Handwriting recognition for Mouse Scanner

By

Tran Duc Muoi, Vu Thi Hai Yen
Computer Vision, 2014
Dr. Le Thanh Ha

HANDWRITING RECOGNITION FOR MOUSE SCANNER

Computer Vision Assignment, 2014

Instructor: Dr. Le Thanh Ha

Team members:

Tran Duc Muoi	11020204	K56CA
Vu Thi Hai Yen	11020409	K56CA

Abstract

The objective of this report is giving both overview and discussion about our assignment in computer vision course: “Handwriting recognition for mouse scanner”. We were given a LG mouse scanner and our task is developing an application for it. In this report, we will discuss about the mouse scanner itself, our problems statement and our solution. How we constructed and implemented is also mentioned. In 6 weeks from 3 November, we research about the topic above and implement our application systematically. By the time the assignment ends at 15 December, our application can run at the acceptable level.

Acknowledgement

This work is supported much by our instructor, Dr. Le Thanh Ha. We also want to give special thanks to our classmates, K56CA for their help in data gathering phase. We notice that some pieces of open source code and libraries are used in our application as well.

CONTENTS

Handwriting recognition for Mouse Scanner 1

1 Project description 3

1.1 Requirements 3

1.2 Mouse scanner 3

1.2.1 Mouse scanner itself [1] 3

1.2.2 LG Smart scan 3

1.3 About handwriting recognition [2] 4

1.3.1 Definiton 4

1.3.2 Off-line recognition 4

2 Implementation 5

2.1 Algorithms 5

2.1.1 Image extraction 5

2.1.2 Artifical neural networks 5

2.2 Tool and libraries 6

2.2.1 Tool 6

2.2.2 Libraries 6

2.3 input data 6

3 REsult 6

4 Project plans 7

4.1 Period 7

4.2 Working plan 7

4.3 Schedule task 7

5 References 8

6 Appendix A – List of figures 8

1 PROJECT DESCRIPTION

1.1 REQUIREMENTS

- Build an application for mouse scanner
- Application can run at an acceptable level

1.2 MOUSE SCANNER

1.2.1 MOUSE SCANNER ITSELF [1]

LG Mouse Scanner LSM-100 is the world's first mouse with an embedded scanner function that offers convenience, mobility, and simplicity to users. Now users can perform OCR editing of scanned files as well as sending them as e-mails, SNSs and mobile phones. From the offices of global companies to daily life, the LG Mouse Scanner will transform scanning into a much easier and more convenient task.

MOUSE	
Sensor	Laser Sensor, 1,200 DPI
SCANNER	
DPI	Adjustable up to 320 (320/200/100)
Scan Size	Any size up to A3
Pixel Size	640x300 pixels @ 30Hz
MAIN FUNCTIONS	
Main Menu	Image & Text Paste, Share, Save, Print, Edit
Save Format	JPEG/TIFF/PNG/BMP/XLS/DOC/PDF
Application Programs	M/S Office, Adobe Photoshop, Google Translate...
Text Recognition	OCR Function Supported

Table 1 - Some specifications of LG LSM-100

1.2.2 LG SMART SCAN

Figure 1- LG Smart Scan software

LG Mouse Scanner LSM-100 has an attach software named LG Smart Scan. An open source code of this software can be downloaded from driver official website of LG.

This is a powerful software, which can support many functions such as editing image (auto rotation, contrast control, brightness control, etc.); pasting image or text; link to MS Word, Evernote, Google Translate, etc. and OCR function.

Overall, this software is almost sufficient for user to use the mouse scanner except the OCR function. Although it supports over 100 different languages, it can recognize only three of them in a

document. Other week point is that it can recognize only printed character. Therefore, we decide to build an application to overcome this problem: handwriting recognition.

1.3 ABOUT HANDWRITING RECOGNITION [2]

1.3.1 DEFINITON

Handwriting recognition (or HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (optical character recognition) or intelligent word recognition. Alternatively, the movements of the pen tip may be sensed "on line", for example by a pen-based computer screen surface.

Handwriting recognition principally entails optical character recognition. However, a complete handwriting recognition system also handles formatting, performs correct segmentation into characters and finds the most plausible words.

Off-line handwriting recognition involves the automatic conversion of text in an image into letter codes, which are usable within computer and text-processing applications.

On-line handwriting recognition involves the automatic conversion of text as it is written on a special digitizer or PDA, where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching. This kind of data is known as digital ink and can be regarded as a digital representation of handwriting. The obtained signal is converted into letter codes, which are usable within computer and text-processing applications.

Because image comes from mouse scanner is a static image, we will discuss further about the off-line handwriting recognition.

1.3.2 OFF-LINE RECOGNITION

The data obtained by the mentioned conversion is regarded as a static representation of handwriting. Off-line handwriting recognition is comparatively difficult, as different people have different handwriting styles. In addition, as of today, OCR engines are primarily focused on machine printed text and ICR for hand "printed" (written in capital letters) text. There is no OCR/ICR engine that supports handwriting recognition as of today.

Off-line recognition divided in to two phases: Character extraction and character recognition. There are two frequently used algorithms in character recognition phase. One is artificial neural networks, which will be implemented in our application, and another is feature extraction.

2 IMPLEMENTATION

2.1 ALGORITHMS

2.1.1 IMAGE EXTRACTION

Our image extract algorithms generally are:

- From static image, extract each character into a blob (a rectangle surround the character)
- Scale blob into binary 20x20-pixel grid.
- Assign value 0 or 1 to each element of grid, based on its intensity (>200 or <200)
- Fill the grid with value 0

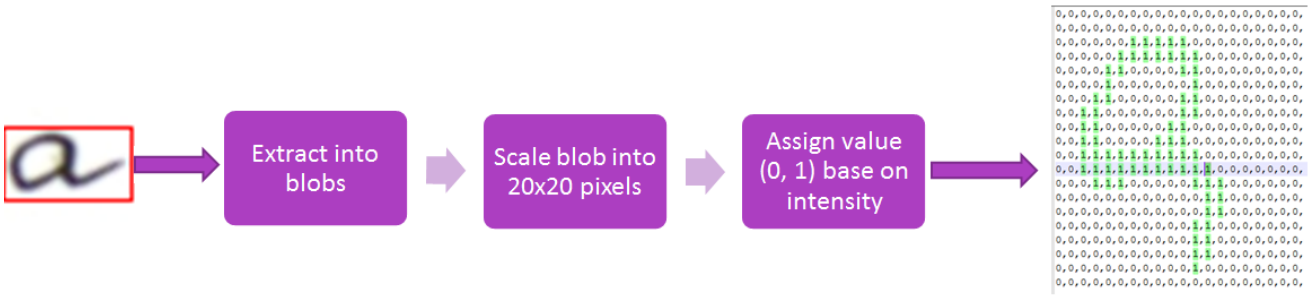

Figure 2 - Image extraction

2.1.2 ARTIFICAL NEURAL NETWORKS

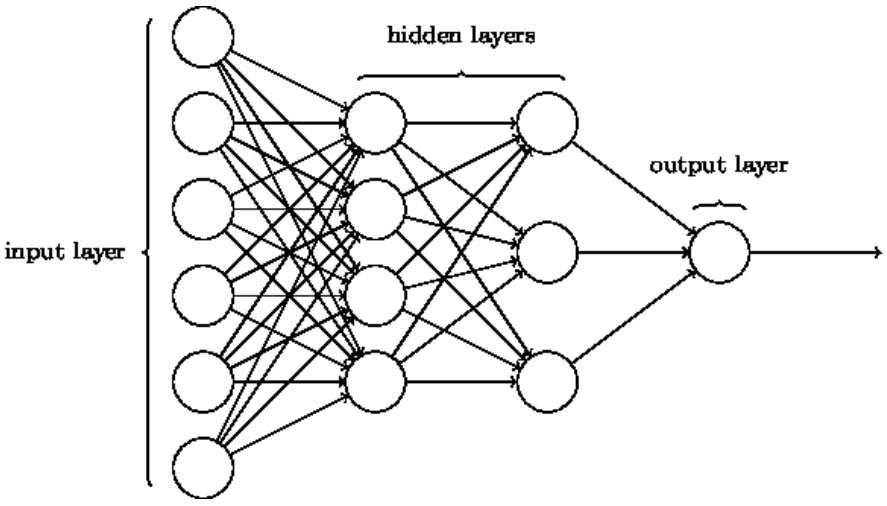

Figure 3 - Example of ANN design [3]

We construct ANN as follow:

- 400 input neural (400 pixel of each blob)
- 1 hidden layer with 64 neural
- 26 output neural (26 letters)
- Sigmoid function in hidden layer and output layer
- Train by Resilient Propagation algorithm

This architecture inspired by an article [3] by Michael Nielsen in December 2014.

2.2 TOOL AND LIBRARIES

2.2.1 TOOL

In this project, we used Microsoft Visual Studio 2013 to implement application in C# and Microsoft Paint for editing input data.

2.2.2 LIBRARIES

2.2.2.1 AForge.net

AForge.NET [4] is a C# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence - image processing, neural networks, genetic algorithms, machine learning, robotics, etc. In our project, we used AForge.net in extraction character into blobs.

2.2.2.2 Encog Machine Learning Framework

Figure 4 - Encog Logo

Encog [5] is an advanced machine-learning framework that supports a variety of advanced algorithms, as well as support classes to normalize and process data. Machine learning algorithms such as Support Vector Machines, Artificial Neural Networks, Genetic Programming, Bayesian Networks, Hidden Markov Models, Genetic Programming and Genetic Algorithms are supported. Most Encog training algorithms are multi-threaded and scale well to multicore hardware. Encog can also make use of a GPU to further speed processing time. A GUI based workbench is also provided to help model and train machine learning algorithms. Encog has been in active development since 2008 and available for Java, .Net, and C++.

In our project, we used Encog framework to create, train the ANN and recognize the test set. However, we have to transform the data structures to fit the Encog framework.

2.3 INPUT DATA

We asked our teammates to write down nearly 300 characters for each person. For each character, we rotated it for 10 and -10 degree. So every our friend did provide approximately a thousand samples. Totally, our input data has 15583 samples of English handwriting lowercase characters.

3 RESULT

Following is a demonstration of our application. It reaches the accurate level of over 70%. State it with our expectation, we find it is acceptable. Moreover, we leave some improvements as the future works such as improving the ANN structure, refining the data input and try another feature extraction.

For the information, our application, OCR-LearningTool [6], is available on the following link: $https://github.com/hidrodo/OCR-LearningTool$

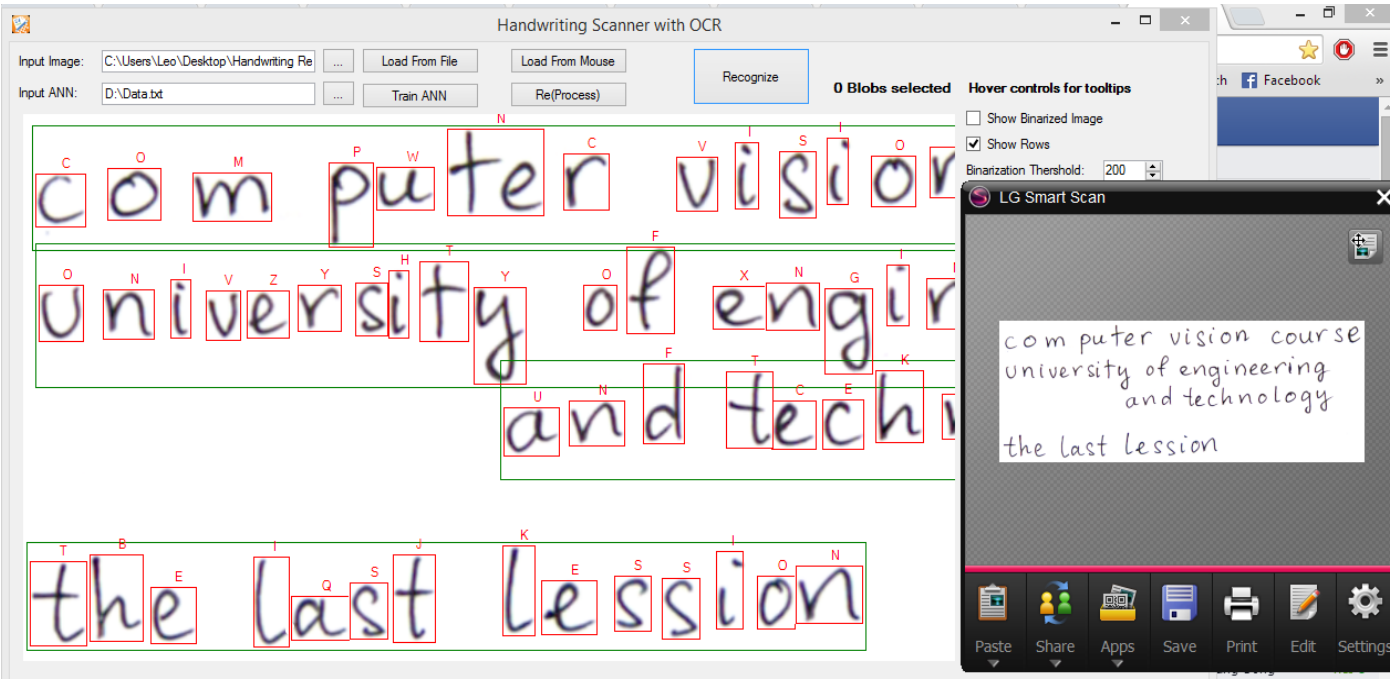

Figure 5 - A demonstration of the application

4 PROJECT PLANS

4.1 PERIOD

Approximate 6 weeks from 3 November to 15 December.

4.2 WORKING PLAN

Task	**Tran Duc Muoi**	**Vu Thi Hai Yen**
Devices Research	LG Smart Scan software, Mouse Scanner	Mouse Scanner
Algorithms	Image extraction, Libraries finder	Artificial Neural Network
Implementation	All the coding works, data gathering	Data gathering
Documentation	PowerPoint slide, report	Report

4.3 SCHEDULE TASK

Task	Week					
	1	2	3	4	5	6
Research on devices and attached software						
Brainstorm ideas						
Algorithms decision and research						
Implementation						
Documentation						

5 REFERENCES

[1] "LG LSM 100 Specification," [Online]. Available: http://www.lg.com/us/computer-accessories/lg-LSM-100-mouse-scanner.

[2] Wikipedia, "Wikipedia," [Online]. Available: http://en.wikipedia.org/wiki/Handwriting_recognition.

[3] M. Nielsen, "neuralnetworksanddeeplearning," [Online]. Available: http://neuralnetworksanddeeplearning.com/chap1.html.

[4] AForge.net. [Online]. Available: http://www.aforgenet.com/.

[5] Encog. [Online]. Available: http://www.heatonresearch.com/encog.

[6] T. D. Muoi, "Github," [Online]. Available: https://github.com/hidrodo/OCR-LearningTool.

6 APPENDIX A – LIST OF FIGURES

Figure 1- LG Smart Scan software 3

Figure 2 - Image extraction 5

Figure 3 - Example of ANN design [3] 5

Figure 4 - Encog Logo 6

Figure 5 - A demonstration of the application 7