A CNN-based Sign Language recognizer using Motion History Image

Intelligent Control final report

Prof. Han-Pang Huang

Group 5

王韻涵
王隨賢
馮　笛

June 2017

Outline

1. Introduction

The purpose of the project was to create a method for the automatic recognition of expression of Sign Language (SL) in order to facilitate and promote the communication between hearing and deaf people in everyday life.

Our method combines Motion History Image technique with Deep Learning obtaining a high testing accuracy even using low quality video inputs.

In the second paragraph we introduce SL and some previous research done in recognition of SL's movements. In the third paragraph we discuss our original approach to the problem and in the fourth paragraph results are presented.

2. Sign Language and previous work in recognition

SL is a system of communication that, opposingly to spoken languages, uses hand shapes, movement of the hands, arms, body, and facial expressions as language fundaments. Currently it has been estimated that approximately 70 millions women and men all over the world communicate with some SL's variant as first language or mother tongue[1].

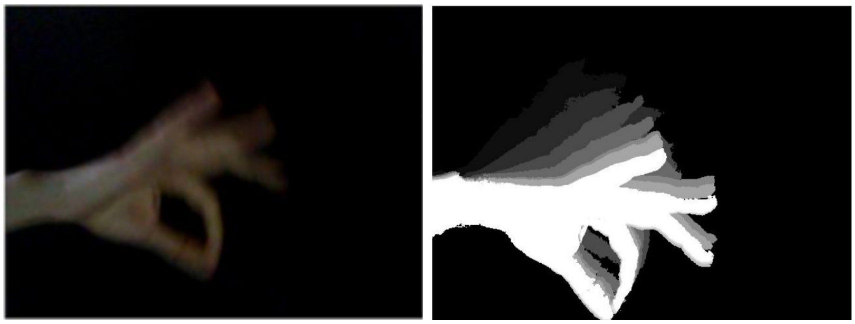
It is important not to confuse the sign language with body language, since this merely represents a form of non-linguistic communication. Between the 137 SLs currently recognised[2] we chose to consider the American SL (ASL), spoken by more than 500 thousands signers and which is largely used in many applications in literature.

[1] World Federation of the Deaf (WFD), *Sign Language*, wfdeaf.org, (July 2017)

[2] Lewis; Simons; Fennig. (2013) *Ethnologue: Languages of the World* (17th ed.)

Many researchers had focused on sign language recognition. Some[3] [4] [5] [6] recognize signs with the help of wearable devices sometimes called “smart gloves”, reaching fairly high recognition accuracy. This kind of approach to the problem however, requires the installation of a hardware that could result expensive for the user or which calibration could be complex from signer to signer.

Another kind of approach is based on the utilization of images coming from a common camera. Özge et alii[7] utilize Motion History Image (MHI) to transform a sequence of frames (from the videos of signs) in a single image, and use this image to classify the sign using nearest neighbor. Unfortunately it seems to consider the restricted cases in which the sign can completely be expressed using simple hands positions. In addition, the videos considered are exclusively taken with clean, black background, suggesting that the system might not be robust.



A sign is recorded and from the videos’ frames (left) a single MHI image is obtained (right) [6]

[3] Abhinandan Das et al., *Smart glove for Sign Language communications*. International Conference on Accessibility to Digital World (ICADW), 2016, Guwahati, India.

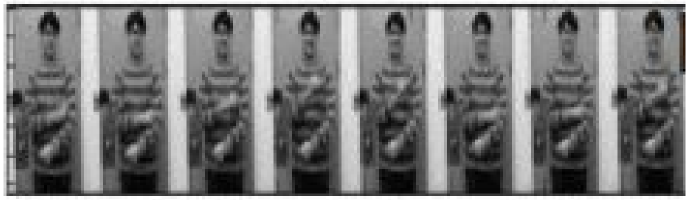
[4] K Abhijith Bhaskaran et al., *Smart gloves for hand gesture recognition: Sign language to speech conversion system*. International Conference on Robotics and Automation for Humanitarian Applications (RAHA), 2016. Amritapuri, Kollam, Kerala, India.

[5] Aadarsh K. Singh et al., *A low-cost wearable Indian sign language interpretation system*. International Conference on Robotics and Automation for Humanitarian Applications (RAHA), 2016. Amritapuri, Kollam, Kerala, India.

[6] Jian Wu, Lu Sun and Roozbeh Jafari. 25 August 2016. *A Wearable System for Recognizing American Sign Language in Real-Time Using IMU and Surface EMG Sensors*. IEEE Journal of Biomedical and Health Informatics. Volume 20, Issue 5, Sept. 2016.

[7] Özge Yalçınkaya, Anıl Atvar and Pınar Duygulu. *Turkish sign language recognition application using Motion History Image*. 24th Signal Processing and Communication Application Conference (SIU), 2016. Zonguldak, Turkey.

Recently, an interesting approach has been presented by Ji et alii[8], concatenating frames from the signs videos in long, horizontal images. These images are then used to train a Convolutional Neural Network that once trained is able to classify the sign from the sampled image. Although they obtain high recognition accuracy during testing, the method could be computationally expensive and might not be used for fast real time recognition.



Example of input image of the CNN in [8]

Other methods base the recognition on the skeleton movements read by devices such as kinect[9]. However, as already mentioned, we retain that the use of additional hardware might compromise the portability and scalability of a SL recognizer.

3. Method

Our approach to the problem doesn't require any hardware installation on the SL speaker. The only necessary hardware is a camera, with possibly a minimum resolution of 100x100, in order to record the sign that is being made by the signer. These simple requirements allow us to be able, in future, to integrate this method in a smartphone app, spreading the utilization of the recognizer.

[8] Yangho Ji, Sunmok Kim, Ki-Baek Lee. *Sign Language Learning System with Image Sampling and Convolutional Neural Network*. IEEE International Conference on Robotic Computing (IRC). April 2017, Taichung, Taiwan.

[9] Yuqian Chen and Wenhui Zhang. *Research and implementation of sign language recognition method based on Kinect*. 2nd IEEE International Conference on Computer and Communications (ICCC), 2016. Chengdu, China.

The video coming from the camera, is transformed in an MHI of 100x100 px, which is then input in a previously trained CNN-based neural network, that classifies the images as one of the labels from the training data.

Following, we introduce the data retrieving process, as well as the production of the MHI starting from the raw recorded videos of the signs. Finally we will describe the structure of the Neural Network built.

3.1 Data retrieving

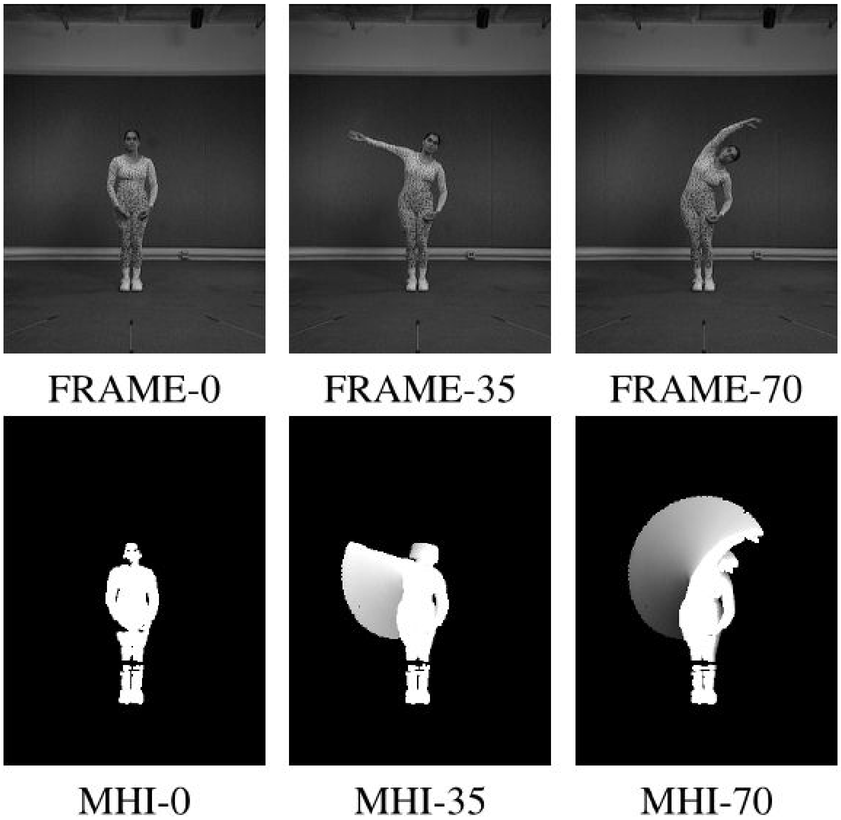
After attempting to use different existing SL datasets[10], we decided to personally record videos of 14 words from ASL. As a matter of fact, the available open source datasets contain not enough data to train a Neural Network.

Each individual recorded 10 times the same movement for each word, obtaining 140 videos (of approximately 5 sec) for each signer. The original videos have then been processed (view 3.2 for details) producing a total of 1307 MH Images.

3.2 Image processing and production of MHI

An MHI is obtained comparing the pixels value between each couple of consecutives frames of the same video. If the value of the pixel differs more than a threshold value $\boldsymbol{\tau}$, the corrispettive pixel in the MHI becomes “active” and it is set to white. A pixel’s value will gradually translate to black every time is not active for a couple of consecutive frames. This results in a grayscale image of the “motion history” of the video, where the grayer the pixel is the earlier in the video a movement was detected.

[10] American Sign Language Lexicon Video Dataset (ASLLVD)

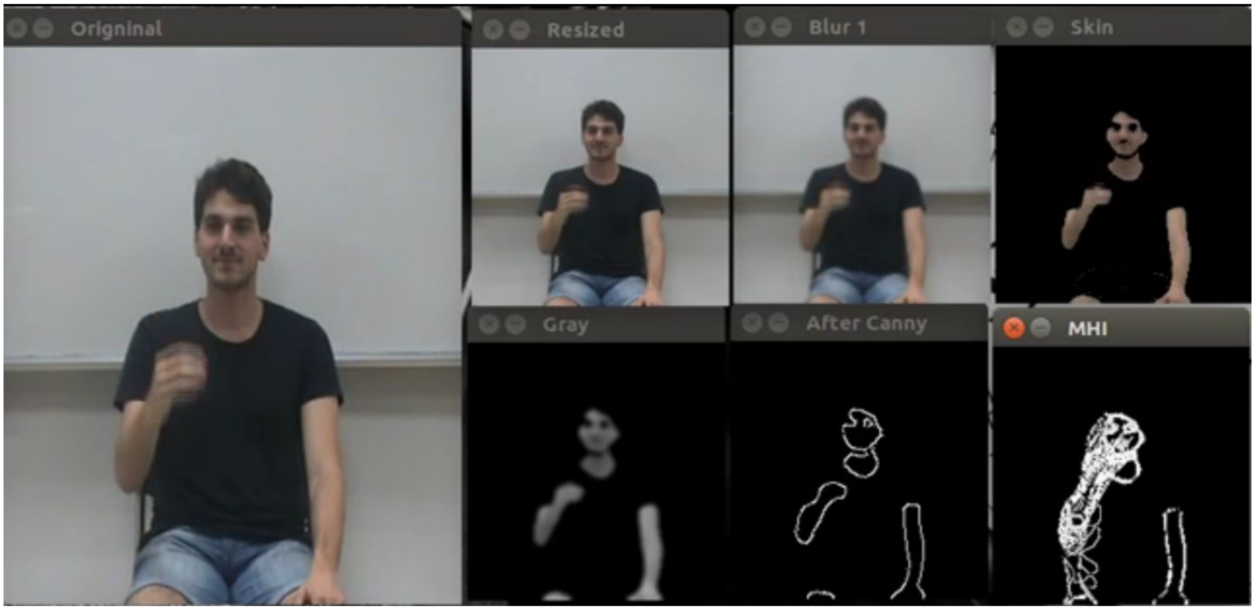


Example of a MHI for ballet movement

Since the quality of the frames can be very different from video to video, before being given to the MHI maker algorithm, each frame go through some processing step:

1. The frame is resized from its original dimension to 100*100 pixels, reducing the computational complexity and the processing time of the MHI making process.
2. Successively, a blurring filter with kernel size 5 is applied to the frame. This allows the reduction of noises signals from the frame, getting a more clean image.
3. After blurring, all the pixels (expressed in RGB values) of the frame which are not some shade of the human skin color are set to black value. After this process the remaining part of the original image is the one that contains face, arms and hands of the signer. We believe, in fact, that significant information of the motion rely in these components.

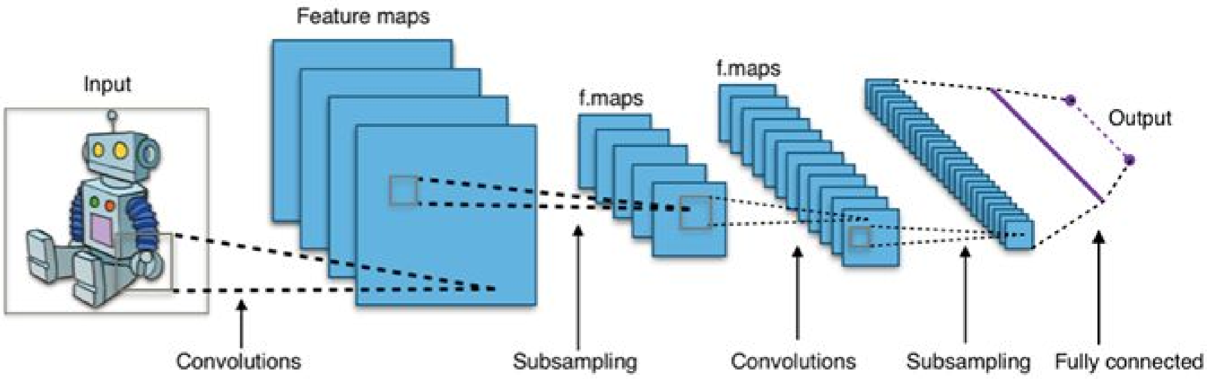
4. The RGB frame is then flatten in a grayscale image.
5. At last, a Canny filter is applied to the gray frame, obtaining a clear image of the edges of the arms, face and hands of the signer.



An example of the image processes applied to a single frame

3.3 Convolution neural network

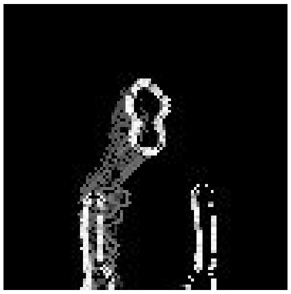
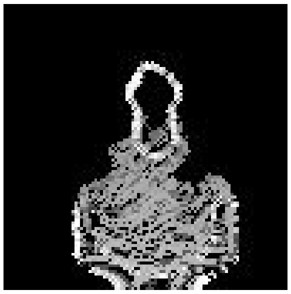
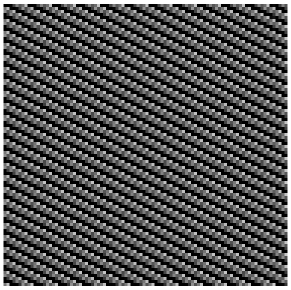
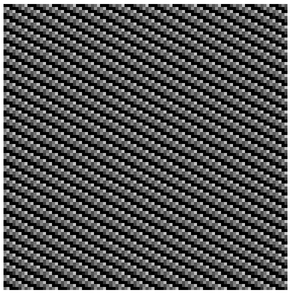
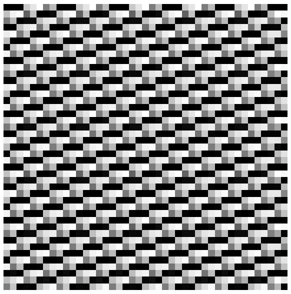
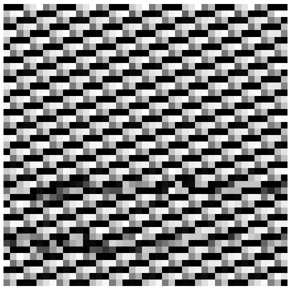
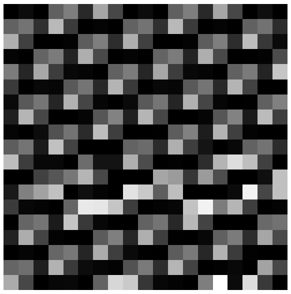
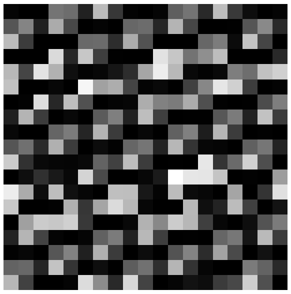
It is known that a convolutional neural networks are composed by a series of alternated layers of convolution and max pooling followed by a more common, fully connected neurons layer.



As the information passes through the layers, we can extract the features which become more and more abstract. This idea can help us to decide the number of convolution layers. The following table provides the details of our network.

Layer	Filter	Output size
Input		100*100
Convolutional	15*(7*7)	(94*94)*15
Max pooling	(2*2)	(47*47)*15
Convolutional	30*(5*5)	(43*43)*30
Max pooling	(2*2)	(21*21)*30
Convolutional	50*(3*3)	(19*19)*50
Max pooling	(2*2)	(9*9)*50
Flatten		4050
Hidden		70
Output		14

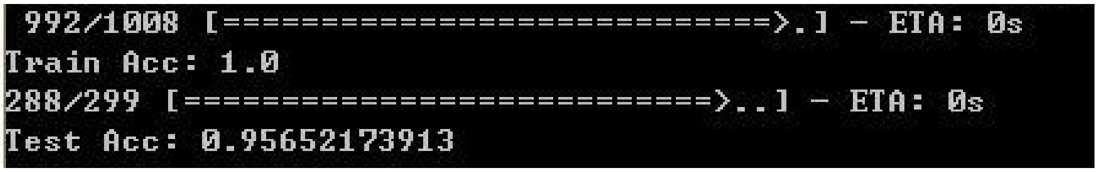
Although we have decided the numbers of layers in the convolutional neural network, we should decide the filters size in each layer. We can determine it by checking the output from each layer. The following table shows the output.

	“Beautiful”	“Work”
Input image		
Output of first convolution layer		
Output of second convolutional layer		
Output of third convolution layer		

It is clear that there is a huge different in the output from the second convolutional layer although we can not tell the difference in the output of the one. As the result, we checked the output in each layer for each input image. If we couldn't distinguish the difference, we added neurons to that layer.

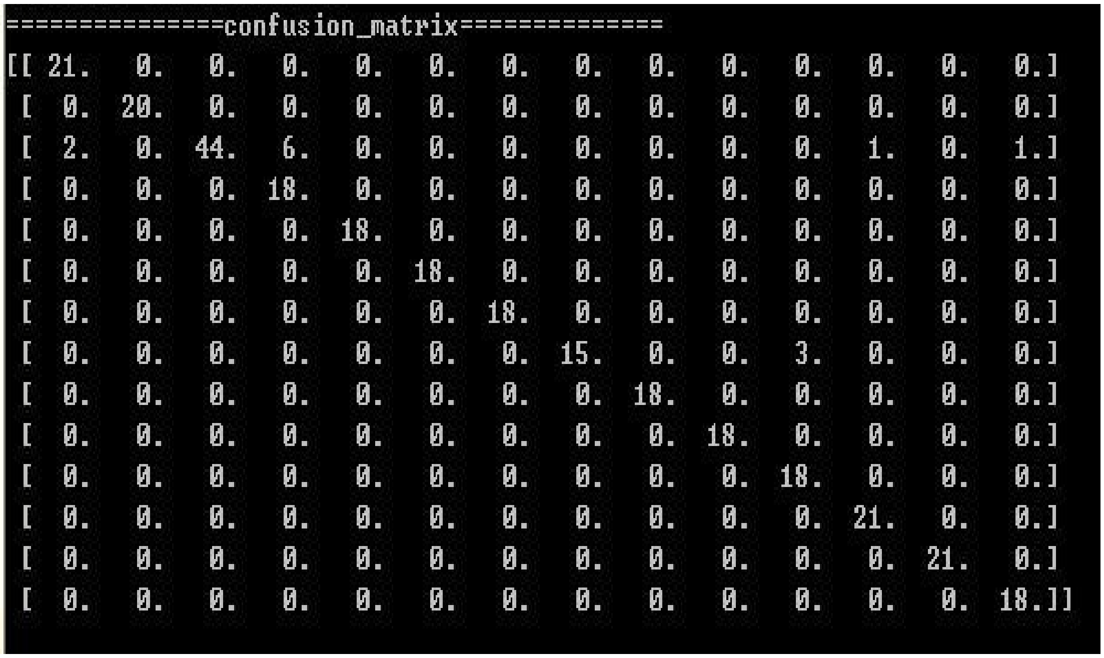
4. Results and discussion

There are 1008 images used as training set and 299 for testing. For training, we chose 10 as batch size and 15 as epochs. After training was over, the accuracy for training set was 100% while the testing one was above 95%.



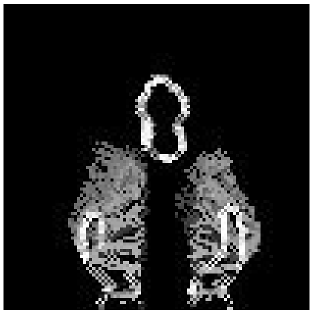
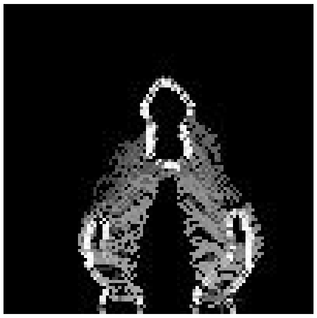
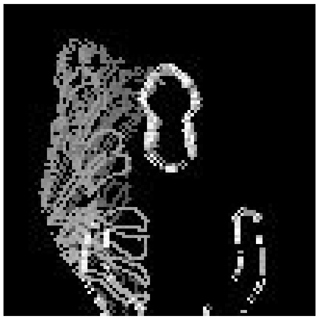
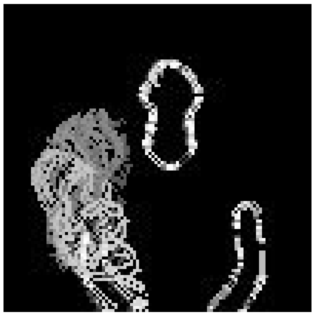
```
 992/1008 [=============================>.] - ETA: 0s
Train Acc: 1.0
288/299 [============================>..] - ETA: 0s
Test Acc: 0.95652173913
```

Following, the confusion matrix of the testing data.



```
==============confusion_matrix=============
[[ 21.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  0.  20.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  2.   0.  44.   6.   0.   0.   0.   0.   0.   0.   0.   1.   0.   1.]
 [  0.   0.   0.  18.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0.  18.   0.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0.   0.  18.   0.   0.   0.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.  18.   0.   0.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.  15.   0.   0.   3.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.  18.   0.   0.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0.  18.   0.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  18.   0.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  21.   0.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  21.   0.]
 [  0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  18.]]
```

It is shown that 11% of the error comes from mis-recognizing the word "control" as "friend" while another 16% comes from mis-recognizing the word “hello” as “my”. A confrontation between these two couples of images is following given.

	
control	friend
	
hello	my

As one can see, the MHI image for the word “control” appears similar with the one for “friend”. To avoid the confusion between these two words, it may be helpful to add some new feature except MHI image. On the other hand, “hello” appears in fact different with “my”.

We can guess that the reason might be the small amount of data available for training and testing. As the result, our training and testing set can not represent all the situations. So, if there is some MHI image which is not similar with the image in training set, the convolution may misrecognize it. Therefore, adding additional training data may increase the accuracy and the generality of the SL recognizer.