BABEŞ-BOLYAI UNIVERSITY CLUJ–NAPOCA
FACULTY OF MATHEMATICS AND INFORMATICS
SPECIALIZATION: COMPUTER SCIENCE

License Thesis

Road sign recognition

Abstract

There are numerous methods for road sign recognition but the rubust and cost-effective solution is still an active research subject. The recognisability of traffic signs is affected by many variables, like luminous intensity, shading, partial coverage and other obstacles. The developed system is recognising road sings using machine learning techniques taking in consideration the above mentioned factors. The most imoprtant ? is accuracy and speed. The application displays the examined image and analyses it on more background threads. Extracts random rectangles from the original picture and decides if it contains a traffic sign or not, and if it does categorises it. In case of a hit saves the coordinates and enframes it on the picture.

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

JULY 2015 SZABO ÁGNES-TERÉZ

ADVISOR:
CSATÓ LEHEL BABEŞ–BOLYAI UNIVERSITY,
FACULTY OF MATHEMATICS AND INFORMATICS

BABEŞ-BOLYAI UNIVERSITY CLUJ–NAPOCA
FACULTY OF MATHEMATICS AND INFORMATICS
SPECIALIZATION: COMPUTER SCIENCE

License Thesis

Road sign recognition

SCIENTIFIC SUPERVISOR:

CSATÓ LEHEL BABEŞ–BOLYAI UNIVERSITY,
FACULTY OF MATHEMATICS AND INFORMATICS

STUDENT:

SZABO ÁGNES-TERÉZ

JULY 2015

UNIVERSITATEA BABEŞ-BOLYAI, CLUJ–NAPOCA
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

Lucrare de licenţă

Recunoaşterea semnelor de circulaţie

CONDUCĂTOR ŞTIINŢIFIC:

CSATÓ LEHEL
UNIVERSITATEA BABEŞ–BOLYAI,
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

ABSOLVENT:

SZABO ÁGNES-TERÉZ

IULIE 2015

BABEŞ-BOLYAI TUDOMÁNYEGYETEM KOLOZSVÁR
MATEMATIKA ÉS INFORMATIKA KAR
INFORMATIKA SZAK

Licensz-dolgozat

Forgalmi táblák felismerése

TÉMAVEZETŐ:
CSATÓ LEHEL
BABEŞ–BOLYAI TUDOMÁNYEGYETEM,
MATEMATIKA ÉS INFORMATIKA KAR

SZERZŐ:
SZABO ÁGNES-TERÉZ

2015 JÚLIUS

Contents

1. Chapter

Introduction

1.1 Goal

There are many systems that can detect and recognise road signs on pictures taken in real life enviroment, but the robust cost-effective solution is still an active research subject. Since there are numerous types of traffic signs and many of them are similar to one another, the task of classifying them is challenging. The recognizability of the road signs is also affected by the following things: luminous intensity, shading, partial coverage, and other obstacles.

Some people are afraid of driving because they can not divide their atention in many places. If they would have a system to notify them of visible traffic signs, they might gain confidence. My goal is to make a fast and efficient system.

1.2 Implementation

This system recognises road signs with machine learning algorithms.I use artificial neural networks, a subclass of supervised learning.

When an application is placed in real enviroment it is crucial for it to be reliable. Wrongly identifying a road sign can have fatal consequences.

Because traffic signs can change with time it is important that the designed system can easily accomodate to these changes, without the need to change it to the core. Artificial neural networks are perfect for this, because if the data changes it is enough to perform the learning process again, there is no need do change the code.

2. Chapter

Road signs

2.1 What are road signs

Traffic signs control the flow of traffic, warn the driver of hazards ahead, guide the driver to their destination, and inform them of roadway services. There are three basic types of traffic signs all with different shapes:

- signs that give orders, circular
- warning signs, triangular
- signs that give information, rectangular

The stop sign is an exception, it is an octagon.

A further guide to the function of a sign is its colour. The colors used are red, white, black, and blue.

Most of the road signs are composed of two parts: a solid band on the outside and the inside figure or number. The inside symbol is usually black.

2.2 Charateristics

2.3 Data set

2.3.1 Training data

I used ... . It contains 1213 training images, their size varies from 16x16 to 128x128 pixels, they appear in every perspective and under every lighting condition. The images are divided in 43 categories based on the containd traffic signs.

Altough there is a large amount of training images, they are not devided equally among categories. This makes it difficult for the network to learn the specifics of the classes that contain small amounts of images.

I try to equalize the number of training images by generating more images into that category. The original training set contains images that completely contain the traffic sign. Since my purpose is to recognise partially visible road signs too, I generate new images from the old ones by cutting out 70-80% of them.

Figure 2.1: Example of the training images

2.3.2 Test data

There are 900 test images, containing zero to six traffic signs. They are significantly larger pictures: 1360 x 800 pixels. On the 2.2 figure is an example of the test images, that contains two road sings.

Figure 2.2: An example of the test images

3. Chapter

Segmentation

In order to find traffic signs on an image, I use segmentation. Segmentation is a process of cutting sub-images out of the original one.

There are several different types of procedures for image segmentation. These include: edge and line oriented segmentation, and representation schemes, region growing methods, clustering, and region splitting.

The method I used is based on randomly selecting segments from the original image. I generate square segments by randomly selecting a point from the original image and a random length.

Because a picture taken from real environment may contain more than just one road sign, it is important to search the entire image. The large number of segments results in a high probability of detecting each road sign on the image. The size of a sign may also vary, therefore, during segmentation it is important that the size of the segmented images vary the same way.

By randomly selecting the segments from the original picture a significant number of the generated images will not contain a traffic sign or they will contain it only partially. This is the reason why the classification of image must recognize road signs even if only 70-80% is visible on the selected segment.

4. Chapter

Preprocessing images

In order to simplify the data the learning algorithm is going to work with, I preprocess the images based on their color. The neural network works with the gray value of the pixels. Because the images are taken in real life environment, they contain other objects (branches). They were taken in different light conditions; some of them ending up really obscure or blanch.

4.1 Color spaces

Color space is a method by which we can specify, create and visualize colors. A color is usually specified using three coordinates, or parameters. These parameters describe the position of the color within the color space. Different color spaces are better for different applications.

4.1.1 RGB color space

RGB color space is an additive color system based on tri-chromatic theory. The tri-chromatic theory describes the way three separate , red, green and blue, can match any visible color. RGB is easy to implement but non-linear with visual perception. RGB is frequently used in most computer applications since no transformation is required to display information on the screen. RGB space may be visualized as a cube with the three axes corresponding to red, green and blue.

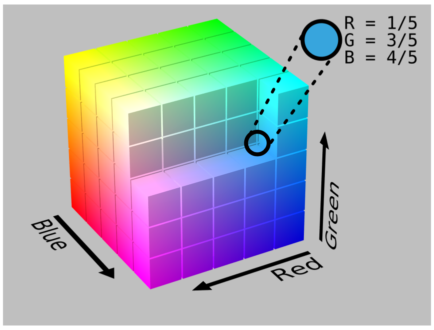

Figure 4.1: RGB Color space

4.1.2 HSV color space

The HSV color space is one of the most common cylindrical-coordinate representations of points in an RGB color model. The hue is the human sensation according to which an area appears to be similar to one, or to proportions of two, of the perceived colors red, yellow, green and blue. The saturation is the colorfulness of an area relative to its brightness. The value (brightness) is the human sensation by which an area exhibits more or less light. The color is then defined as a position on a circular plane around the value axes. Hue is the angle from a nominal point around the circle to the color while saturation is the radius from the central lightness axis to the color.

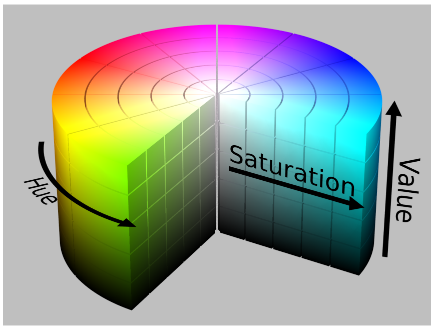

Figure 4.2: HSV Color space

4.2 Implementation

The traffic signs are meant to raise awareness therefore they use vivid colors, like red, yellow, blue, black, white. I create a new black and white image by filtering the colors of the original image. If the hue, saturation and value of a pixel satisfy a certain threshold, the pixel of the new image will be black otherwise white. I established these thresholds based on experiments and the 4.3 figure.

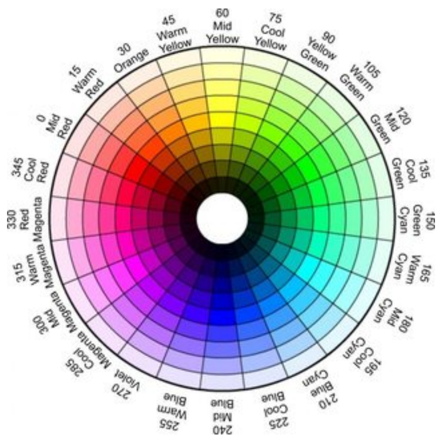

Figure 4.3: Color wheel

5. Chapter

Neural networks

5.1 Artificial neurons

5.1.1 Sigmoid neuron

The sigmoid neuron takes inputs with values between 0 and 1 and produces one output int the same interval.

[kep a nuronrol]

Weights are assigned to each input, representing their importance in the output of the neuron. Each neuron has a bias that is meant to correct small disorders in the behaviour of the network.

The output of a sigmoid neuron is calcualted with the 5.1 formula.

$$\sigma(\sum_{i=1}^{n-1} w_i * x_i + b) \tag{5.1}$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{5.2}$$

The function in the 5.2 formula is called the activation function. Activation functions are ment to place the output in a given interval. In case of the sigmoid function this interval is between 0 and 1.

A characteristic of the sigmoid neuron is that a small change in the weights produces a small change int the output. This can be a disandvantage as much as an advantage. It is an advantage because when the output is close to the desired outcome it can modify it without steping over it. It is a disadvantage because when the output is terribly wrong, it takes a lot of time to correct it.

5.2 Architecture

I used a netwotk with 625 (25*25) input neurons representing the gray value of each pixel, and 46 (number of road sings categories) output neurons. I experimented with different numbers of hidden layers and various numbers of neurons in them.

5.3 Stochastic gradient descent

To change the weights and biases based on the deviation of the calcualted output to the expected output I used stochastic gradient descent, a simplified version of gradient descent. Gradient descent calculates the cost based on the the enire training set, while stochastic gradient descent approximates it based on a number of randomly selected data. If the number is large enough the approximation will be more accurate, but since it doesn't work with the whole data set it is faster.

Equation 5.3 is the quadratic cost function. *n* is the number of training inputs, *y(x)* is the output generated by the network for the x input, and *a* is the desired output for the x input. When the desired and the calculated outputs are close to each other the cost will be small. Since my goal is to approach the calcualted output to the desired one I have to minimise the value of the cost function. For this I introduce the gradient of the cost function, denoted as ∇C defined with the vector in the 5.4 formula. [Doe, 2100]

$$C = \frac{1}{2n}\sum\nolimits_x \|y(x) - a(x)\|^2 \tag{5.3}$$

∇C is the gradient of the cost function C and is defined with the vector in the 5.4 formula.

$$\nabla C = (\frac{\partial C}{\partial w_i}, \frac{\partial C}{\partial b_j})^T \tag{5.4}$$

In order to minimise the cost I repeatedly applied 5.5 and 5.6 rules to the weights and biases. η is the learning rate. It has to be a small munber, but not too small, because the learning process would be too slow.

$$w_i \rightarrow w_i - \eta\frac{\partial C}{\partial w_i} \tag{5.5}$$

$$b_j \rightarrow b_j - \eta\frac{\partial C}{\partial b_j} \tag{5.6}$$

5.4 Backpropagation

Backpropagation is used to adapt the weights and biases in the network to a achieve the desired output.

In equation 5.7 z^l is the wighted input vector in the l^{th} layer. As can be seen int equation 5.8 the output vector in a layer is calcualted using the output of the previous layer, thus forwrding the information. After calcualting the output of the network the 5.9 formula calcualates the error of the final layer, where *L* is the number of layers, and $\nabla_a C$ is a vector containing the $\frac{\partial C}{\partial a_j^L}$ partial dervates.

The backpropagation is done with the 5.10 formula for each layer *l* from *L-1* to 2.

$$z^l = (w^l * a^{l-1} + b^l) \tag{5.7}$$

$$a^l = \sigma(z^l) \tag{5.8}$$

$$\delta^L = \nabla_a C \circ \sigma'(z^L) \tag{5.9}$$

$$\delta^l = ((w^{l+1})^T * \delta^{l+1}) \circ \sigma'(z^l) \tag{5.10}$$

Bibliography

J. Doe, *The Book without Title*. Dummy Publisher, 2100.