Comparative analysis of libraries for computer vision
OpenCV and AForge.NET for use in gesture recognition system

I V Kiselev

Department of business-informatics and IT, Nosov Magnitogorsk State Technical University, 38, Lenin Ave., Magnitogorsk, 455000, Russian Federation
glonken@yandex.ru

Abstract. The article presents the results of the experiment, which based on a comparative analysis of two libraries for solving problems in the field of computer vision in intelligent gesture recognition system: OpenCV and AForge.NET. For the test was developed by intelligent gesture recognition system at both libraries and a block diagram of the gesture recognition system has been developed. A recognition algorithm based on neural network technologies has been developed, suitable sample descriptors and classifier structure have been selected to achieve the best results. The effectiveness of the gesture recognition test software was also evaluated. Below is a brief description of both libraries, a more detailed description of the research and test results.

1. Introduction
Gesture recognition is one of the important task in such areas as digital image processing, computer vision, development of intelligent security systems and access control. It consists in identifying one or more gestures by comparing the input images with the existing images in the database [2].

To create an intelligent system for gesture recognition, there are ready-made open-source libraries: OpenCV, Caffe, AForge.NET, Torch, Theano and others. Now we will stop out attention at the two most popular libraries, OpenCV and AForge.NET and carry out a comparative analysis of these two libraries.

2. Library AForge.NET
AForge.NET is a library written in C/C++. There are opportunities for solving problems in the field of artificial intelligence in the library AForge.NET. The range of tools used by the library is quite diverse: image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics and more. There is an html-documentation for this library, which may be necessary for novice developers [6].

The main components of the library AForge.NET presented below:

- AForge.Imaging: a library designed to work with images and filters;
- AForge.Vision: a library that uses computer vision techniques;
- AForge.Video: a package of libraries for performing work related to video data;
- AForge.Neuro: a library that uses the capabilities of neural networks;
- AForge.Genetic: a library designed to solve a variety of problems using genetic algorithms;
- AForge.Fuzzy: a library, which works with fuzzy logic;
- AForge.Robotics: a library, which supports methods used in the field of robotics;
• AForge.MachineLearning: a library that uses machine-learning elements.

Over time AForge.NET became part of the Accord.NET Framework, which is a relatively new software platform, entirely created in C#. Library Accord.NET uses tools and features, which previously were not in AForge.NET: the use of vector machines, projective discriminant analysis and support hidden Markov models [1], application of new methods of training neural networks, new image filters and other opportunities. Accord.NET Framework can be used in applications Microsoft Windows, Xamarin, Unity3D, Windows Store, Linux, and mobile devices [3].

Library Accord.NET Framework can be divided into several major groups:

• Accord.Vision: real-time face detection and tracking, and common methods for detecting, tracking, and converting objects into image streams. Contains cascade definitions, Camshift trackers, and Dynamic Template Matching. Includes pre-created classifiers for human faces and some facial features such as noses;
• Accord.Imaging: contains point of interest detectors (e.g. Harris, SURF, FAST, and FREAK), image filters, image matching and image docking techniques, and feature extractors such as oriented gradient histograms and Haralick descriptive functions;
• Accord.Controls.Vision: Windows Forms components and controls for tracking head, face, and hand movements and other computer vision-related tasks;
• Accord.Controls.Imaging: Windows Forms controls for displaying and processing images. Contains a convenient ImageBox, which can control mimics of the traditional MessageBox to quickly display or check images;
• Accord.MachineLearning: the application of support vector machines, decision trees, naive Bayesian models, Gaussian Mix models, and General algorithms such as Ransac, cross-validation, and Grid-Search for machine learning applications;
• Accord.Math: contains a library of matrix extensions, along with a set of numerical matrix decomposition methods, numerical optimization algorithms for bounded and unbounded problems, special functions, and other tools for scientific applications;
• Accord.Statistics: contains probability distributions, hypothesis testing, statistical models, and methods such as linear and logistic regression, hidden Markov models, random field method, principal component method, fractional least squares regression, linear discriminant analysis, and many others.

3. OpenCV Library

OpenCV (Open Source Computer Vision Library) is a computer vision library written in a high-level language (C/C++). It is also available for other languages such as Python, Java, Ruby, Matlab, Lua, etc [7]. The Library can be used on various operating systems, including Linux, Mac OS X, iOS, Android, etc. Contains algorithms for: image interpretation, camera calibration by reference, optical distortion elimination, similarity detection, object movement analysis, object shape detection and object tracking, 3D reconstruction, object segmentation, gesture recognition, etc [4].

The OpenCV library is capable of using multicore processors. In case you need automatic optimization on various hardware platforms Intel, you can purchase additional library IPP, with English Integrate Performance Primitives [8]. This library includes procedures with low-level optimization, which can be used for a variety of algorithmic areas.

This library is very popular due to its openness and the possibility of free use for both educational and commercial purposes, as well as due to the large number of documentation in different languages to work with the library.

The main modules of the library are the following:

• opencv_core: core, basic structures, calculations (mathematical functions, pseudorandom number generation, DFT, DCT, XML input/output, etc.);
• opencv_imgproc: image processing (filters, transformations, etc.);
• opencv_highgui: simple UI, loading/saving images and videos;
• opencv_ml: machine learning methods and models (svms, decision trees, etc.);
• opencv_features2d: various descriptors (SURF);
• opencv_video-motion: analysis and object tracking (optical flow, motion patterns, background elimination);
• opencv_objdetect: detection of objects in the image (Haar wavelets, HOG, etc.);
• opencv_calib3d: camera calibration, stereo matching search and 3D data processing elements;
• opencv_flann: the fast library search nearest neighbors (FLANN);
• opencv_contrib: companion code that is not yet ready for use;
• opencv_legacy: obsolete code, preserved for backward compatibility;
• opencv_gpu: acceleration of some OpenCV functions by CUDA (NVidia).

4. Implementation of gesture recognition algorithm

Using computer image processing, it is possible to isolate individual parts in an image, which will make it possible to control the manipulator, but there is a problem that is limited to the area of emulation of the manipulator. In this case, the palm of a person acts as a computer "mouse", which must perform commands clearly and immediately. There is also the problem of the distance at which control is permissible and within which it is possible to recognize an object as the palm of a person. In the developed application, it is possible to use several types of cameras, the main purpose of the function is the ability to control the "mouse" from a sufficiently large distance, which in turn will increase the number of areas of use of the software. In order to identify problem areas and determine the basic algorithm for error-free hand recognition, several works of domestic and foreign researchers were considered.

First, you need to refine the methods of gesture recognition using the available modules in the libraries. It is very easy to add your own code to OpenCV, which simplifies the adaptation of the library to the task [10]. AForge.NET is a part of the framework Accord.NET but there is no recognition of the desired gesture in this framework, which took more time to adapt the framework for specific tasks [5].

Now it is necessary to train the neural network to recognize gestures [9]. Modules for creating neural networks are present in both libraries. Testing carried out with four different gestures (Figure 1). To create a training sample, were taken 2000 photos, 500 images for each gesture. The sample contains photos of gestures from different angles, with different lighting and changing backgrounds. The size of all images is 480×360. Training of classifiers took place on a computer with an Intel Core i5 processor with a frequency of 2.4 GHz and 8GB of RAM.

(a) – the first gesture
(b) – the second gesture
(c) – the third gesture
(d) – the fourth gesture

Figure 1. The proposed four gestures.
It is worth noting that training classifiers using Haar cascades would take much longer. It is necessary to take into account heuristic methods that will affect the quality of gesture recognition, as artificial intelligence algorithms and computer depend on various heuristic parameters: for example, the selection of the number of neurons in the hidden layer or setting the level of false alarms when training the classifier.

5. **Comparative testing of libraries**

First, we tested the training sample using OpenCV. Testing took place on a set of 1000 photos with different backgrounds and lighting. Elements on the main diagonal of the matrix characterize the number of correctly classified gestures, all non-zero numbers outside the main diagonal show the number of erroneous choices. "Predicted" is the gesture to which the program assigned the sample and "Original" - what it is actually refers. The confusion matrix for the results testing on a test sample with OpenCV is given in **Table 1**.

### Table 1. Test’s results on OpenCV

| Predicted | 1  | 2  | 3  | 4  |
|-----------|----|----|----|----|
| Gesture   |    |    |    |    |
| 1         | 243| 1  | 1  | 2  |
| 2         | 0  | 247| 2  | 1  |
| 3         | 2  | 2  | 246| 6  |
| 4         | 5  | 0  | 1  | 241|

The percentage of incorrectly recognized images is 2.3%.

Now we test for the framework AForge.NET. Test conditions are the same. The confusion matrix for the results testing on a test sample with AForge.NET is given in **Table 2**.

### Table 2. Test’s results on AForge.NET

| Predicted | 1  | 2  | 3  | 4  |
|-----------|----|----|----|----|
| Gesture   |    |    |    |    |
| 1         | 245| 1  | 1  | 0  |
| 2         | 2  | 244| 6  | 2  |
| 3         | 1  | 2  | 242| 0  |
| 4         | 2  | 3  | 1  | 248|

The percentage of incorrectly recognized images is 2.1%.

The lesser the number of errors is obtained by detection of the "open palm" (Figure 1 (d)). This could be explained by the fact that for the "open palm" the number of selected features during training is greater than for the rest.
6. Conclusion

As a result of numerical experiments and tests, it was found that both libraries give good results in the recognition of the proposed four gestures after learning and changing the initial code for the task. OpenCV, because of its’ structure, can be useful as a basis for further development, while AForge.NET has a full framework Accord.NET that simplifies the work with simple projects and integrations, but is less adapted for further development. The developed algorithm is universal because it can be supplement with a database of noisy images to increase the accuracy of recognition in complex images and video sequences.

References

[1] Yang J, Xu Y 1994 Hidden Markov Model for Gesture Recognition Technical Report CMURI-TR-94-10 (Pittsburgh, PA: Carnegie Mellon Univ., Robotics Institute).

[2] Eickeler S, Kosmala A, Rigoll G 1998 Hidden Markov model based continuous online gesture recognition, In Proc. Fourteenth Int. Conf. on Pattern Recognition. Vol. 2. pp. 1206–1208.

[3] Lv F, Nevatia R 2006 Recognition and segmentation of 3-D human action using HMM and multi-class adaBoost, In Proc. 9th Eur. Conf. on Computer Vision (Berlin; Heidelberg: Springer-Verlag) iss. 4, pp. 359–372.

[4] Wang X, Xia M, Cai H, Gao Y, Cattani C 2012 Hidden-Markov-models-based dynamic hand gesture recognition Math. Problems Eng 2012 986134-1–986134–11.

[5] Nam Y, Wohn K 1996 Recognition of space-time hand-gestures using hidden Markov model, In Proc. ACM Symposium on Virtual Reality Software and Technology (Hong Kong) pp. 51–58.

[6] Accord.NET 2019 URL: http://code.google.com/p/accord/.

[7] OpenCV 2019 URL: https://opencv.org/.

[8] Cherkasov K V, Gavrilova I V, Chernova E V, Dokolin A S 2018 The use of open and machine vision technologies for development of gesture recognition intelligent systems Journal of Physics Conference Series 1015(3) 032166.

[9] Gavrilova I V 2018 Concept of automated system of Russian sign language recognition», Perspective development of science, technology and technology 52-54

[10] Cherkasov K V, Gavrilova I V 2018 Selection of image recognition technology in intelligent gesture recognition system Perspective development of science, technology and technology 54-57