

Development of a Vision System for an Intelligent Robotic Hand Prosthesis Using Neural Network Technology

Andrew A. Boshlyakov^{1*} and Alexander S. Ermakov¹

¹Bauman Moscow State Technical University, 2nd Baumanskaya str., 5/1, 105005, Moscow, Russia

Abstract. A brief review of the existing auxiliary prosthetic control systems was carried out. The concept of an intelligent prosthesis is proposed, which will expand the possibilities of application and simplify the use of the prosthesis. The required actions of the vision system in automatic and manual capture modes are considered. The sequence of operation of the subsystems of the technical vision system is determined. The possibility of implementing a prosthesis vision system based on neural network technology is shown. The method of using a ready-made neural network for recognition of objects by a prosthesis is considered. The possibilities of using the considered neural network technologies in the mathematical education of engineers are presented. A version of the prosthesis design is proposed. The possibility of constructing the described prosthesis is shown.

1 Introduction

The problem of replacing missing limbs with prostheses has existed for hundreds of years, however, even with modern advances in science and technology, prosthetic technology has not reached the level of an ideal replacement for a living human hand.

Today, bioelectric hand prostheses allow people who have lost upper limbs for some reason to partially restore lost functions. Modern devices have different degrees of freedom, a different number of drives, perform different numbers of captures and are designed for certain tasks [1, 2, 3, 4]. However, the complex design of the prosthesis in most cases causes difficulties in managing it, while the primitive design does not allow to realize the movements necessary for a person in daily activities. Moreover, a person does not feel such limbs, and he has to act with constant eye contact [5, 6, 7, 8, 9].

If the prosthesis could somehow tell the user how best to take the object, adjust its actions depending on the parameters of the captured object, or if necessary, complete the capture yourself, taking only the “take” command from the person, then control of such a device would be much it would be simpler and even allow in some situations to act “blindly” [10].

When performing an operation to capture a wide range of objects, an intelligent prosthesis must perform the following actions:

* Corresponding author: boshlyakov@bmstu.ru

1. The definition of the command to capture [11, 12, 13]. The prosthesis should proceed to capture directly only when required by the user. In addition, according to the user's actions, the decision-making system should receive primary information about the proposed capture method.
2. Classification of the subject [14, 17]. Knowing which class of objects an object belongs to, one can choose a capture corresponding to this group, or a group of captures [15].
3. Recognition of object parameters [16]. In many cases, objects of the same class require a different approach to capturing them due to different dimensions, different orientations, etc.
4. Accurate grip compression [18, 19, 20]. In this case, it is necessary not only not to deform the object, but also to prevent its possible loss.
5. The work of the decision-making system at each of the listed stages [21]. For the correct operation of the capture operation, it is necessary to analyze the user's commands, and the classification results, and the recognized parameters of the object, and the forces arising from compression.

Moreover, the design of the prosthesis should satisfy its functional capabilities and provide the ability to place components of the above systems [22, 23].

In carrying out such steps, the prosthesis should be easy to operate, and with its help it will be possible to interact with many objects encountered by people in everyday life, at work and during other periodic activities. Which, undoubtedly, will allow people with limited opportunities due to the lack of a hand to lead a full-fledged or as close as possible to a full-fledged lifestyle.

Thus, in order to increase the number of items available for capture and further use due to the intellectualization of the prosthesis, a number of technical solutions are required. In particular, neural network technologies for object recognition are relevant for solving such problems. Their use along the way can become an important link in the mathematical education of engineers due to the presence of relatively simple effective and inexpensive ways to implement such technologies.

2 Overview of existing auxiliary denture management systems

In addition to information coming directly from a person, modern prostheses have auxiliary systems that help the device better understand the grip that the user wants to apply.

1. The simplest, but no less effective, approach is implemented in the **bebionic3** prosthesis of the English company RLS Steeper [1]. The base of the thumb has a fixed number of positions, each of which defines a group of available grips and is set manually by the user. The number of different captures is also fixed.
2. Another English company proposed alternative approaches that differ in each configuration of the device:
 - a) **i-limb ultra**. i-limb ultra is the base brush [2]. Its features include:
 - vari-grip mode, which allows you to redistribute the load between your fingers;
 - thumb moves manually;
 - 2 ways to control: mobile application and combinations of muscle contractions;
 - setting the speed of the fingers;
 - restraint auto-locking, preventing objects from slipping.

Switching between types of grip can be carried out both through a mobile application (only for devices on iOS) and muscle control. Muscular control uses 4 combinations:

- constant muscle contraction (for setting a period of time);
- double reduction;
- triple reduction;
- simultaneous contraction of the flexor and extensor muscles.

Each of them is assigned a specific action selected by the user.

- b) **i-limb revolution** [2]. It is the next prosthesis in the i-limb line. Having retained all the features of its predecessor, revolution, grip chips technology has been added. Chip grips technology is another form of denture control. Through Bluetooth, the prosthesis communicates with the so-called grip chip, which secures a certain type of grip through a mobile application. When the prosthesis approaches the chip, an automatic change of grip takes place. The chip is round, about 2.5 cm in diameter, equipped with light and sound indication.
- c) **i-limb quantum** [2]. i-limb quantum is the latest in the i-limb line. Compared to revolution, it has a fourth control system in the form of i-mo technology. This technology allows you to change the type of grip, performing certain movements with the prosthesis. For this, the prosthesis is equipped with motion sensors. In all other respects, the prosthesis is identical to the i-limb revolution.

3 Prosthetic intellectualization concept

Let us first consider the general principle of constructing an intellectual prosthesis [24]. The above prosthetic control systems have a common, namely, the choice of the appropriate group of contractions, and only after that the use of a specific grasp [3, 4]. Mostly this requires actions from the person to switch the prosthesis to the appropriate mode with his free hand. In addition, this type of control does not allow capturing accurately when interacting with many classes of objects [5, 10].

Let us consider in more detail the systems proposed for implementation, as well as changes in the requirements for the design of the prosthesis that will be caused by their implementation.

3.1 Human-Machine Interface

The human-machine interface for controlling the prosthesis is a system that reads control signals from a person using specialized sensors and sends a set of data corresponding to a given grip and finger position to the input of the decision-making system. Depending on the operating mode of the prosthesis (automatic capture or manual capture with partial help of automation), the system reads and recognizes either the command to perform the capture or the parameters of the desired capture.

Modern bioelectric prostheses are usually controlled by one of three main control methods [11]:

1. reading signals from the human brain;
2. neuroelectric interface;
3. electromyographic (myoelectric) interface [12].

3.2 Technical Vision System

Technical Vision System (TVS) are widely used in various technical devices. Moreover, in commercially available prostheses TVS is not currently used. The most similar TVSs are TVS manipulators and TVS mobile robots.

The similarities between the designed TVS prosthesis and the TVS of mobile robots are in a variable working area, variable illumination and high performance requirements. The main difference is that when used in a prosthesis, the system receives strict restrictions on energy consumption and size, which also leads to a limitation of computing power [14].

The TVS of manipulators and the prosthesis are similar in the spectrum of tasks to be solved - the classification of objects and the determination of some of their parameters. At the same time, significant differences are observed both in size and capacity, and in functioning in a variable working area with high speed [15].

We define the concept of a "class of objects" as an individualized family of objects that share certain common characteristics. For example, the class "computer mouse" or "pen". The accuracy, speed and correctness of the capture of an object by a prosthesis depend on the class of the object, its size and orientation relative to the grip.

Different objects have a fundamentally different structure, respectively, require a different approach to the implementation of the capture. For example, the cup is taken by the handle, the arm is placed on the mouse on top, and the fountain pen is simply fixed in one position. Accordingly, having information about the class of the object, you can select the desired grip - the position of the fingers and hand. This information is especially important when working in automatic mode, as in this case, a person cannot directly control the process.

In this case, the capture process will depend on the parameters of a particular instance of the class. At least from their size. For example, a ping-pong ball and a medium-sized rubber ball will require different approaches. However, this is not enough to accurately and accurately complete the task. The fact is that in many cases capture of an object of a class from a certain specific position is required, which also requires data on the orientation of the hand relative to the axes of the object. For example, when capturing a computer mouse, you need to take it with the buttons in the direction of the fingers of the hand, and not perpendicular to the axis of the hand.

So, TVS of the prosthesis should successively solve the problems of classification [16] of an object and recognition of its certain signs [17]. The system should include a computing device of sufficient power with small sizes and power consumption and a sufficient set of sensors to obtain information about the object in the working area when working in various conditions of humidity and light. Moreover, due to the specifics of the work, it is necessary to ensure the necessary system speed and the small dimensions of its elements.

3.3 Force-Moment Management

Force-Moment Management for a prosthesis is a type of regulation based on processing information about the compression force of an object and the moments that occur in the engine [18].

Based on a volumetric study of various popular bionic hands of a group of scientists from Yale University [19], the following conclusions can be drawn::

1. in most prostheses, DC motors are used;
2. approximately half of the prostheses have a worm gear to change the direction of the axis of rotation of the engine;
3. most prostheses have a system for controlling the moment of the engine;

4. the vast majority of prostheses provide the so-called “power grip”, that is, the girth of the subject with five fingers for a tight hold;

5. most prostheses have 2 phalanges in each finger.

At the same time, the traditional positional drive control often does not allow you to accurately take items. A positioning error can result in either too much compression of the subject, or insufficient compression. Adding feedback to the engine torque system will more accurately determine the desired position of the grip [20]. It should be noted that force-moment control can be organized without complex intelligent systems based on conventional inexpensive electrical and electromechanical components.

3.4 Decision Making System

Decision making system - a system that selects, based on data from other systems, how best to capture. It also corrects the actions of the user, and when receiving a command for automatic capture, it independently controls the actions for its implementation. Similar systems are found in most modern technical devices, although devices with the connection of individual systems directly, without centralization, are also used [21].

The system under consideration receives data from the human-machine interface and from the vision system, and also exchanges data with the control system at a low level. To ensure the correct operation of the entire prosthesis, the decision-making system must have a margin of speed relative to other systems, as well as be able to correctly adjust the prosthesis's work if any module fails, or, if it is impossible, to implement a safe shutdown.

3.5 Design Features of the Intellectual Prosthesis

Currently, there are many prostheses: traction, bionic, electromechanical, etc. Undoubtedly, their designs differ from each other, but slightly, while maintaining common features [22]. The introduction of intelligent systems into the prosthesis will also lead to a number of changes due to the need to find a place to place the components that appeared and due to certain requirements for the location and relative position of the components themselves.

For the system in question, the design shown in Figure 1 is proposed.

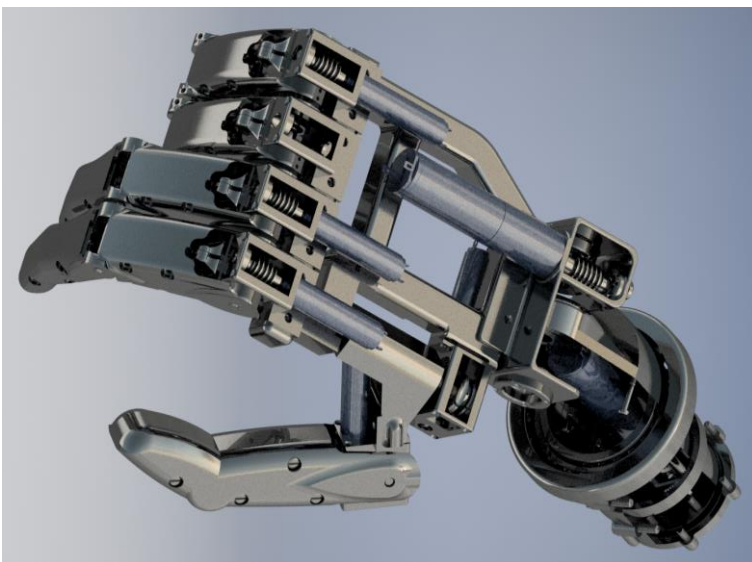


Fig. 1. Proposed prosthesis design without intelligent system components.

The prosthesis arm can rotate 360° around the axis of the forearm, and also has a rotation axis in the wrist. The distal and middle phalanges of the fingers have a rigid mount at an angle of 20°, and the middle phalanges of the fingers 1-4 have a rigid mount that connects them with the palm and ensures their movement when the proximal phalanges move. The thumb has two degrees of freedom, allowing it to change position relative to the palm and set in motion the middle phalanx.

This design is distinguished by:

1. minimization of the number of drives;
2. the possibility of arranging the control electronics and the required sensor system;
3. the use of worm gears to protect motors from overloads [23].

4 Application of neural network technology for the prosthesis technical vision system

Based on the required functions of the TVS of the intelligent robotic prosthesis of the hand (IRPH) described above, the requirements for the system and the possibilities for their implementation, we consider several potential modes of operation (Fig. 2):

1. The main mode of operation. Recognition of an object from a camera image, its classification by means of TVS IRPH, with successful classification - determination of additional parameters unique to each class of objects..
2. Standby time. The absence of any active TVS actions in the absence of a recognition command. Waiting for this command. Energy saving due to system shutdown.
3. The mode of checking the item. Verification of the correctness and quality of recognition of a specific object using TVS tools with the possibility of external control. It is required for the initial and current debugging of the system, as well as for independent verification by the user of an object in case of difficulties with its recognition.
4. Class check mode. Verification of recognition of the class and parameters of the object with the subsequent transfer of data about the stages of verification to an external device. It is required for debugging the system, as well as for generating a user report on incorrect recognition. It serves as the basis for a possible in the future function of remote update of TVS IRPH software.

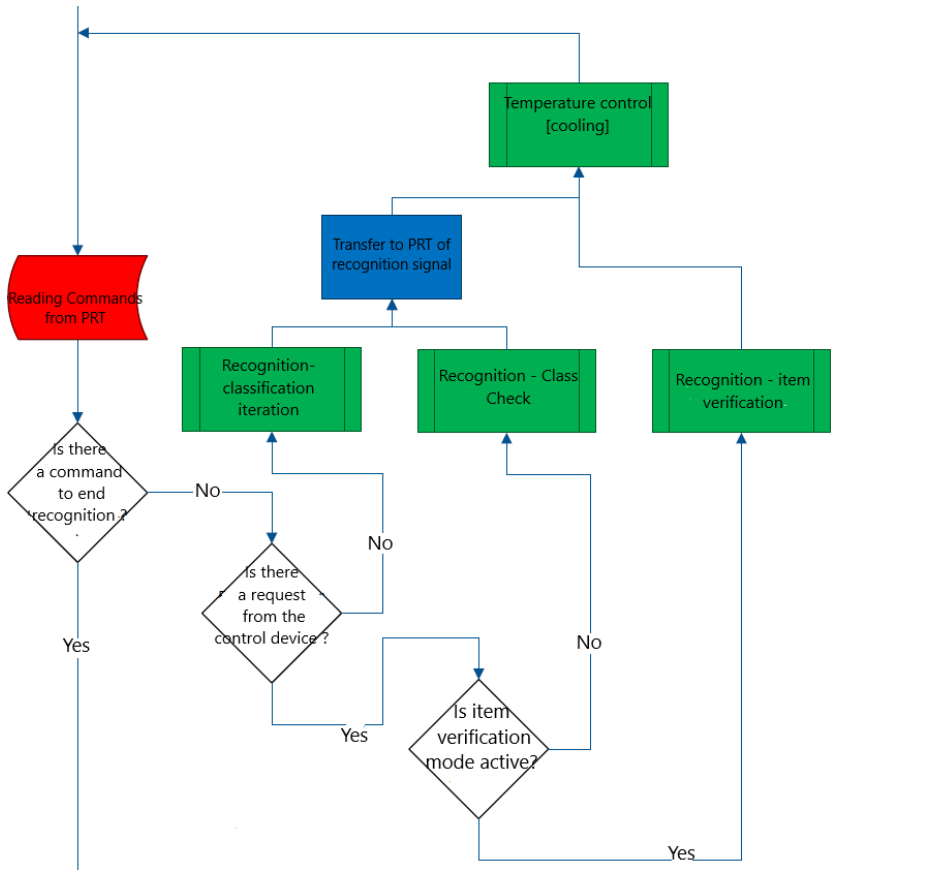


Fig. 2. Selecting an operating mode - a fragment of the main program.

In general, the classification problem has 2 fundamentally different approaches. This is a classification using recognition algorithms and classification using artificial neural networks.

Let us consider in more detail the classification using neural network technology. Today, neural networks are gaining more and more popularity. They found their application in tasks of various kinds. For example, pattern recognition, their classification, tasks of predicting any processes that are random in nature, approximation of functions, etc.

At a qualitatively high level, the problem of image classification began to be solved in 2013, breaking through the barrier of 15% of the classification errors of thousands of types of objects on the ImageNet data set. Since then, a lot of different models of neural networks have been designed and trained, and a 5% error barrier has been broken. The most successful of them are: VGG16, ResNet50, Inception, GoogLeNet, YOLO, MobileNet and many others. Most of them are based on convolutional neural networks.

Most neural networks used to classify images have similar operating principles. These are convolutional neural networks. Their layers in the image consist of a set of different convolution filters. Each of the filters is responsible for finding a specific pattern, and when it finds a certain part of the image in which there is this pattern, the filter sends a signal to the next layer. In turn, the signals of the previous layer make up a new image for the next layer. Building cascades of convolutional layers and training the model, we obtain layers containing abstractions of images. The first layers may contain small parts - lines. Next are

combinations of details - figures. The following layers may already contain forms, and at the end whole objects.

With regard to the development of the IRPH as a learning task, the use of neural networks has several advantages relative to classical methods of image processing:

- the use of ready-made neural networks with a small adjustment for a specific task allows you to reduce time-consuming costs;
- accuracy of object recognition using ready-made neural networks in many cases exceeds the accuracy that can be achieved using classical algorithms;
- in the educational task, it is permissible to use the increased power of computing structures necessary for the operation of a neural network;
- The concept of the TVS IRPH operation allows us to consider the speed of recognition of neural network objects to be sufficient.

5 Experimental check

To confirm the possibility of using neural network technologies for TVS IRPH, an experimental check was carried out.

Using the MobileNet artificial neural network (ANN) on the Raspberry Pi 3B +, the following results were achieved (Fig. 3):

- 1) $t_{set}=80.1^{\circ}\text{C}$; CPU: $87 \pm 2\%$
- 2) FPS: 0.4...1.9
- 3) Recognition accuracy: high

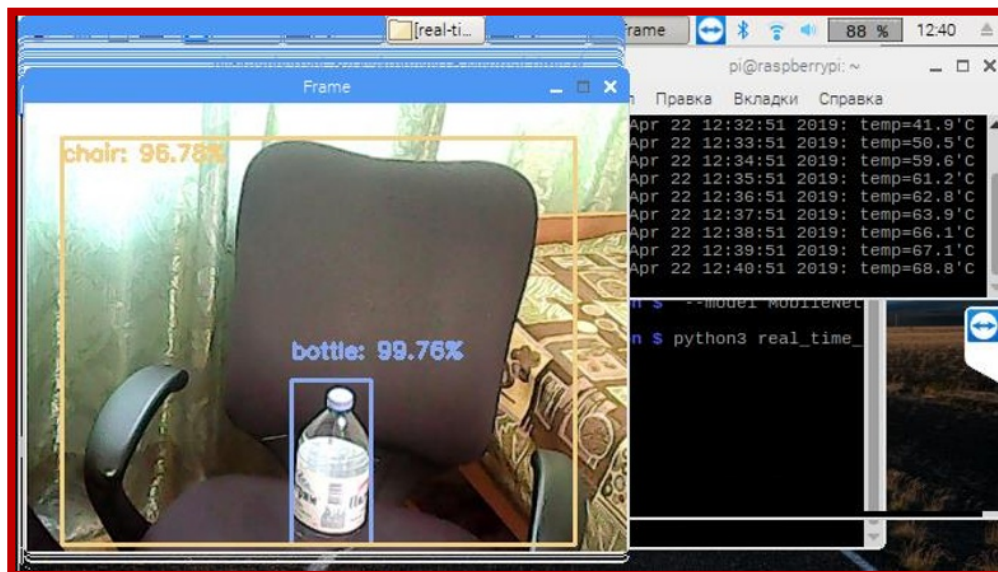


Fig. 3. Experimental verification of the MobileNet neural network on the Raspberry Pi 3B +.

When using the ANN YOLO v3 - tiny on the Raspberry Pi 3B +, the following results were achieved (Fig. 4):

- 1) $t_{set}=80.4^{\circ}\text{C}$; CPU: $86 \pm 1\%$
- 2) FPS: 0.5...1.5
- 3) Recognition accuracy: above average



Fig. 4. Experimental verification of the neural network YOLO v3 - tiny on the Raspberry Pi 3B +.

6 Configuring Raspberry Pi to work with TVS IRPH

One of the main tasks when using neural network technology in training is the correct installation of a ready-made neural network.

The work of a single-board computer as a means of recognizing objects and highlighting their signs requires:

1. connecting the camera;
2. permission to exchange data with external devices via USB B;
3. access to the GPIO input-output ports for controlling the indicator block;
4. The installed openCV library for the operation of a neural network and for determining the characteristics of a class object.

To connect the camera, just enter 2 commands in the terminal: (sudo apt-get update, sudo apt-get dist-upgrade - check for system updates), and then in the Raspberry OS settings (Preferences → Raspberry Pi Configuration) you need to enable the use of the camera - tab Interfaces, item Camera, click on Enabled..

USB and GPIO ports are enabled by default. To use the GPIO ports you need to register:

```
import RPi.GPIO as GPIO
# Setting the number of GPIO pins we will work with
LED = 4
# Reset port states (all configured for input - INPUT)
GPIO.cleanup()
```

Installing openCV on the Raspberry Pi (as well as on almost any linux-like system) is performed by a sequence of commands:

Update the system:

```
$ sudo apt-get update && sudo apt-get upgrade
```

Install the developer tools, including CMake:

```
$ sudo apt-get install build-essential cmake unzip pkg-config
```

Install a set of image and video libraries (they are important for the ability to work with image and video files):

```
$ sudo apt-get install libjpeg-dev libpng-dev libtiff-dev
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
$ sudo apt-get install libxvidcore-dev libx264-dev
```

Install GTK (GUI backend):

```
$ sudo apt-get install libgtk-3-dev
```

Install a package that will reduce GTK warnings:

```
$ sudo apt-get install libcanberra-gtk*
```

Install packages that contain numerical optimizations for OpenCV:

```
$ sudo apt-get install libatlas-base-dev gfortran
```

Set the Python 3 development headers:

```
$ sudo apt-get install python3-dev
```

Download the opencv and opencv_contrib code:

```
$ cd ~
$ wget -O opencv.zip https://github.com/opencv/opencv/archive/4.0.0.zip
$ wget -O opencv_contrib.zip
```

```
https://github.com/opencv/opencv_contrib/archive/4.0.0.zip
```

Unpack the archives and rename the directories::

```
$ unzip opencv.zip
$ unzip opencv_contrib.zip
$ mv opencv-4.0.0 opencv
$ mv opencv_contrib-4.0.0 opencv_contrib
```

Install pip, the Python package manager:

```
$ wget https://bootstrap.pypa.io/get-pip.py
$ sudo python3 get-pip.py
```

Install virtualenv and virtualenvwrapper (they allow you to use virtual Python environments):

```
$ sudo pip install virtualenv virtualenvwrapper
$ sudo rm -rf ~/get-pip.py ~/.cache/pip
```

To complete the installation of these tools, update the `~/.profile` file:

```
$ echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.profile
$ echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.profile
$ echo "export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3" >>
~/.profile
$ echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.profile
$ source ~/.profile
```

Let's create our virtual environment OpenCV 4 + Python 3 on the Raspberry Pi and move on to it:

```
$ mkvirtualenv cv -p python3
$ workon cv
```

Install the Numpy package (OpenCV prerequisite):

```
$ pip install numpy
```

Let's go back to the OpenCV repository and create the directory:

```
$ cd ~/opencv
$ mkdir build
$ cd build
```

Run the CMake instructions to configure the OpenCV 4 build:

```
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
```

```
-D ENABLE_NEON=ON \  
-D ENABLE_VFPV3=ON \  
-D BUILD_TESTS=OFF \  
-D OPENCV_ENABLE_NONFREE=ON \  
-D INSTALL_PYTHON_EXAMPLES=OFF \  
-D BUILD_EXAMPLES=OFF ..
```

Increase the swap space:

```
$ sudo nano /etc/dphys-swapfile  
# EDIT TO: CONF_SWAPSIZE=2048
```

Restart the swap service:

```
$ sudo /etc/init.d/dphys-swapfile stop  
$ sudo /etc/init.d/dphys-swapfile start
```

Compile (try to compile) OpenCV 4:

```
$ make -jx
```

Install OpenCV 4 with two additional commands:

```
$ sudo make install  
$ sudo ldconfig
```

Create a symbolic link from the OpenCV installation at the system level (to the package directory in the virtual environment):

```
$ cd ~/.virtualenvs/cv/lib/python3.5/site-packages/  
$ ln -s /usr/local/python/cv2/python-3.5/cv2.cpython-35m-arm-linux-gnueabi.so  
cv2.so
```

At the same time, the main assembly of the library should take place during the execution of the “make -jx” command, where x is the number of processor cores used in the operation. However, in practice, in the case of the Raspberry Pi 3B +, the command is correctly executed only when using 1 core (which is confirmed by both practical experience and information from various Internet resources). On Raspberry Pi 3B +, testing openCV operability after successful installation of the library showed that the library got up correctly and works.

When using the Raspberry Pi 4 during the installation process, no problems were identified, however, testing to complete the installation showed the configuration to be inoperative. The same is confirmed by numerous examples from online sources. Based on the results of several installation attempts, including in various ways, it was concluded that it was impossible to use openCV on the Raspberry Pi 4 as of November 2019.

Thus, during the preparation of Raspberry Pi versions 3B + and 4, as applied to the task of using as part of TVS IRPH, a choice was made in favor of Raspberry Pi 3B + as it allows to realize the system’s performance at the current moment..

7 Conclusion

The paper proposes the concept of constructing an intelligent prosthesis that allows the capture of objects with a significant increase in the usability of the device and a significant expansion of its capture capabilities. The possibility of constructing such a prosthesis is shown..

The concept of using neural network technology to solve the problem of object recognition by the vision system of an intelligent robotic hand prosthesis is proposed. The experimentally proven possibility of its implementation on a single-board Raspberry Pi 3B + computer is shown.

The use of neural networks in relation to the TVS IRPH can simplify the process of developing the system. Such a variant of their application also opens up wide opportunities for the use of neural network technologies in the training of engineers.

References

1. RSL Steeper web site. Leeds (United Kingdom): RSL Steeper; 2013.: <http://rslsteeper.com/>
2. Touch Bionics. Mansfield (Massachusetts): Touch Bionics Inc; 2013. : <http://www.touchbionics.com/>
3. Thalmic labs, device description Myo armband.: <http://www.myo.com>
4. H2L, device description Unlimited hand.: <http://unlimitedhand.com/en>
5. M. Windrich, M. Grimmer, O. Christ, S. Rinderknecht, P. Beckerle, *Active lower limb prosthetics: a systematic review of design issues and solutions*, Biomedical engineering online, 15(3), p. 140 (2016).
6. K. Schorger, J. P. Simon, D. Clark, A. Williams, *Pneumatic hand prosthesis project (Doctoral dissertation)* (2018).
7. M. Controzzi, F. Clemente, D. Barone, A. Ghionzoli, C. Cipriani, *The SSSA-MyHand: a dexterous lightweight myoelectric hand prosthesis*, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25(5), pp. 459-468 (2017).
8. D. A. Bennett, S. A. Dalley, D. Truex, M. Goldfarb, *A multigrasp hand prosthesis for providing precision and conformal grasps*, IEEE/ASME Transactions on Mechatronics, 20(4), pp. 1697-1704 (2015).
9. R.A.R. C. Gopura, D.S.V. Bandara, *A Hand Prosthesis with an Under-Actuated and Self-Adaptive Finger Mechanism. Engineering*, 10(07), p. 448 (2018).
10. O. Christ, P. Beckerle, *Towards active lower limb prosthetic systems: design issues and solutions* (2016).
11. P. Geethanjali, *Myoelectric control of prosthetic hands: state-of-the-art review*, Medical Devices (Auckland, NZ), 9, p. 247 (2016).
12. D. Brunelli, A.M. Tadesse, B. Vodermayr, M. Nowak, C. Castellini, *Low-cost wearable multichannel surface EMG acquisition for prosthetic hand control*, 2015 6th international workshop on advances in sensors and interfaces (IWASI), IEEE, (pp. 94-99) (2015)
13. N.V. Turushev, M.G. Grigoriev, D.K. Avdeeva, *Bioelectrical nanosensors: Prosthesis interface appliance perspectives*, 2014 the 4th International Workshop on Computer Science and Engineering - Summer. Dubai: Science and Engineering Institute, 2014.
14. C. Varytimidis, K. Rapantzikos, Y. Avrithis, S. Kollias, *Dithering-based Sampling and Weighted α -shapes for Local Feature Detection*, IPSJ Transactions on Computer Vision and Applications, 7, pp. 189-200 (2015).
15. J. DeGol, A. Akhtar, B. Manja, T. Bretl, *Automatic grasp selection using a camera in a hand prosthesis*, In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, pp. 431-434 (2016)
16. M. Ambai, T. Kimura, C. Sakai, *Fast and Accurate Object Detection Based on Binary Co-occurrence Features*, Information and Media Technologies, 10(3), pp. 464-467 (2015).
17. Z. Cai, Q. Fan, R. S. Feris, N. Vasconcelos, *A unified multi-scale deep convolutional neural network for fast object detection*, European conference on computer vision, Springer, Cham, pp. 354-370 (2016).
18. Q. Xu, *Robust impedance control of a compliant microgripper for high-speed position/force regulation*, IEEE transactions on industrial electronics, 62(2), pp. 1201-1209 (2015).

19. J.T. Belter, J.L. Segil, A.M. Dollar, R.F. Weir, *Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review* – <https://www.rehab.research.va.gov/jour/2013/505/page599.html>
20. V. Serebrennyj, A. Boshlyakov, A. Ogorodnik, *Current Control in the Drives of Dexterous Robot Grippers*, International Conference on Interactive Collaborative Robotics, Springer, Cham, pp. 239-248 (2018).
21. M. Marzband, N. Parhizi, M. Savaghebi, J. M. Guerrero, *Distributed smart decision-making for a multimicrogrid system based on a hierarchical interactive architecture*, IEEE Transactions on Energy Conversion, 31(2), pp. 637-648 (2016).
22. M.L. Handford, M. Srinivasan, *Robotic lower limb prosthesis design through simultaneous computer optimizations of human and prosthesis costs*, Scientific reports, 6, 19983 (2016).
23. P. Slade, A. Akhtar, M. Nguyen, T. Bretl, *Tact: Design and performance of an open-source, affordable, myoelectric prosthetic hand*, 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 6451-6456 (2015).
24. A. Boshlyakov, G. Poyarkov, P. Kotov, A. Ermakov, *Intelligent hand prosthesis with enhanced object capture*, AIP Conference Proceedings 2171, 190001 (2019).