

Handwritten Digit Recognition By using libnumrecognition

## **Handwritten Digit Recognition By using libnumrecognition**

Yan Cheng Cheok

[yccheok@yahoo.com](mailto:yccheok@yahoo.com)

<http://sourceforge.net/projects/numrecognition/>

Version 1.1

06 September 2005

Based on libnumrecognition Release 0.01

### **Contents**

#### **1. Getting Started**

- 1.1. About libnumrecognition
- 1.2. About this tutorial
- 1.3. Setting up your Environment

#### **2. Using libnumrecognition in your application**

- 2.1. Recognize single digit
- 2.2. Recognize multiple digit

#### **3. Image feature used in libnumrecognition**

- 3.1. LRTBHV feature

#### **4. UML design of libnumrecognition**

Handwritten Digit Recognition By using libnumrecognition

## 1. Getting Started

### 1.1 About libnumrecognition

libnumrecognition is a Java library created for handwritten digit recognition purpose.

With the default neural network model shipped together with libnumrecognition, the highest recognition rate is 97.87% and the lowest recognition rate is 90.92%.

Currently, libnumrecognition is in the preliminary state. It can only accept images in the noiseless, single/ multiple digit form. However, development work on libnumrecognition is still going on. **libnumrecognition is hoped to be evolved into a complete digit recognition framework, which can accept images with noisy background, multiple objects form.** Only with that capability, we are able to use libnumrecognition to solve real world problem.

### 1.2 About this tutorial

This tutorial is as an introduction for embedding libnumrecognition library on your Java application. It also explain the working mechanism behind libnumrecognition.

### 1.3 Setting up your environment

The following software are required in order to develop your application by using libnumrecognition

- Java 2 Platform, Standard Edition, v 1.4.2 (J2SE) or above for core Java development tool- <http://java.sun.com/j2se/>
- Java Advanced Imaging 1.1.2 (JAI) or above for multiple image I/O format support - <http://java.sun.com/products/java-media/jai/index.jsp>
- Java Object Oriented Neural Engine (JOONE) for neural network support - <http://sourceforge.net/projects/joone>
- ImageJ 1.34 or above for image processing support - <http://rsb.info.nih.gov/ij/>
- libnumrecognition - <http://sourceforge.net/projects/numrecognition/>

For JOONE and ImageJ, they are shipped together with libnumrecognition. For J2SE and JAI, you need to go to SUN website for downloading.

## Handwritten Digit Recognition By using libnumrecognition

After you have obtained all the library, put the following files into your lava CLASSPATH

- libnumrecognition.jar – from libnumrecognition
- joone-engine.jar – from JOONE
- ij.jar – from ImageJ
- jai\_codec.jar – from JAI
- jai\_core.jar – from JAI

Now, everything is done :) You may proceed to chapter 2 to learn how to use libnumrecognition in your application.

## **2. Using libnumrecognition in your application**

### **2.1. Recognize single digit**

To perform digit recognition, some basic knowledge on neural network and image processing. Hence, when libnumrecognition is designed in mind, developer may use it without any prior knowledge in image processing or neural network

The below are the code to perform single digit recognition.

## Handwritten Digit Recognition By using libnumrecognition

```
// Construct a factory to generate image feature.
//
NumberImageFeatureFactory imageFeature = new NumberImageLRTBHVFeatureFactory();

// Construct a neural network for number image recognition purpose.
//
NumberNeuralNetworkRecognizer neuralNetwork = new NumberNeuralNetworkRecognizer(
    // This is the pre-trained neural network by using 10000 hand written
    // number images from http://yann.lecun.com/exdb/mnist/
    // This neural network has 96 input neurons, 200 hidden neurons.
    // It is trained by using learning rate 0.9, momentum 0.1
    // with iteration 2000.
    //
    "LRTBHV-training-data.txt-I=96-H=200-LR=0.9-M=0.1-C=2000-.snet",

    // We will use LRTBHV feature.
    //
    imageFeature,

    // With default feature parameter.
    //
    FeatureParameter.DEFAULT_FEATURE_PARAMETER);

// Try to recognize handwritten number image (1 digit).
//
System.out.println("myimage.bmp"+" is recognized as "+neuralNetwork.recognize
("myimage.bmp"));
```

**Diagram 1 : Example of code for single digit recognition**

This is the brief explanation on what the above codes is doing

- First, we will extract out the image feature from the image. Each digit has its own characteristic. The characteristic, which will distinguish the images from each other, is what we so call “image feature” Currently, there are two features provided by libnumrecognition, “Chain Code” and “Left, Right, Top, Bottom profile and Horizontal, Vertical projection” (LRTBHV)

After several experiments by the author, LRTBHV gives a better performance compared with Chain Code. Also, libnumrecognition only provide one neural network model based on LRTBHV. Hence, user are expecting to use LRTBHV feature all the time until there are better features introduced by next release of libnumrecognition.

- After that, we will construct a neural network object.

## Handwritten Digit Recognition By using libnumrecognition

A neural network, which is trained with 10000 handwritten digit images from <http://yann.lecun.com/exdb/mnist/> train image database, is shipped together with this library as default neural network model.

By performing benchmarking on 8920 handwritten digit images from <http://yann.lecun.com/exdb/mnist/> test image database, the following recognition rate is achieved.

Category	Number of Pass	Number of Fail	Recognition Rate
Zero	873	19	97.87
One	824	68	92.38
Two	826	66	92.6
Three	821	71	92.04
Four	848	44	95.06
Five	832	60	93.27
Six	850	42	95.29
Seven	831	61	93.16
Eight	811	81	90.92
Nine	827	65	92.71

**Diagram 2 : Recognition rate for the default neural network model**

For the feature parameter injected into neural network, *FeatureParameter.DEFAULT\_FEATURE\_PARAMETER* should serve well for most of the purpose.

## Handwritten Digit Recognition By using libnumrecognition

- By supplying image file name into the neural network, we will get the recognition result immediately. All the image format which is supported by Java Advanced Imaging, will be supported also in libnumrecognition.

By using the above code, here is the result of the recognition:

Number Image	Recognition Result
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

**Diagram 3 : Result from the diagram 1 example code**

### 2.2. Recognize multiple digits

The below are the code to perform multiple digits recognition.

**Diagram 4 : Example of code for multiple digits recognition**

## Handwritten Digit Recognition By using libnumrecognition

```
// Construct a factory to generate image feature.
//
NumberImageFeatureFactory imageFeature = new NumberImageLRTBHVFeatureFactory();

// Construct a neural network for number image recognition purpose.
//
NumberNeuralNetworkRecognizer neuralNetwork = new NumberNeuralNetworkRecognizer(
    // This is the pre-trained neural network by using 10000 hand written
    // number images from http://yann.lecun.com/exdb/mnist/
    // This neural network has 96 input neurons, 200 hidden neurons.
    // It is trained by using learning rate 0.9, momentum 0.1
    // with iteration 2000.
    //
    "LRTBHV-training-data.txt-I=96-H=200-LR=0.9-M=0.1-C=2000-.snet",

    // We will use LRTBHV feature.
    //
    imageFeature,

    // With default feature parameter.
    //
    FeatureParameter.DEFAULT_FEATURE_PARAMETER);

// Try to recognize handwritten number image (multiple digits).
//
System.out.println("myimage.bmp is recognized as ");
String[] s = neuralNetwork.recognize("myimage.bmp", 144);
for(int i=0; i<s.length; i++) {
    System.out.println(""+s[i]);
}
```

This is the brief explanation on what the above codes is doing

- Recognition will be done from left most to right most.
- Compared with the single digit recognition code, an extra 144 parameter (int maxXDistanceOfNextDigit) is used. This parameter is used to specific the maximum distance of each digit. If the distance among the digits exist maxXDistanceOfNextDigit, it will be treated as a new number.

By using the above codes, here is the result of the recognition:

Handwritten Digit Recognition By using libnumrecognition

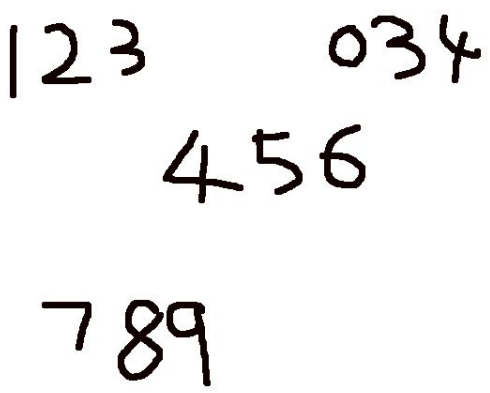
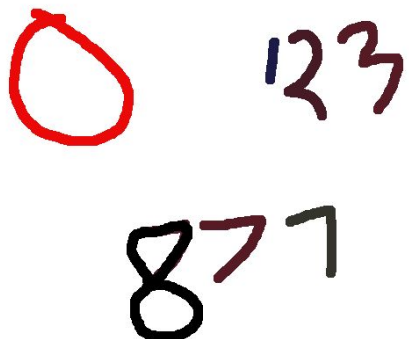
Number Image	Recognition Result
 The image shows four groups of handwritten numbers: '123' and '034' on the top left, '456' in the center, and '789' on the bottom left.	123 789 456 034
 The image shows three groups of handwritten numbers: a red '0' on the left, '123' on the top right, and '877' on the bottom center.	0 877 123

Diagram 5 : Result from the diagram 4 example code

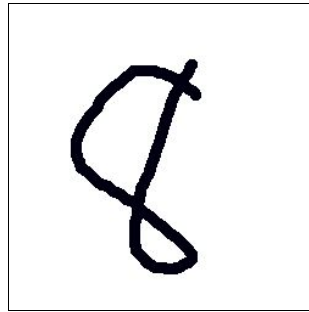
**3. Image feature used in libnumrecognition**

**3.1 LRTBHV feature**

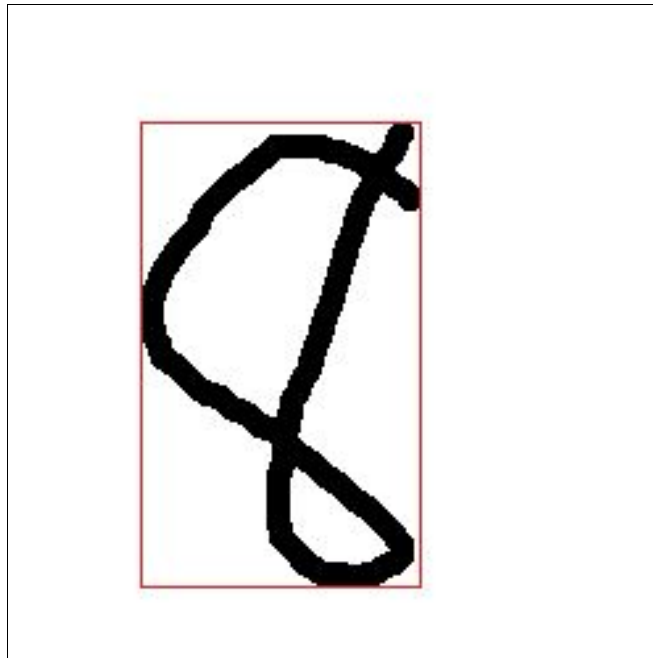
The below diagram insulate how LRTBHV feature are extracted from an image.



Handwritten Digit Recognition By using libnumrecognition

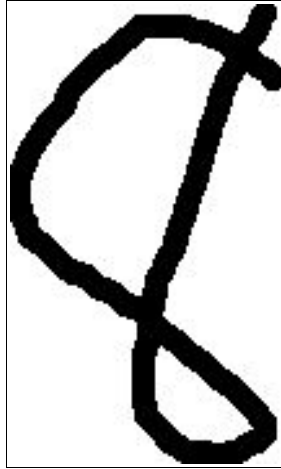


**Diagram 6 : Original image**

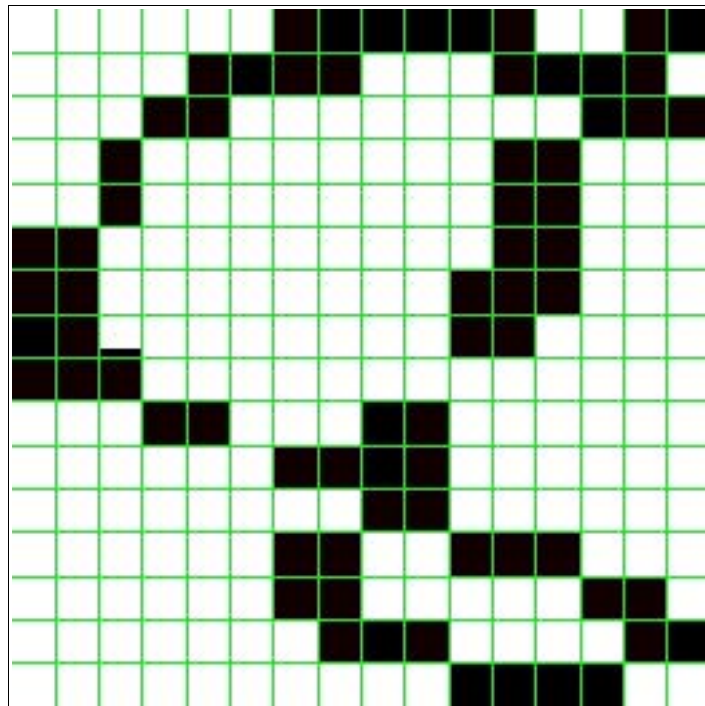


**Diagram 7 : Original image with ROI drawn**

Handwritten Digit Recognition By using libnumrecognition

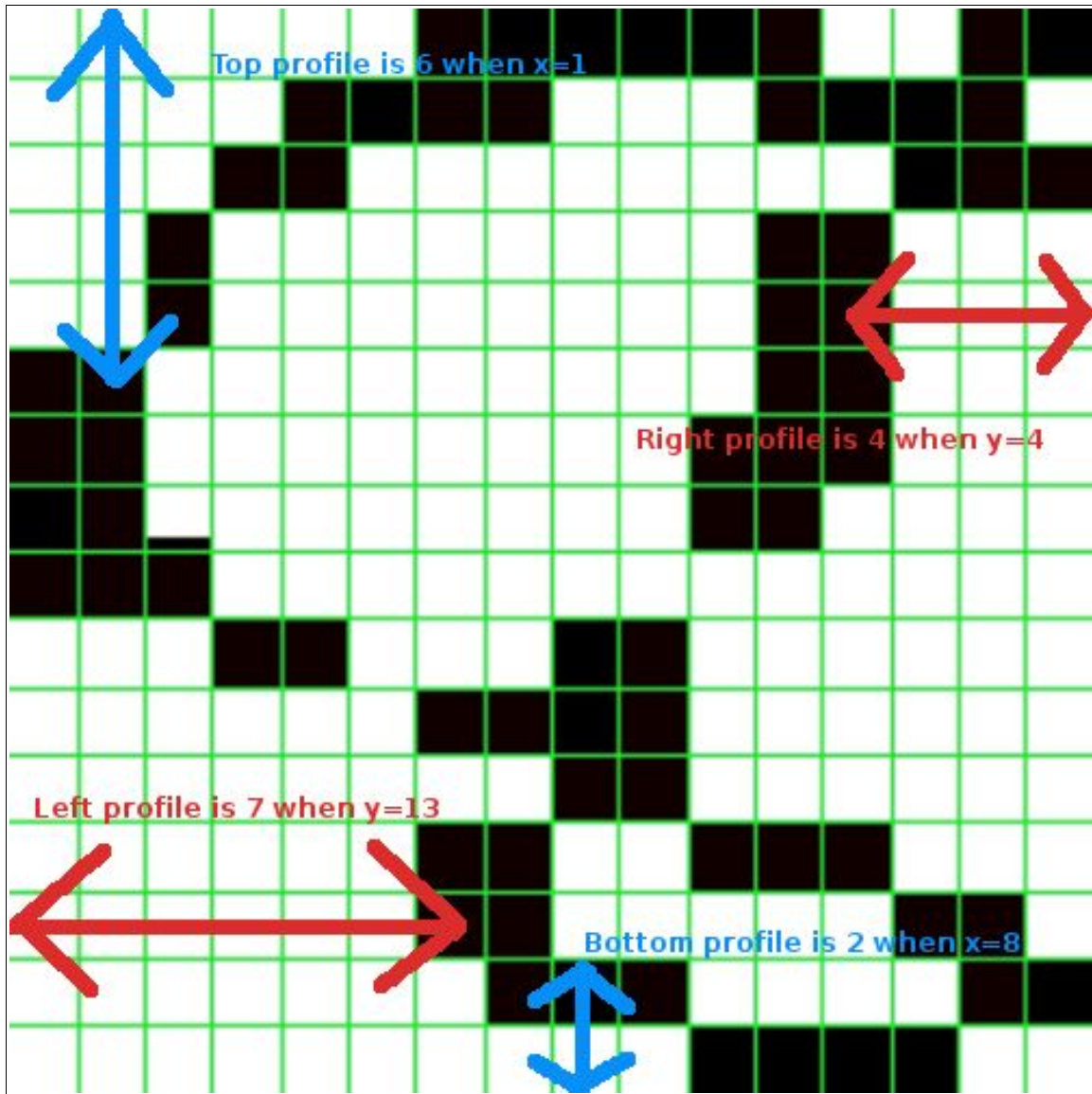


**Diagram 8 : Image extracted from ROI**



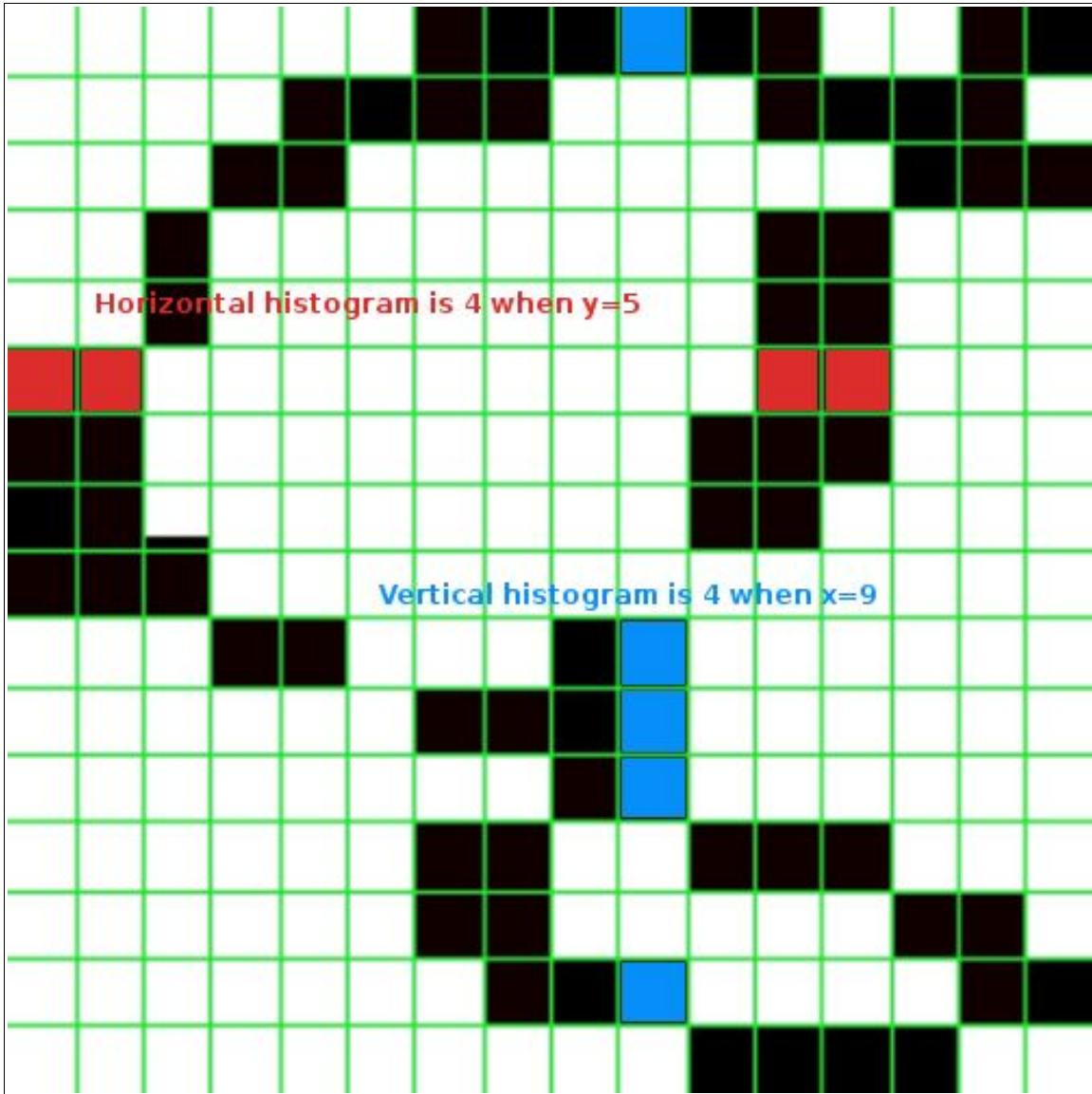
**Diagram 9 : Scaled image for size normalization purpose**

Handwritten Digit Recognition By using libnumrecognition



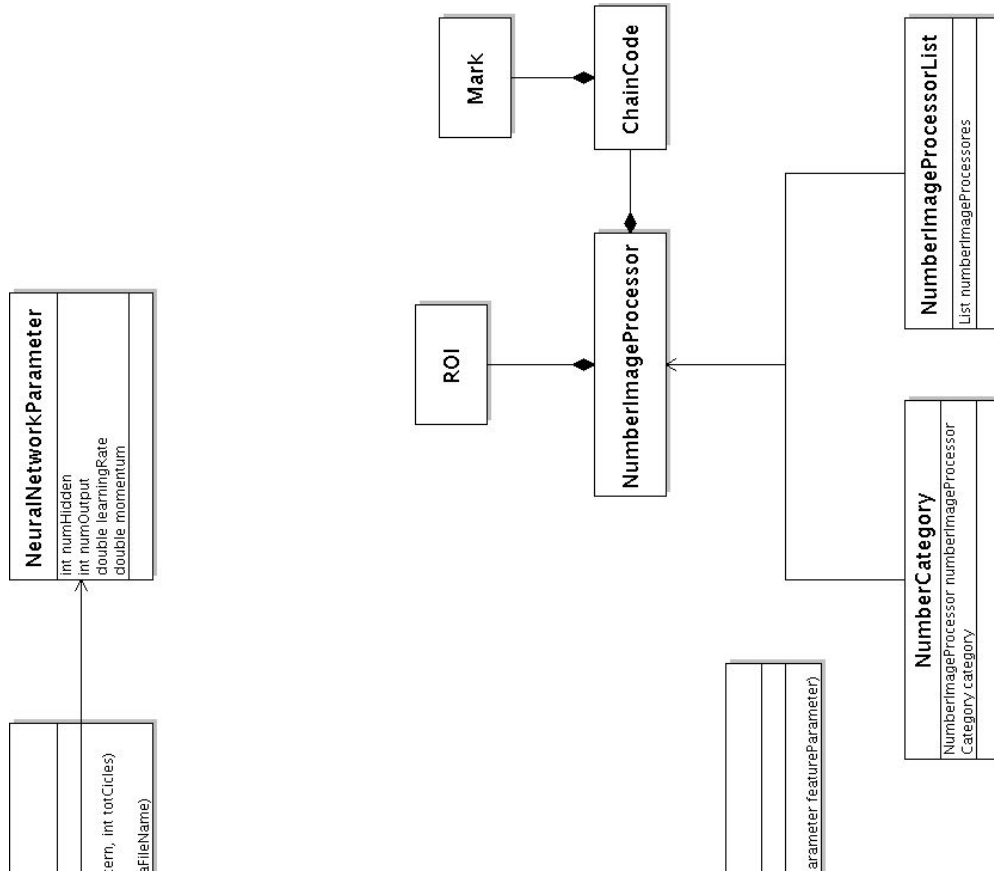
**Diagram 10 : How Left profile, Right profile, Top profile and Bottom profile of an image be found**

Handwritten Digit Recognition By using libnumrecognition



**Diagram 11 : How Horizontal projection and Vertical projection of an image be found**

### 4. UML design of libnumrecognition



# Handwritten Digit Recognition By using libnumrecognition

