanical model of a multifingered arm with 21 degrees of freedom. The main objective of the work is the creation of gesture interface. Gesture interface includes the set of gestures, the synthesis of finger control schemes for 26 gestures, as well as gesture recognition task with the help of convolutional neural network training. As the demonstration we propose to observe the results of 26 gestures recognition with the help of constructed convolutional network. For 26 classes 15600 images at different distance and at different angles were created. As a result of convolutional neural network training the accuracy of a test set classification is 76 percent.

Key words: neural network, multifingered hand, manipulator, collaborative robotics

1 Introduction

The collaborative robotics is a human-computer interaction. One of the methods of collaborative robotics is an interaction based on human gestures.
Gesture recognition task can be used in different spheres of our life: on the factory in the bustling workshop; in space, where sound waves can not propagate because of vacuum; in communication with deaf people; in household appliances control and many other tasks.

2 Problem formulation

A hand model in the software package Universal Mechanism is considered. The model consists of a palm with a single rotational degree of freedom and five fingers connected by rotational joints. Also, in this model the thumb metacarpal bone is considered similarly with other fingers' bases, i.e. a spherical type of joint connection. First phalanges are connected with a palm by a spherical joint with two degrees of freedom each. Thus, the model has 21 degrees of freedom, which is close to an actual number of degrees of freedom of the human hand. The size of a model hand is similar to my own hand's size.

The main objective of the work is the synthesis of finger control schemes, as well as the model hand's gesture recognition with the help of neural network

training. On the image below one can see a schematic representation of a model hand with degrees of freedom.

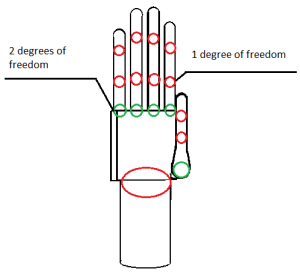


Fig. 1. Model hand, degrees of freedom

3 Experiment

Artifitial neural network - mathematical model, as well as its software implementation, built on the principle of the organization and functioning of biological neural networks.

The idea of convolutional neural networks is similar to the idea of our brain's structure. We used classical convolutional neural network in this task because according to our research and study of other similar works, it shows good results in tasks of image recognition and classification.

Description of the system:

To solve the tasks of gesture recognition we will work with a convolutional neural network.

The convolutional neural network was created in the Neural network toolbox in a software package Matlab. This program allows us to classify gesture images. A vocabulary of sign language for one hand was created, where one static gesture corresponds to one letter of the English alphabet. We teached the program to perceive static gestures of hands at different distances, at different angles of shooting.

This convolutional neural network for this task consists of 7 layers:

1. ImageInputLayer - receives the data set of images of 60x60x3 dimension, where 60x60 (pixels) is the size of the training images, 3 is the color channels (RGB).The task of this system is to distribute a set of images by classes (class - one gesture corresponding to the letter of the English alphabet). For each class we prepared 600 images. The program receives images randomly from folders with labels corresponding to classes' names

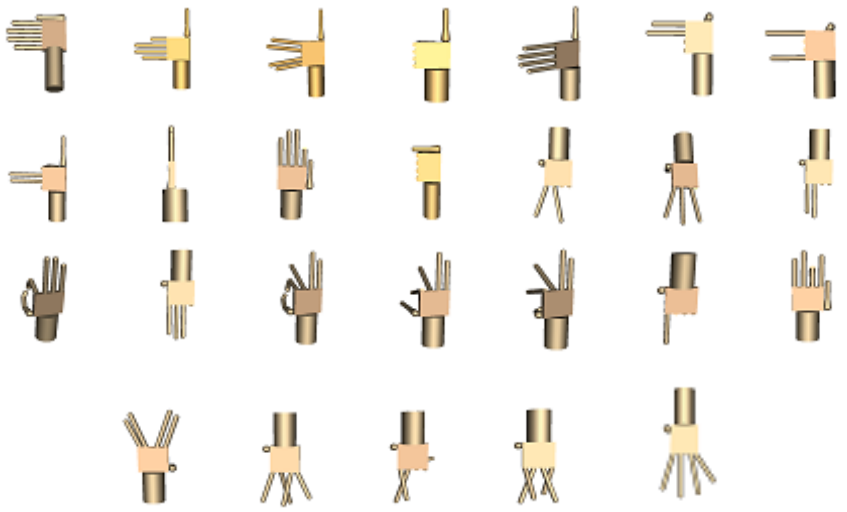


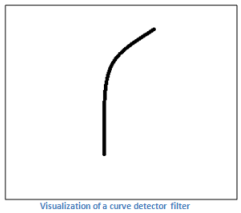
Fig. 2. The set of gestures A - Z

(A, B, C, D, etc.). This system uses 90 percent of the images for training, and 10 percent for testing. The result of testing is accuracy of classification.

2. Convolution Layer is the main block of a convolutional neural network. Input data for this layer is a 60x60x3 matrix. One of the features of the convolution layer - the kernel that runs through the entire data set, starting from the upper left corner and moving to the right step by step, multiplies with each section. The result of this matrix multiplication is summed and put to the appropriate place of new matrix. That is, the result of one multiplication must be one number. The dimension of the convolution kernel is a variable parameter and depends on the task, to solve this problem, we used a 6x6x3 kernel. With these values the training was most successful. Unlike the size of the kernel, we can't choose the kernel's values, since they must be revealed during the learning process. After passing through the entire data set, the output of the kernel is a matrix (55x55x3) - a map of features. Where a feature is an image property (lines at a certain angle, color,curves, etc.), the number of features is an adjustable parameter. For this task, the number of features required is 30.
   Let's consider this process in a simplified example. The program received an image.
   Let the first kernel be a detector of curves. To simplify the understanding, we ignore the fact that the depth of the core is 3, we consider only the upper layer. The kernel has a pixel structure in which numerical values are higher along the region that determines the shape of the curve.

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter



Visualization of a curve detector filter

Fig. 3. Curve detecting kernel

In the initial position, the kernel is in the upper left corner, it multiplies the kernel values by the pixel values of this region. Let's look at the example of the image that we want to classify, and put the kernel in the upper left corner.

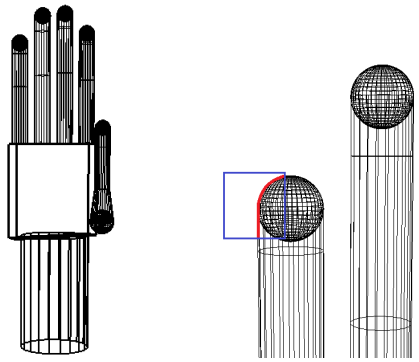


Fig. 5. Target curve on the image

Fig. 4. Input image

Multiply the value of the kernel by the image area values.

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

$*$

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Fig. 6. Successful curve detection

As a result, we get the value 6600 - a large number. Let's try to multiply the kernel to another area of the image.

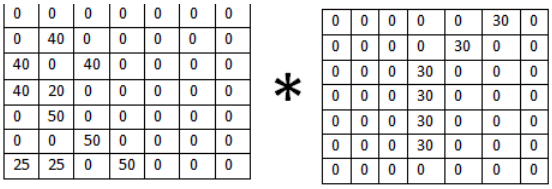


Fig. 7. Unsuccessful curve detection

As a result, the product is zero, which means that the kernel has not found the desired feature in this area. The filter we described is simplified. In fact, the feature card would look differently.

3. ReLULayer is the activation function after the convolutional layer, however, the maximum function $f(x) = max(0, x)$ is selected for activation. This function cuts out unnecessary or bad signs. A high value of 6600 from the previous argument shows that, perhaps, there is a curve on the image, and such a probability activated the filter. In the right upper corner, the value for the feature map will be 0, because there was nothing in the picture that could activate the kernel (in other words, there was no curve in this area).
4. MaxPooling - this layer takes small individual image fragments (2x2 in the case of this system) and combines each fragment into one value. The operation of the pooling reduces the spatial volume of the image (it becomes 27x27x3).
5. FullyConnectedLayer, The fully-connected layer refers to the output of the previous layer, and the features that are more associated with the individual class are determined.The output of this layer is an 26 -spatial vector.
6. SoftMaxLayer is an activation layer, that maps information at the input of a set of elements to classes.
7. ClassificationOutputLayer - displays and classifies information. For training the backpropagation method is used.

4 Conclusion

As a result, the gesture interface has been created, including the control scheme the image recognition program with the help of convolutional neural network. The network was tested on the task where we tried to classify 26 gestures and received the 76 percent accuracy of test images classification.

One can see on the image below the result of test images classification - accuracy.

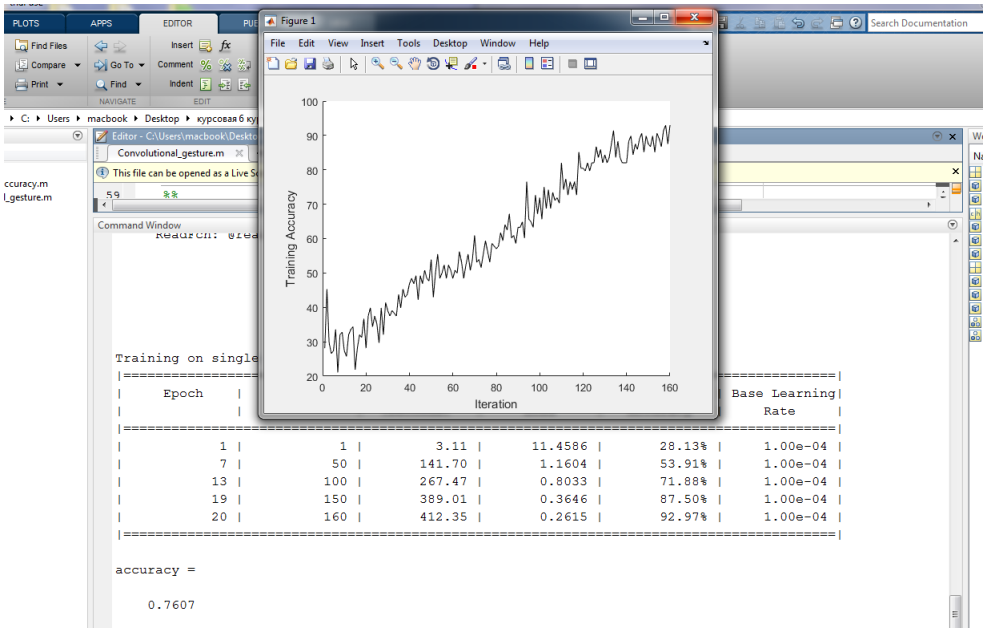


Fig. 8. Accuracy

In the nearest future we are planning to increase the accuracy of the classification task. Moreover we will consider gestures in motion (at the moment we use only images of static gestures).

References

[1] Nagapetyan V.G. Gesture recognition methods on the base of long-range images analysis. Moscow, 2013.

[2] John J. Craig. Introduction to robotics. Mechanics and control. - M.-Izhevsk: SRC "Regular and chaotic dynamics", Institute for Computer Research, 2013.

[3] Yurevich E.I. Fundamentals of robotics. - 2nd edition. - St. Petersburg: BHV-Petersburg, 2005.

[4] Formalski A.M. Anthropomorphic mechanisms movement. 1982

[5] Simon O. Haykin, McMaster University, Ontario Canada. Neural Networks. A Comprehensive Foundation.

[6] Michael Nielsen. Neural network and deep learning: http://neuralnetworksanddeeplearning.com/