Wrocław University
of Science and Technology

Faculty of Electronics

Embedded Robotics

Intermediate project

Recognizing letters of American Sign Language Alphabet

Author:
Radoslaw Zwolski

Supervisor:
Witold Paluszyński, Ph.D

Abstract

Goal of the project was to create an application that recognizes letters of American Sign Language (ASL). Input of the application are images registered by camera. Recognized letters are being written on the screen of the device in runtime.

Assumptions were made that the application will be run in environment with sufficient amount of light to properly illuminate hand of the user (acquiring hand image is made by color recognition). Also, background behind the user should be sufficiently different from skin color in order to not be taken into consideration during image preprocessing. Commitment related to amount of letters that can be recognized was made. It should recognize at least 3 letters.

Wrocław 2018

1 Introduction

Goal of the project was to create an application that recognizes letters of American Sign Language (ASL). Input of the application are images registered by camera. Recognized letters are being written on the screen of the device in runtime.

Assumptions were made that the application will run in environment with sufficient amount of light to properly illuminate hand of the user (acquiring hand image is performed by color recognition). Also, background behind the user should be sufficiently different from skin color in order to not be taken into consideration during image preprocessing.

Application that has been finished and which is described in this article is able to recognize 4 letters of American Sign Language alphabet (A, B, C and D). In fact, the application is able to recognize any static hand gesture when it is provided with appropriate teaching data.

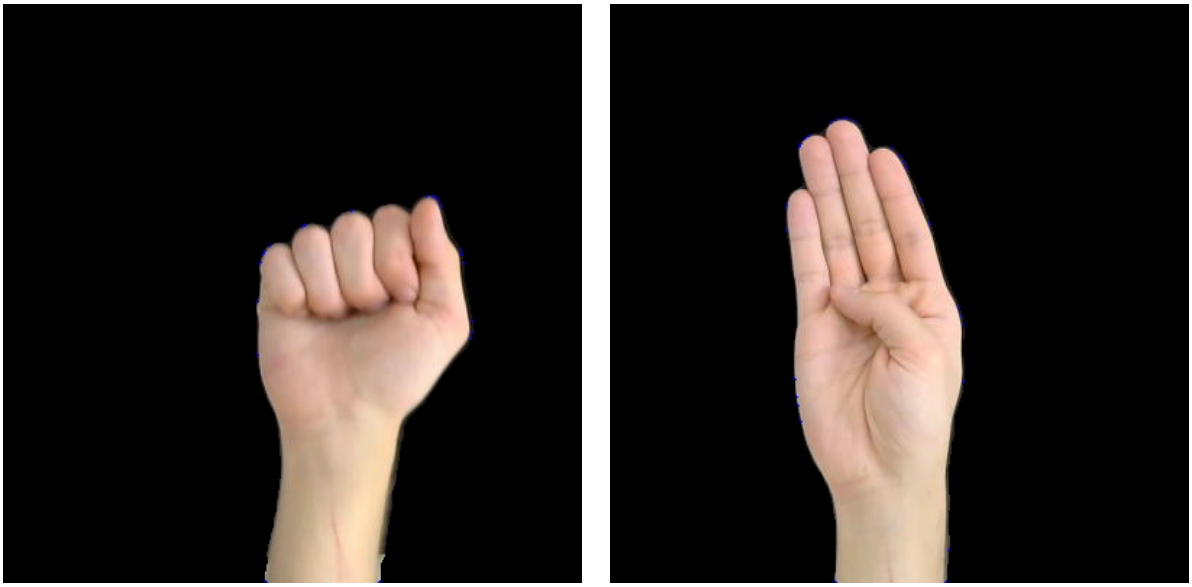
2 Background

The project is an introduction to a larger piece of work which will be the author's Master's final project and has been done to deepen knowledge and experience in subjects of digital processing of images and machine learning. More advanced and enhanced applications like that can be used as translators for people that do not know sign language and also can be found in use of social robotics.

Latest version of the application requires that a person whose hand gesture is being recognized has to have their's sleeve folded as the neural network used for classiffication has been taught with photos where not only hand can be visible but also arm until elbow. Examplary photos from collected teaching set can be seen on Figure 1.

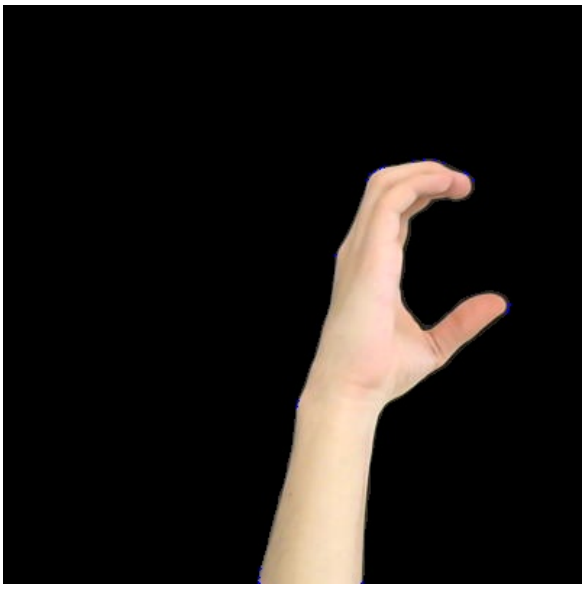
In summary, 2400 photos of hands from 3 people have been taken to create initial dataset (200 photos of every person's hand for every letter that have been mentioned previously).

Figure 1: Examplary samples of letters from teaching set



(a) A

(b) B



(c) C

(d) D

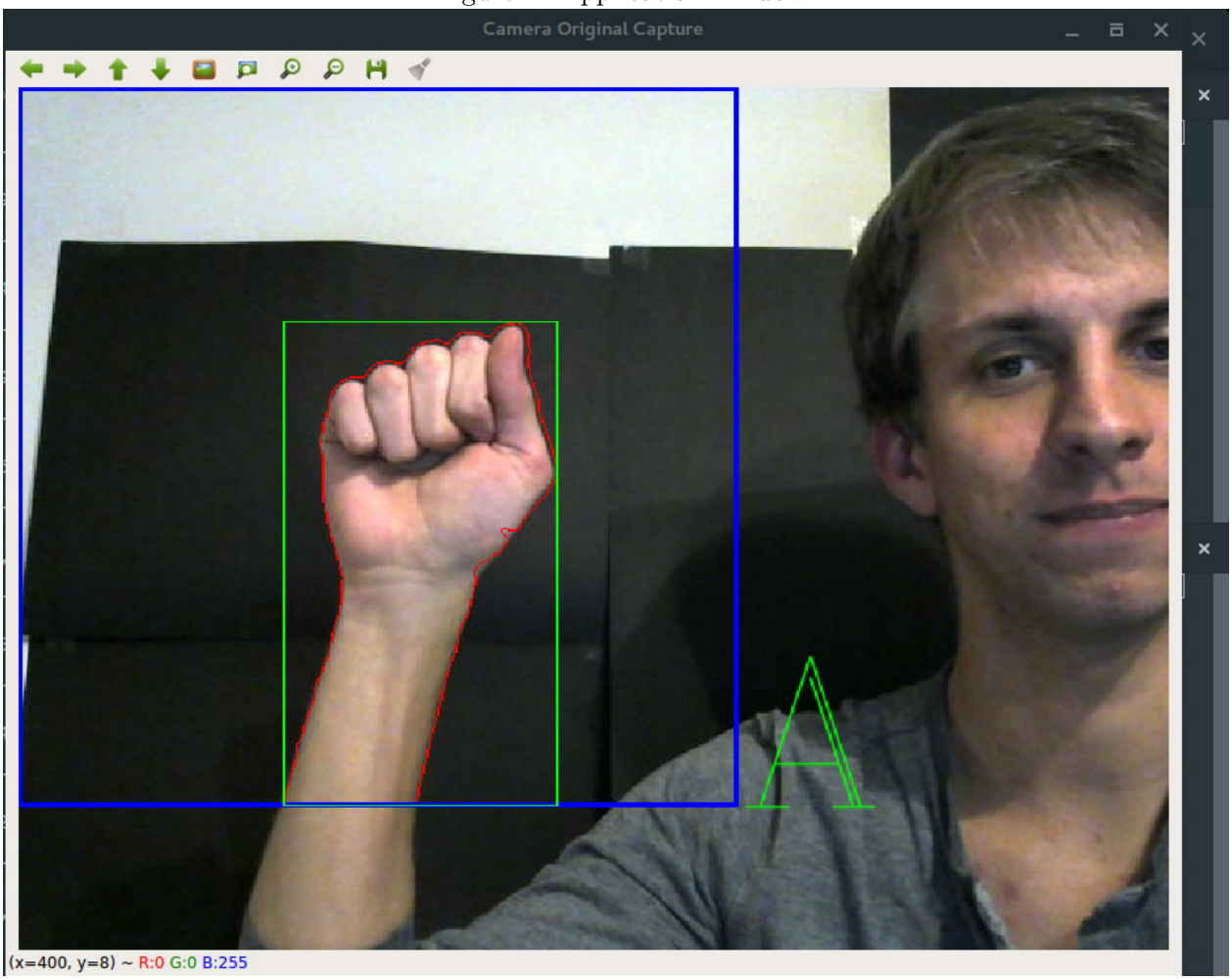
3 Implementation

Whole application was implemented in C++14[1] with use of OpenCV[2] library which provides open source computer vision functionalities that makes it easier to handle image processing. Also from OpenCV ver. 3.0. it is possible to use sufficiently working and written as object oriented entities machine learning algorithms as it consist of: SVM, MLP, Decision Trees, Random Trees, K-Nearest Neighbours, etc. Those algorithms are implemented in namespace *cv::ml*.

View of the application can be seen in Figure 2.

[1] read more about the programming language a[1]
[2] learn more about the library at [2]

Figure 2: Application window



3.1 Classifier model

For classification of sign language letters *cv::ml::MLP* from OpenCV library has been used.

Of course to use any machine learning model preprocessing of image is needed as putting coloured RGB image on the output of neural network would make teaching of network very hard from the point of time complexity. Furthermore, neural network which has been taught with coloured images would be not be robust and changes of lighting conditions would influence correctness of classification.

3.2 Preprocessing of the image

First the image is being cut to extract a square image which is stretched from top-left corner and which side is 400 pixels long. Application requires that user will place right hand inside this square. The square can be seen in Figure 2 along with whole application window.

3.2.1 Extraction of hand image

Hand is being extracted from the square image described in the beginning of subsection 3.2. The image is being converted from RGB to YCrCb format. It allows extraction of hand by by a particular interval of colours. Described application creates a mask image which contains pixels of 0 and 1 values. Pixel

will have value 1 in mask if it is in color range between **[0, 133, 77]** YCbCr and **[255, 173, 127]**[3] YCrCb.

Mask, which is result of such extraction is being applied onto original image resulting in image that can be seen in Figure 1.

Based on the mask it is possible to find contours of hand and bounding rectangle of the hand. Contours are red lines around the hand in Figure 2 and bounding rectangle is green rectangle around hand.

Again, image is cut so it shows only a hand without any padding around it. Image that is used for further processing is shown in Figure 3.

Figure 3: Result image of hand extraction

3.2.2 Feature extraction

Feature extraction has to be made to make training of neural network faster and to make whole classification task more resitant to changes of input image. Thanks to that part of image processing application will not depend on hand of the user i.e. size of hand, width or length of fingers, colour of skin will not influence recognition of gestures.

Extracted hand image is converted into gray scale. Then Sobel operator is applied and after that thresholding is performed[4]. Such processing results in image of edges of the hand. Edges are indicated by white pixels. This image's pixel values are only 0 and 1. Thresholded image is resized to decrease time of neural network's training.

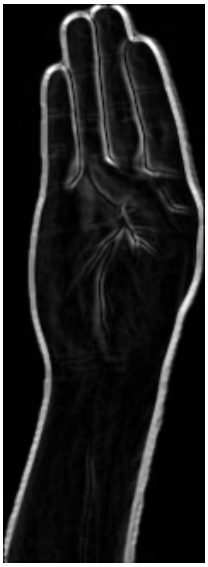
Images which are results of particular preprocessing operators and filters are show in order in Figure 4.

[3]Such colour range has been found in article which is pointed in References as [3]. According to that article this is the best colour range for skin description

[4]More informations about image processing algorithms and operators at [4]

Figure 4: Examplary samples of letters from teaching set

(a) hand image after hand extraction



(b) hand image after conversion to gray scale and applying Sobel opeator

(c) hand image after applying threshold

(d) hand image after resizing

4 Summary

To see the code of application please contact the author of this report.

4.1 Conclusions

It is crucial to perform proper image preprocessing and feature extraction in order to make such program to work. It will be susceptible for error in cases when there will be two gestures looking very similarly. The hardest thing to do in order to make such application working properly is choosing a proper classification model and training algorithm as well as its parameters.

During testing of application different people were showing their hands. Thanks to those tests it was possible to find and correct the problems related to extraction of image of the hand. Some people were very pale and in good lighting image captured by camera shows as if their skin was white like paper in some places. Extraction of hand has been fixed to extract hand as area inside its contour.

4.2 Further development and improvement

There is a lot of room for improvement as the application still does not recognize letters properly. It does recognize letters "B", "C" and "D", though it still has some problems with recognition of "A" as it is not resistant for changes of angle. A little twist of wrist can change output of the program.

Improvement can be made by changing the way of feature extraction as putting all of pixels of edges onto neural network is not very robust and such approach can provide a lot of errors. Instead convex hull points and defects could be extracted and for example distances between those points would be features that could be used as input of the network. In such case teaching of the network would be immensly shortened. Such approach is described in [5].

To exclude the problem which is connected with extraction of hand due to colour mismatch, it can be fixed by using convolutional neural networks and deep learning instead of MLP. Then the program would be able to recognize any number of hands and in any environment where its hand shape could be seen.

References

[1] C++ reference. [Online]. Available: http://en.cppreference.com/w/

[2] OpenCV: OpenCV modules. [Online]. Available: https://docs.opencv.org/3.3.0/

[3] Y. Chen, B. Luo, Y. L. Chen, G. Liang, and X. Wu, "A real-time dynamic hand gesture recognition system using kinect sensor," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2015, pp. 2026–2030.

[4] Metody przetwarzania obrazów z wykorzystaniem biblioteki opencv. [Online]. Available: http://www.focus.agh.edu.pl/theses/MGR04.pdf

[5] H. A. Jalab and H. K. Omer, "Human computer interface using hand gesture recognition based on neural network," in *2015 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW)*, Feb 2015, pp. 1–6.