Institutionen för systemteknik
Department of Electrical Engineering

Examensarbete

Detection of Bacodes using Machine Learning

Examensarbete utfört i Datorseende
vid Tekniska högskolan vid Linköpings universitet
av

Olle Fridolfsson

LiTH-ISY-EX--YY/XXXX--SE

Linköping 2014

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings tekniska högskola
Linköpings universitet
581 83 Linköping

Detection of Bacodes using Machine Learning

Examensarbete utfört i Datorseende
vid Tekniska högskolan i Linköping
av

Olle Fridolfsson

LiTH-ISY-EX--YY/XXXX--SE

Handledare: **Freddie Åström**
ISY, Linköpings universitet
Ola Friman
Company

Examinator: **Lasse Alfredsson**
ISY, Linköpings universitet

Linköping, 05 March, 2014

Avdelning, Institution Division, Department Computer Vision Laboratory Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden	**Datum** Date 2014-003-005

Språk Language ☐ Svenska/Swedish ☒ Engelska/English ☐ ______________	**Rapporttyp** Report category ☐ Licentiatavhandling ☒ Examensarbete ☐ C-uppsats ☐ D-uppsats ☐ Övrig rapport ☐ ______________	**ISBN** — **ISRN** LiTH-ISY-EX--YY/XXXX--SE **Serietitel och serienummer** Title of series, numbering **ISSN** —

URL för elektronisk version
`http://www.control.isy.liu.se`
`http://www.ep.liu.se`		

Titel
Title

Maskininlärning för detektion av streckkoder

Detection of Bacodes using Machine Learning

Författare
Author

Olle Fridolfsson

Sammanfattning
Abstract

Abstract goes here.

And it can consist of several paragraphs.

Nyckelord
Keywords

key1, key2

Abstract

Abstract goes here.

And it can consist of

several paragraphs.

Sammanfattning

Svenskt abstract kan man placera här.

Acknowledgments

I would like to thank a lot of people…

Contents

Chapter 1

Introduction

Text...

1.1 Some LaTeX resources

A great starting point when you are new to LaTeX is to read [?].

There are many interesting things about LaTeX found in the standard references by Lamport [?] and Gossens et al. [?]. These describe most everything one needs to know about creating documents with LaTeX 2_ε. Gossens et al. has also written a book dealing with graphics in LaTeX, mostly Post-Script based, [?]. Of course there exists many other good references to LaTeX out there too.

Example 1.1: An example of an example

In this example please note that there is a substantional difference between [?] and the first edition of the book [?].

Chapter 2

Overview

Text...

2.1 System overview

To increase the speed a good way is to reduce the amount of data. One way to do this is to use a cascade system, where some amount of data is discarded in each step, illustrated in figure 1. This method can be used both during training and testing.

A great starting point when you are new to LaTeX is to read [**?**].

There are many interesting things about LaTeX found in the standard references by Lamport [**?**] and Gossens et al. [**?**]. These describe most everything one needs to know about creating documents with LaTeX 2_ε. Gossens et al. has also written a book dealing with graphics in LaTeX, mostly Post-Script based, [**?**]. Of course there exists many other good references to LaTeX out there too.

Example 2.1: An example of an example

In this example please note that there is a substantional difference between [**?**] and the first edition of the book [**?**].

Chapter 3

Preprocessing of data

A big amount of gray scale images containing different kinds of code will be available. For each image the corresponding ground truth will also be available. One part of the images will be used as training data and the rest will be used as test data. The idea is to divide each image into blocks of same size. The amount of training data

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

will then be the number of blocks in each image times the number of training images. The blocks can either overlap each other or just lay next each other. Overlapped blocks will lead to higher accuracy but more data have to be processed. Each data will consist of a feature vector:

Each feature will in some way describes the corresponding block. The only information that is available is the intensities of the pixels in each block; consequently all features will always, in some way, be based on the pixel values.

Chapter 4

Machine learning methods

4.1 AdaBoost

The machine learning method that will be tried out first is Boosting which is described in [2]. The basic idea is to train a number of weak classifiers which during the testing will be combined to a strong classifier. To each data in the training dataset there are corresponding weights which are equal for all data at the beginning. The weak classifiers are trained sequentially and after each step the weights are adjusted depending if they were correctly or incorrectly classified.

There exist several variants of Boosting algorithms. The one that will be tried out first is discrete AdaBoost. The weak classifiers in discrete AdaBoost are split functions which simply classifies the data as true or false. The split functions consist of a number of different parameters. The most basic function, which will be tried out first, only has one parameter, a threshold. The function search for a threshold in one dimension at a time and choose the one whish best separates the data.

Chapter 5

Features used for detecting barcodes

5.1 Standard deviation

For the first step in the cascade a good method is to simply compute the standard deviation of each block. Blocks which contain code will have a high standard deviation; hence all data with standard deviation under a certain threshold can be discarded. In this step the amount of data will be decreased a lot.

5.2 Structure tensor

In the next steps one might consider to compute the gradients in the blocks. This can be done by convolving the images with a sobel filter. The gradient image can then be used in several ways. One way is to calculate the eigenvalues of the structure tensor for each block and then use these to estimate the structure inside the block. This can be a good way to distinguish between 1D-code and 2D-code. In a block containing 1D-code the gradient will only vary in one direction, this means it will have an i1D- structure. However in a block containing 2D-code the variation will be fairly equal in both directions. The structure tensor can also be used to calculate Harris-corners, which can be used as a feature

5.3 FAST corner detection

5.4 Distance map

5.5 Local binary pattern

One feature that might be considered is the so called Local Binary Pattern, which is described in [3]. The basic idea is to compute a binary code in every pixel, based on the difference of the intensity between the pixel and the surrounding pixels, illustrated in figure 2. The binary code will then be transformed to a decimal scalar value. If a 3x3 neighborhood is used there will be 256 different possible values. For each block a histogram will be calculated for all these values. Every bin in the histogram will then be used as a feature. If there is a bin for every possible value, there will be 256 features.

Chapter 6

Cascade

Chapter 7

Evaluation

7.1 Evaluation of features

7.2 Evaluation of cascade

Chapter 8

Conclusions