UNIVERSITÄT KOBLENZ · LANDAU
Fachbereich 4: Informatik

Texture-Based Text Detection in Digital Images Using Wavelet Features and Support Vector Machines

Bachelorarbeit
zur Erlangung des Grades
Bachelor of Science
im Studiengang Computervisualistik

vorgelegt von

Johann Raskatow

Betreuer: Dr.-Ing. Marcin Grzegorzek, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Erstgutachter: Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Zweitgutachter: Natalia Vassilieva, HP Labs: Information Fusion and Real Time Delivery Lab, St. Petersburg, Russia

Koblenz, im September 2010

Kurzfassung

In dieser Bachelorarbeit wird ein neues texturbasiertes Verfahren zur Detektion von Texten in digitalen Bildern vorgestellt. Das Verfahren kann im wesentlichen in zwei Hauptaufgaben unterteilt werden, in Detektion von Textblöcken und Detektion von einzelnen Wörter, wobei die einzelnen Wörter aus den detektierten Textblöcken extrahiert werden. Im Groben agiert das entwickelte Verfahren mit mehreren Support Vector Machines, die mit Hilfe von waveletbasierten Merkmalen mögliche Textregionen eines Bildes zu wirklichen Textregionen klassifizieren. Die möglichen Textregion werden dabei durch unterschiedlich ausgerichtete Kantenprojektionen bestimmt. Das Resultat des Verfahrens sind X/Y Koordinaten, Breite und Höhe von rechteckigen Regionen eines Bildes, die einzelne Wörter enthalten. Dieses Wissen kann weiterverarbeitet werden, beispielsweise durch eine Texterkennungssoftware, um an die wichtigen und sehr nützlichen Textinformation eines Bildes zu gelangen.

Abstract

In this bachelor thesis a new texture-based approach for the detection of text in digital images is presented. The procedure can be essentially divided into two main tasks, in detection of text blocks and detection of individual words, whereby the individual words are extracted from the detected text blocks. Roughly, the developed method acts with multiple support vector machines, which classify possible text regions of an image into real text regions, using wavelet-based features. In the process the possible text region are defined by edge projections with different orientations. The results of the approach are X/Y coordinates, width and height of rectangular regions of an image, which contains individual words. This knowledge can be further processed, for example by an optical character recognition software to get the important and useful text information.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäßübernommen wurden, sind als solche gekennzeichnet.

Die Vereinbarung der Arbeitsgruppe für Studien- und Abschlussarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	ja ☐	nein ☐
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	ja ☐	nein ☐

Koblenz, den 28. September 2010

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter will give a short overview, about the task text detection in general and the content of this thesis. In addition this chapter also will address to the problems which text detection entails, which solution statement this thesis deals with and a lot of related works, which inspired me during the development.

1.1 Text Detection in General

Text in digital images carries rich and important information, which can be very helpful in many areas of computer science, be it for automatic annotation of digital images, indexing in multimedia databases or identification of relevant features such as postal code, for automatic sorting of postal documents. The general tasks to extract text information from digital images are text recognition and text detection. Text recognition is defined as the task which recognize text in text regions and translate them into machine-encoded text. Text detection is defined as the task which localizes text in images at block, line or word level without recognizing individual characters. An example at word level is shown in Figure 1.1. To perform text detection before recognition entails many advantages. First of all, text embedded in images usually does not cover the majority of pixels, so that it is not an economic way to perform character recognition on non text regions. Second, the background inside a localized text region is usually less complex, than the whole image, if text characters are clearly visible. Furthermore, the localization of text strings is easier and more robust, than the localization of individual characters, because in the most cases text strings posses typical shapes and are aligned line by line. This thesis will present an approach which detect at first text blocks, followed by a text line extraction from the detected text blocks and finally words are detected from the text lines, so that the proposed approach covers all variants of text detection.

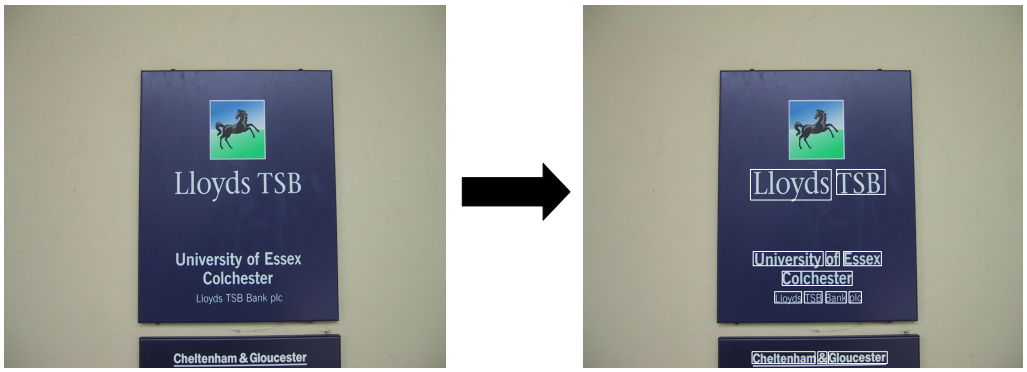


Figure 1.1: Text Detection at Word Level

1.2 Difficulties

The development of a fast and robust text detection algorithm involve several difficulties such as:

- Text can be embedded in complex background
- To find effective features to discriminate text with other text-like textures, such as leaves or window curtains
- Text pattern varies with different font-size, font-color and languages

1.3 Related Work

In this section some existing methods for text detection are reviewed. In the literature, text detection is regarded as an unique research area. Previous text detection methods in digital images are well classified by Jung et al [JKJ04] into two main groups, namely Region-Based and Texture-Based methods:

- **Region-Based Methods**
  Region-Based methods can be further divided into two classes: connected components and edge-based approaches. These methods are also known as bottom-up approaches. They directly segment images into regions by identifying elementary substructures such as connected components or edges,

and then grouping/merging these substructures successively into larger structures, until text areas are detected. Geometrical analysis based on different thresholds or several heuristic are finally applied in order to filter out possible false alarms. In connected component methods, the basic elements are created using the similarity of neighbour pixels in grayscale or color levels, whereas the edge-based methods focus on the high contrast between the text and the background, identifying first the edges caused from the text contours and then grouping them, if possible.

- **Texture-Based Methods**
  Texture based methods are based on the assumption that text present in images exhibits some distinct textural properties, which may be used to distinguish it from the background. To extract the textural properties of a text region in an image, Gabor filter, Wavelet transformations, Fast Fourier Transformations ect. are usually used.

1.3.1 Connected Components Methods

`Shim et al.`[SDB98] use the homogeneity of intensity of text regions in images. Pixels with similar gray levels are merged into a group. After removing significantly large regions by regarding them as background, text regions are sharpened by performing a region boundary analysis based on the gray level contrast. The candidate regions are then subjected to verification using size, area, fill factor and contrast.

`R. Jiang et al.`[JQXW06] introduce a novel connected components(CC) method which works as follows: First, the input image is decomposed into connected components by clustering algorithm including text and non text CC. To segment text from background a two-stage classification module is used. In which all the CCs are verified by a cascade classifier and a Support Vector Machine(SVM). The classifier is combined by a series of weak classifiers. Most apparently non-text CCs are discarded as early as possible to save a great deal of computation. SVM concentrates on CCs accepted by the cascade and does further verification. Only those accepted by both cascade classifier and SVM are output in final result. 15 features are totally selected to discriminate text CCs from non-text CCs . All these features can be divided into 5 categories: geometric features, shape regularity features, edge features, stroke features and spatial coherence features. The cascade classifier consists of a series of weak classifiers, each concentrates on one feature mentioned. A weak classifier is composed by a feature and two thresholds: one upper threshold and one lower threshold. For each input CC, the weak classifier measures the feature and makes the decision whether the CC is text or not. At

the beginning all CC extracted in the clustering step are put into the first weak classifier. It measures certain feature on CCs one by one and categorizes them into positive or negative. The negative CC are rejected immediately and for positive similar processing is repeated in following weak classifiers until the end of the cascade. Without the cascade, the system would be quite computationally exhaustive. Due the advantage of cascade, there is no need to calculate all 15 features for all CCs. The cascade classifier, helps to accelerate the processing greatly.

1.3.2 Edge-Based Methods

`Roshanak Farhoodi and Shohreh Kasaei` [FK08] proposed a new method to segment text blocks from images based on finding text edges using information content of the subimage coefficients of the discrete wavelet transformed input image. Here, the coefficients of the horizontal, vertical, and diagonal subimages of the first level are used. Then the edges are combined to an edge map, to form the exact location of the characters. Here a Sobel operator is applied on each subimage and a weighted OR operator is used to decide whether a pixel belong to an edge in the image or not, in relation to the coefficients in the vertical, horizontal and diagonal subimages. As the next step a morphological dilation is performed on the processed edge map, using a structuring element with the size of 1x6. Finally the regions that are not acceptable as text regions are removed based on general structure rules, like: texts always contain edges, texts are some bars whose widths are larger than their heights, texts are bounded in size, and texts have a special texture property.

`Julinda Gllavata`[Gll07] proposed two projection-based methods which belong to the region-based methods. Both methods are mainly based on the assumption that the text background contrast is high and furthermore the density of edges in the areas of the text contours is higher compared to the other parts of the images. These methods consist essentially of three steps: 1. Image preprocessing 2. Edge detection 3. Text line localization analysing the projections profiles. They differ in the last step: The first method employs a global threshold for text-localization, whereas the second employs an adaptive threshold depending on the complexity of the image. The Global method performs very well in detecting text embedded in quite simple background although the contrast text/background may be quite low, but its performance diminishes with the increasing of the background complexity. In contrast to this, the Local method detects and localizes the text more accurately.

1.3.3 Texture-Based Methods

`Datong Chen`[COB04] proposed a texture-based text detection method by applying a machine learning location scheme. It consists of two steps. The first step locates candidate text regions in images with a fast algorithm. This localization process avoids applying the machine learning classifiers on the whole images as well as to further reduce the variation of text size by extracting individual text strings (lines). To obtain a fast algorithm, candidate text blocks are located by exploring heuristic characteristics. A threshold in this algorithm is used to adjust the weakness of the heuristic feature based classifiers in distinguishing text and backgrounds. The resulting false alarms will be removed in the following verification step. In the verification step, a size normalization is first performed on the candidate text lines. A machine learning approach, a support vector machine (SVM), is employed to separate text regions from background regions in the candidates. Due the large variance of the grayscale values of text characters, training of SVMs and the verification of text lines are all performed in feature spaces.

`Z.Ji et al.`[JQXW06] proposed a novel text detection method in video frames using hybrid features. These approach works broadly as follows: a small overlapped sliding window is scanned over an image from which language independent, texture based and edge based features are extracted. 24 features are totally used, 8 from wavelet transform coefficients, 12 from gray level co-occurrence matrix features, and 4 from oriented edge intensity ratio. In the following, each window is classified as text or non text window with SVM classifier. Then a vote mechanism is employed to judge every small block as text or non text. Lately a morphological filter is performed to precisely locate the text regions. The experiments they executed, shows the effectiveness and robustness over a comprehensive database.

1.3.4 ICDAR2003 Text Locating Competition

In addition to the mentioned methods, the evaluated methods at the ICDAR2003[1] [LPS+05]text locating competition will be presented in this section. The reasons for are introduced in Chapter 3. In short, there is no ordinary evaluating system and to be able to assess the developed approach this methods are taken to this chapter. The methods which were evaluated at the competition were:

`Ashidas`[LPS+05] System is based on the following steps: First fuzzy clustering algorithm(pixel color based) is applied to a given image, resulting in a set of binary images called color separation images. Second, some blobs in each color separation image are grouped under simple heuristic constrains to calculate the

[1] http://algoval.essex.ac.uk/icdar/Competitions.html

geometric features. Finally, an Support Vector Machine trained on these features selects the blobs corresponding to character patterns.

`HWDavids`[LPS+05] System can be described as follows: The first step is to apply four Sobel edge operators on an input image and to compute from each image position the edge intensity. From this a gradient density image is produced using a low-pass filter. Then a binarised image is computed by threshholding. On this a lot of morphological(closing, opening) operations are applied to eliminate connected strokes and to remove isolated regions. Additionally, a conditional morphological operation is applied on the connected components, which is based on a CCA (connected component analysis) algorithm. Finally components are classified as text or non text by some heuristic methods.

`Wolfs`[LPS+05] System employs a similar set of operations to the `HWDavid` system, but there are a few differences. First, the classification heuristics are replaced with an Support Vector Machine. Secondly, the order of the classification and morphology operators are reversed compared with `HWDavid`. Furthermore the `HWDavid` was nearly 60 times faster than Wolf, which is probably explained by the fact that Wolf used an Support Vector Machine at an early and therefore data-intensive processing stage.

`Todorans`[LPS+05] System uses multi-scale texture and edge analysis which can be divided in the following processes: First, a texture filter is applied to extract the candidate text regions. For this a local energy was computed, estimate for each color channel at three different scales using second order derivative filters. The filters used in estimation are Gaussian kernels and the local energy values are clustered in an 9 dimensional space using the K-means algorithm by expecting that the cluster corresponding to the lowest energy comprises the text region. Secondly vertical edges are extracted from the original image masked with text regions provided by the texture filter step. The vertical edges representing small portions of candidate characters are merged by morphological closing in horizontal direction. Then blobs are extracted from the image of filtered vertical edges which represent characters and word parts. Using geometric features a set of blobs was filtered and combined into text lines. The above processing steps were applied at each scale of an image pyramid.

1.4 Contribution

During the research for related works, a trend using neural networks such as support vector machines to classify candidate text region into text or non text region

on the basis of texture features, resulted. This trend was the point of departure to develop a new approach for text detection. Furthermore region-based approaches without the assumptions about size, proportion, direction ect. lead to heavy classification tasks. Consequently, the idea for a new approach was to develop a hybrid text detection algorithm i.e. to combine texture-based approaches with region-based methods, which should be a good solution for these task. Under this principle the whole development was oriented. The contributions in this thesis are:

- Development of a new approach for text detection under the described assumption
- Implementation of the developed approach
- Evaluation using evaluation system of the ICDAR2003[2] text locating competition

1.5 Overview

The following thesis can be organized as follows: Chapter 2 describes the whole process cycle of the developed approach. Here the individual steps are described in relation to the ideas behind them, to the algorithms and to the intermediate results which every step produce. Chapter 3 presents experimental results and the evaluation system, which is used to assess the proposed approaches of the ICDAR2003[3] competition. The last chapter contains a summary and ideas how this approach can be improved for future work.

[2]http://algoval.essex.ac.uk/icdar/Competitions.html
[3]http://algoval.essex.ac.uk/icdar/Competitions.html

Chapter 2

Text Detection

The first part of this thesis was to research related works to get inspiration and a first impression in which direction the approach should be developed. The general results of this part were the assumptions, that text in digital images contains a large number of short edges what directly lead to apply edge detection, but also that text regions in digital images should have an unique texture, what leads to a texture classification algorithm. This chapter includes the whole process cycle Figure 2.1 of the developed approach. Step by step the developed approach will be described, in relation to the ideas behind them, to the algorithms and to the intermediate results which every step produce.

2.1 Preprocessing

First step of the preprocessing is to convert the whole input image from RGB to grayscale with the following equation:

$$Y = 0.3R + 0.59G + 0.11B \tag{2.1}$$

Secondly the converted image is normalized Equation (2.2)(2.3). Besides, the width w and the height h of the input image is subtracted by the result of modulo 100.

$$w' = w - (w \mod 100) \tag{2.2}$$

$$h' = h - (h \mod 100) \tag{2.3}$$

This normalization is necessary for the next step, where a Sliding Window with the size of 100x100 iterates over the input image in slide step of 100 pixel and segment possible text regions.

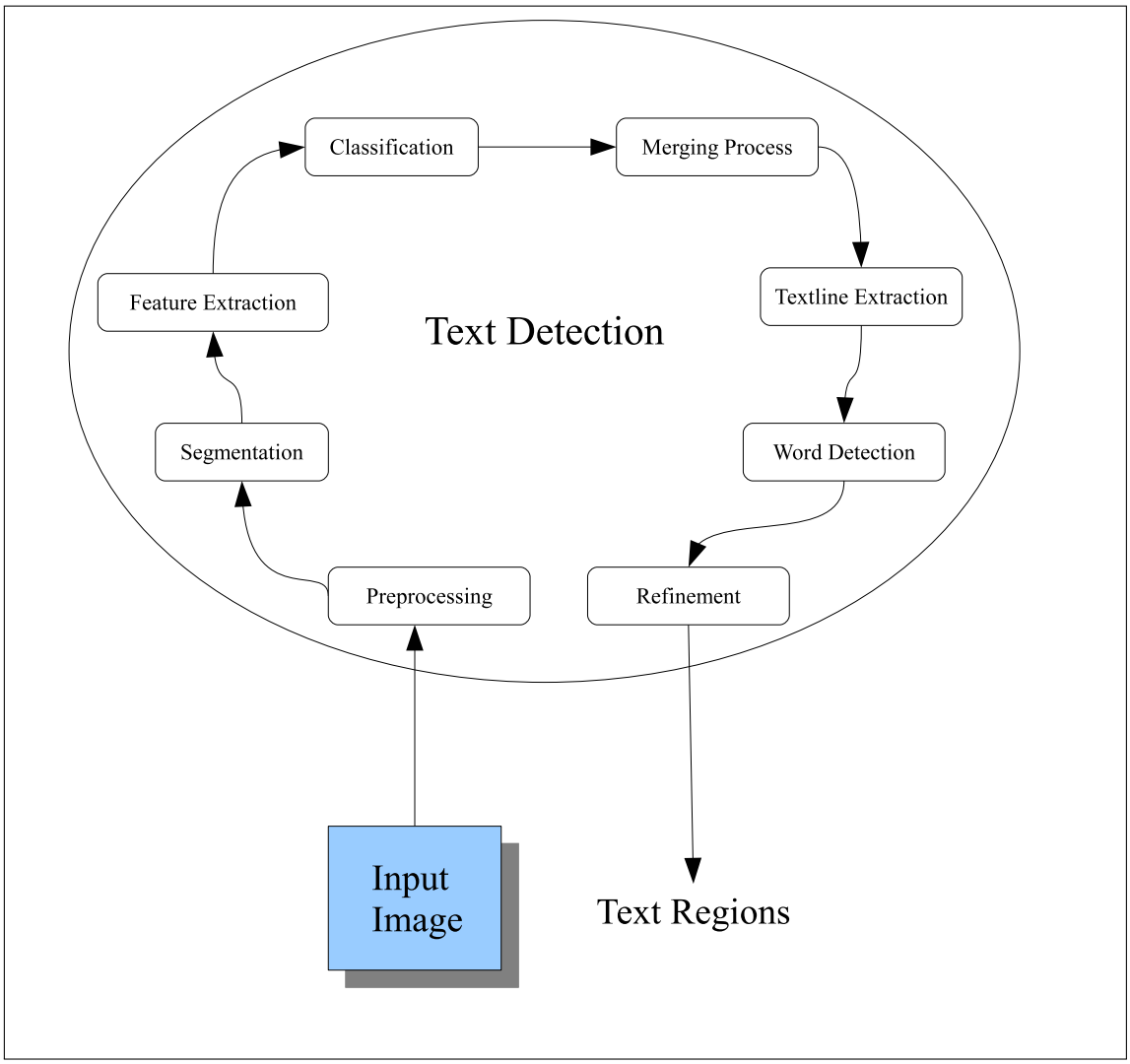


Figure 2.1: Process Cycle

2.2 Segmentation

The second process of the developed approach is to segment possible text regions. Here the property is used that text regions are in possession of many short edges. Roughly this step can be described as follows: A Sliding Window with the size of 100x100 iterate over the resulted image in the preprocessing. For each sliding window (subimage) an horizontal projection is calculated, which represent a histogram of edges in the horizontal direction of an edge image. Some examples for a horizontal projection applied to images contains text and not are shown in Figure 2.2 The first stage of a projection is to convolve an edge detection filter

Figure 2.2: Examples for Horizontal Projection

with the subimage. Here can be used any ordinary edge detection algorithm such as Canny, Sobel etc. The algorithm used in this approach is similar to Sobel and was used by Gllavata [Gll07]. The algorithm is based on the fact that character contours have high contrast to their local neighbours and functions as follows:

```
1  for (int i = 1; i < img->height-1; ++i) {
2  for (int j = 1; j < img->width-1; ++j) {
3
4  int leftD = abs(img[i][j]-img[i-1][j]);
5  int upperD = abs(img[i][j]-img[i][j-1]);
6  int rightUpperD=abs(img[i][j]-img[i+1][j-1]);
7
8  int val = MAX(leftD,upperD);
9  edgeMap[i][j] = MAX(val,rightUpperD);
10  }
11 }
```

Listing 2.1: Edge Detection Algorithm

The value of each pixel of the edgemap is evaluated as the largest difference between the grayscale values of the respective pixel in the original image and its neighbors (in horizontal, vertical and diagonal direction). As a result, all character pixels as well as some non-character pixels, which also show high local intensity contrast are registered in the edge map. Then a simple mean filter is convoluted with the grayscale image to delete some noise edges. The projection itself is a histogram of an edgeImage in the horizontal direction and is computed as shown in following equation.

$$\boldsymbol{HP}[y] = \sum_{\forall x | \boldsymbol{E}(x,y) > k} 1 \text{ for } y = 1...M \tag{2.4}$$

Where $\boldsymbol{E}$ represent the edgeImage and $k =$ is experimental set to 10. After computing such a histogram it can be already decided, whether a subimage is a possible text region or not. For most text regions the histogram have more than two values in a particular area, namely between 20% and 60% of the image width (the values derived from experiments). The results of this process are labelled regions with the size of 100x100 in the normalizes image, which represent the possible text regions.This method achieves already relative good results, especially on simple backgrounds which contains text. An Example is shown in Figure 2.3.

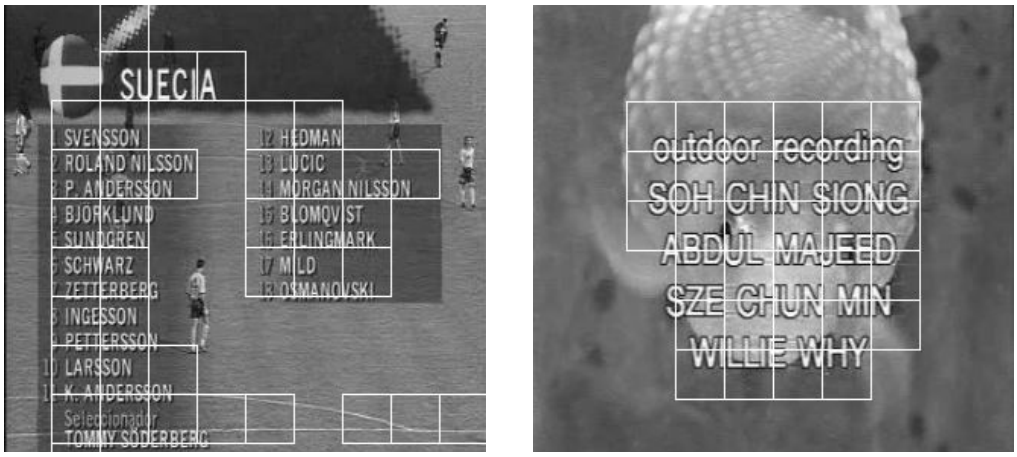


Figure 2.3: Results after Segmentation

2.3 Feature Extraction

If a subimage is labelled as a possible text region in the segmentation procedure, a feature extraction is applied on it. Furthermore if nary subimages were labelled as a possible text region the whole algorithm would determine, and the result would be, that the input image does not contain text. The feature extraction itself is based on the assumption that text regions have unique textures, so that it is possible to distinguish text regions from non text regions. There are already several methods which are used in approaches for text detection. Usually they operate with conventional methods like using Gabor Transformation or Fourier Transformations ect.. This approach uses the Haar Wavelet Decomposition and is an extension of the co-occurrence histogram method. The used algorithm was developed by P.S.Hiremath und S. Shivashankar [PS06] and achieved excellent results in texture classification. Furthermore a wavelet transform-based texture classification algorithm entails several important characteristics:

- The wavelet transform is able to decorrelate the data
- The wavelet transform provides orientation sensitive information which is essential in texture analysis.
- The computational complexity is significantly reduced by considering the wavelet decomposition.

The process in itself can be described as follows: Given an image $\boldsymbol{I}$, the Haar Wavelet decompose a given image $\boldsymbol{I}$ into four subimages, lower frequency image ($\boldsymbol{L}$), vertical high frequency image ($\boldsymbol{V}$), horizontal high frequency image ($\boldsymbol{H}$), and diagonal high frequency image ($\boldsymbol{D}$).

$$\boldsymbol{L}(x,y) = \frac{1}{4}(\boldsymbol{I}(2x,2y) + \boldsymbol{I}(2x,2y+1) + \boldsymbol{I}(2x+1,2y) + \boldsymbol{I}(2x+1,2y+1)) \quad (2.5)$$

$$\boldsymbol{V}(x,y) = \frac{1}{4}(\boldsymbol{I}(2x,2y) - \boldsymbol{I}(2x,2y+1) + \boldsymbol{I}(2x+1,2y) - \boldsymbol{I}(2x+1,2y+1)) \quad (2.6)$$

$$\boldsymbol{H}(x,y) = \frac{1}{4}(\boldsymbol{I}(2x,2y) + \boldsymbol{I}(2x,2y+1) - \boldsymbol{I}(2x+1,2y) - \boldsymbol{I}(2x+1,2y+1)) \quad (2.7)$$

$$\boldsymbol{D}(x,y) = \frac{1}{4}(\boldsymbol{I}(2x,2y) - \boldsymbol{I}(2x,2y+1) - \boldsymbol{I}(2x+1,2y) + \boldsymbol{I}(2x+1,2y+1)) \quad (2.8)$$

This subimages are necessary to compute the co-occurrence histograms, which are constructed across different wavelet coefficients of an image and its complement decomposed upto 1-level. The combinations considered are $(\boldsymbol{L},\boldsymbol{V})$, $(\boldsymbol{L},\boldsymbol{H})$,

$(\boldsymbol{L}, \boldsymbol{D})$,$(\boldsymbol{L}, |(\boldsymbol{D} - \boldsymbol{H} - \boldsymbol{V}|))$ and the same with the complement image. The translation vector is denoted by $\boldsymbol{t}$[a,d], where d is the distance and a the angle. Here a distance of $1(d = 1)$ and eight angles $(a = 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°)$ was considered. The co-occurrence histograms for each combination and the eight angles, are constructed yielding 16 histograms per pair. The feature set comprises in all 384 features, with 3 features each computed from the normalized cumulative histogram i.e., 8 pairs x 16 histogram x 3 features. The method for histogram computation and feature extraction for one pair $(\boldsymbol{L}, \boldsymbol{D})$ and one angle i.e. 0° degree is presented below:

1. A pixel x in $\boldsymbol{L}$ and a pixel y in the corresponding location in $\boldsymbol{H}$ are shown in Figure 2.4 with their 8-nearest neighbours. The neighbouring pixel of x and y considered for co-occurrence computation are shown by the circles in Figure 2.4.

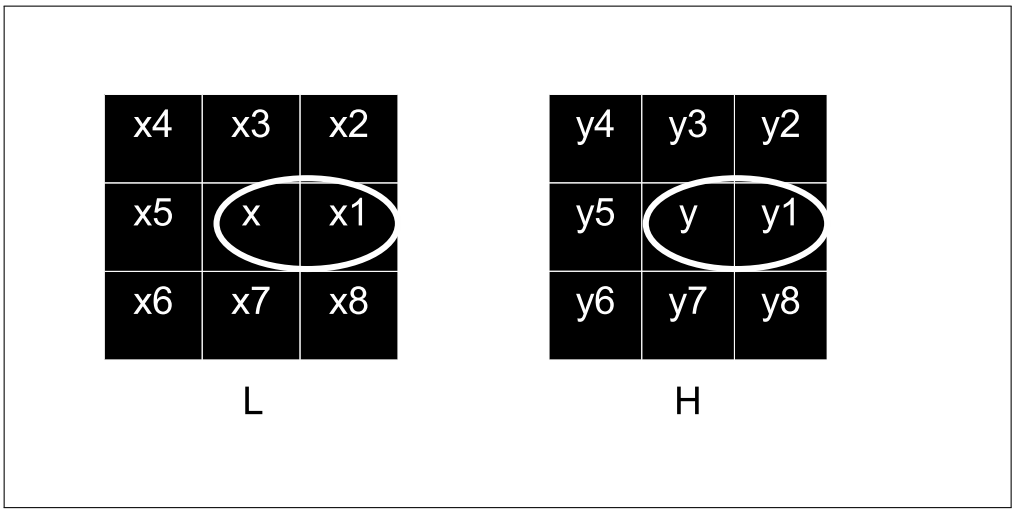


Figure 2.4: A pixel x in $\boldsymbol{L}$ and a pixel y in the corresponding location in $\boldsymbol{H}$

2. Construct two histograms $\boldsymbol{F_1}$ and $\boldsymbol{F_2}$ for L based on the maxmin composition rule stated below:

   $\alpha = \max(\min(x, h_i), \min(y, a_i))$
   $x \in \boldsymbol{F_1}, if \alpha = \min(x, h_i)$
   and
   $x \in \boldsymbol{F_2}, if \alpha = \min(y, a_i)$

3. Repeat steps 1 and 2 for all pixels x in $\boldsymbol{L}$.

4. Three features are then computed from every histogram as explained below:

(a) Consider a histogram $\boldsymbol{F}$.

(b) Obtain cumulative histogram($\boldsymbol{CH}$) for $\boldsymbol{F}$.

(c) Normalize $\boldsymbol{CH}$ yielding $\boldsymbol{NCH}$ (values between 1 and 0).

(d) The points on the $\boldsymbol{NCH}(nch_1, nch_2, ..., nch_{256})$, are the sample points.

(e) From the sample points, compute the following features:

- mean slope between 2 sample points of $\boldsymbol{NCH}$ in several areas, see Figure 2.5:

$$S_{\text{nch}} = \frac{1}{4}\sum_{i=1}^{4} \text{slope}_i$$

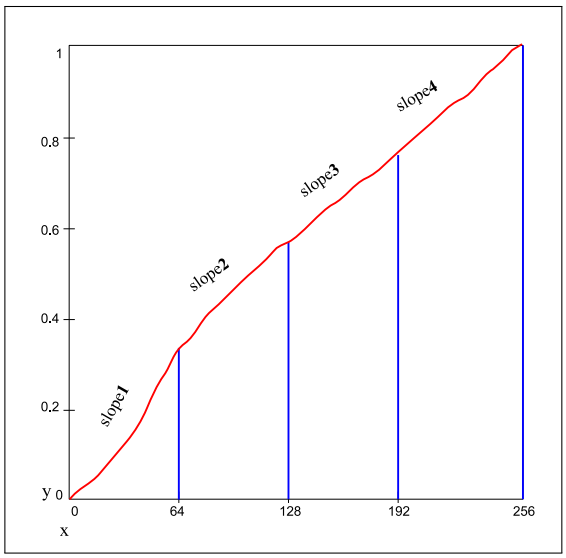


Figure 2.5: Normalized Cumulative Histogram with Slope Area Definition

- mean of the sample points of the $\boldsymbol{NCH}$:

$$\mu_{\text{nch}} = \frac{\sum_{i=1}^{256} \text{nch}_i}{256}$$

- mean deviation:

$$D_{\text{nch}} = \frac{\sum_{i=1}^{256} |\text{nch}_i - \mu_{\text{nch}}|}{256}$$

Due to the large number of features, which could cause overlap of the features and so to incorrect results, the 384 features are reduced by a Principal Component Analysis(PCA) using the covariance method. In addition, the scalability with respect to the training of a Support Vector Machine, which is explained in the next section, can be improved by this step. I.T Jolliffe[Jol86] described this technique well and defined it as:

> The central idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.

2.4 Classification

True to the trend of the related work the classification is done by a Support Vector Machine(SVM) using a Classification SVM Type 1, also known as C-SVM classification, with an RBF Kernel Function and a iteration of 1000000. This technique was well summarized by Chen [COB04]. The advantage of using thus technique is, that it is easier to train as other classification techniques, it needs fewer training samples and has better generalization ability. SVMs are motivated by statistical learning theory which have shown their ability to generalize well in high-dimensional space, such as those spanned by the texture patterns of characters. SVM was proposed by Vapnik [CV95]and obtained excellent results in various data classification in recent years especially in two class-problems, which is also the problem in this thesis. The Key idea of SVMs is to implicitly project the input space into a higher dimensional feature space where the two classes are more linearly separable. This projection, denoted ϕ, is implicit since the learning and decision process only involve an inner dot product in the feature space, which can be directly computed using a Kernel K defined on the input space. In short, given m labelled training samples: $(x_1, y_1), ..., (x_m, y_m)$, where $y_i = \pm 1$ indicates the positive and negative classes, and assuming there exists a hyperplane defined by $\omega, \phi(x) + b = 0$in the feature space separating the two classes, it can be shown that w can be expressed as a linear combination of the training samples i.e.

$\omega = \sum_j \lambda y_j \phi(x_j)$ with $\lambda_j \geq 0$. The Classification of an unknown sample z is thus based on the sign of the SVM function:

$$G(z) = \sum_{j=1}^{m} \lambda_j y_j \phi(x_j)\phi(z) + b \tag{2.9}$$

$$= \sum_{j=1}^{m} \lambda_j y_j K(x_j, z) + b, \tag{2.10}$$

where $K(x_j, z) = \phi(x_j)\phi(z)$ is called the kernel function. The training of an SVM consists of estimating the λ_j and b to find the hyperplane that maximizes the margin, which is defined as the sum of the shortest distance from the hyperplane to the closest positive and negative samples. The used SVM in the proposed approach, was trained with the texture features, as mentioned in the Feature Extraction step. The features were previously extracted from a dataset of 2500 non text and 5000 text images with the size of 100x100, which are manually extracted from the train database of the ICDAR2003[1] competition and are disjunct to the evaluation database. Some examples are shown in Figure 2.6 The reasons for the

Non Text Regions 100x100 Text Regions 100x100

Figure 2.6: Train Images SVM 1

selected ratio was the assumption that than more text features are available than better the classification would be. The results of the classification are still similar to the second step, but with one exception, all non text regions are eliminated, so that the result contains only real text regions.

2.5 Merging Regions

After the classification the results are still separated text regions, which represent only parts of text blocks and in fact, the goal is to get the whole text blocks as

[1] http://algoval.essex.ac.uk/icdar/Competitions.html

a result. For this reason the regions should be merged, to get the text blocks. Before the merging process can be applied, the regions should be resized by the value, which was subtracted from the width and height of the original image in the preprocessing. The reason for this is, that for the evaluation in Chapter 3 it is necessary to have the original X,Y coordinates, even to be able to evaluate the introduced approach. Once the regions are resized the merging step can be applied. This process is done by combining at first the regions in horizontal direction. Here is the rule, that only text regions are merged with the considered regions, even if a neighbour text region in horizontal direction has a smaller distance than 100 pixel and having the same X coordinates. If a distance is bigger than 100, the region is labelled as individual text region and is not merged with the considered region. The result of this step is shown in Figure 2.7 The next stage in the this process

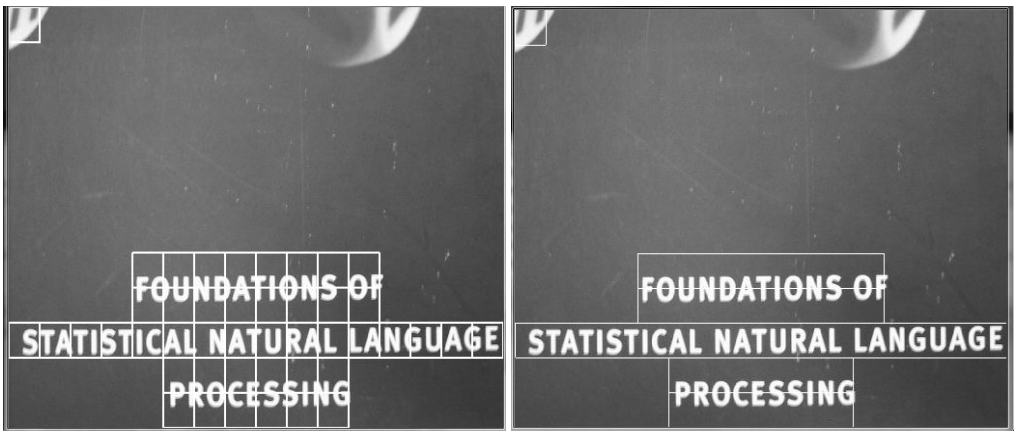


Figure 2.7: Horizontal Merging Result

is to merge text regions in the vertical direction, by the same principle as in the step before. Certainly this step must be applied several times. The reason for it is, that the regions represent not always perfect text lines or text blocks. It can also represent a truncated real text line or block, so that for example a region could contains only the half of a real text line. Furthermore, if two regions can be merged in vertical direction the max (X coordinate) and max (Y coordinate) of both are the resulted positions. Some experiments were carried out on this procedure with the result, that the vertical merging step must be done three times to get the desired text blocks. The result of this step is shown in Figure 2.8 and was actually the goal of this work. The reason for developing the next steps was, that in this research no ordinary evaluation process is available, for example databases,region matching

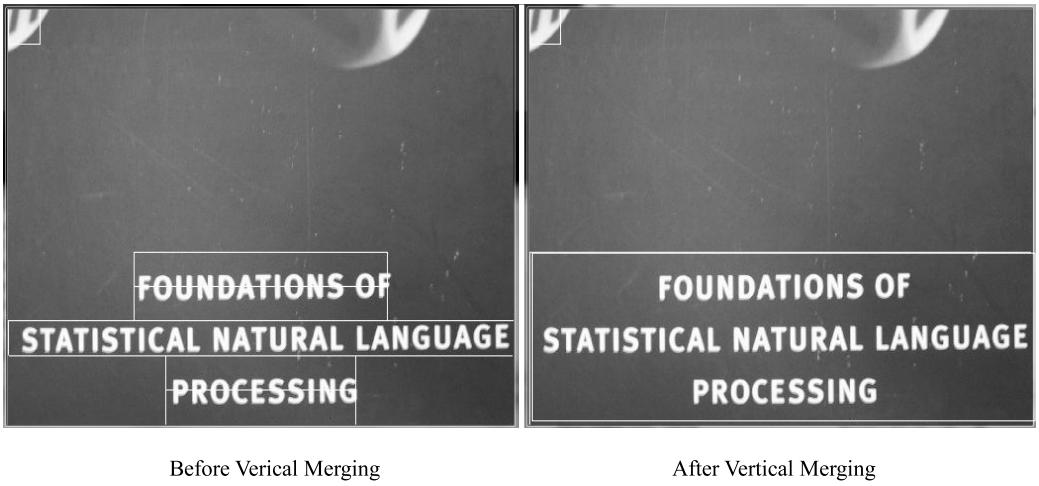


Figure 2.8: Vertical Merging Result

algorithms ect. Only the ICDAR2003[2] competition has published in addition to their results the necessary material for meaningful evaluation, which are described in more detail in Chapter 3. Certainly the goal of the competition differs from this thesis regarding to the detection. Their goal was, to detect individual words, so that the goal of the proposed approach changed and three postprocessing steps were developed. Namely text line extraction, word detection and refinement of the detected words. This steps are described in the next sections in more detail.

2.6 Text Line Extraction

As mentioned in the previous section the first postprocessing step is the text line extraction. The task here is to extract single text lines from a given text block, with the assumption of a horizontal text alignment. Here is also the property used, that text lines are in possession of many short edges, so that it is possible to distinguish the text lines from the background by a horizontal projection like in the segmentation step. To separate the three text lines in the example Figure 2.9 we need to find the $Valley$ on the projection profile where the profile value is smaller than a threshold T and then segment the three text lines at the $Valley$. The threshold T is calculated as shown in the following equation:

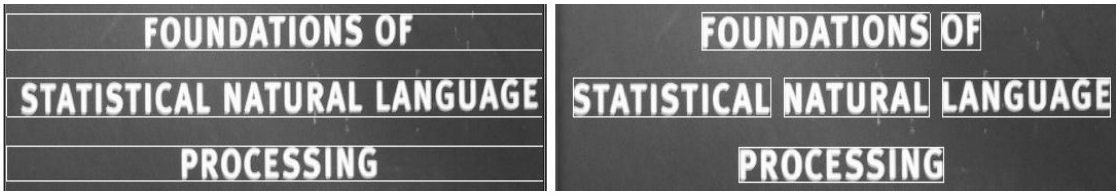
$$T = (\text{Mean} + \text{Min}) * 0.3 \tag{2.11}$$

[2]http://algoval.essex.ac.uk/icdar/Competitions.html

Figure 2.9: Text Line Extraction

where Min and Mean are the minimum value and the average value of the projection profile. This method was also used by Ye et al [YHGZ05]method.

2.7 Word Detection



Before Word Detection After Word Detection

Figure 2.10: Word Detection Result

The next step in this process cycle is to detect words from the extracted text lines. This step acts similar to the last step. Besides that the orientation of the projection is vertical. The threshold calculation is carried out as in the step before. But some experiments on this procedure resulted that the $Valley$ should be larger than 25% of the regions width. The reason for this behaviour is, if a bigger value would be selected the separation would not come off and the result would be a text line. Furthermore, if the value would be selected smaller this process would achieve a result at character level. And with the mentioned value the separation is successful in the most cases.

2.8 Refinement

The final step of the developed approach is the refinement. The reason for the last step is, that not only characters devise high sample points in the projections but

also similar textures like leaves, lines in traffic sign ect. To eliminate these false alarms, a second SVM is trained, which is specialized to distinguish text words from text like regions. The difference to the first SVM is only that other train images are used, namely images with different sizes, see Figure 2.11. With regard

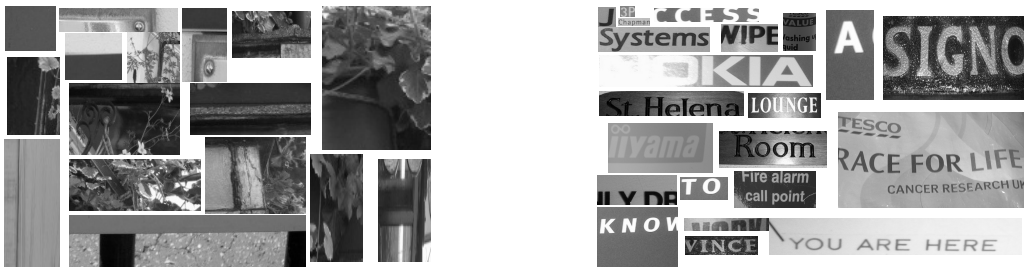


Figure 2.11: Train Images SVM 2

to the text images, only single words are containing. This SVM was trained with 1000 text and non text regions and was applied on the resulted regions from the whole approach. Only regions, which this SVM classify to a real text region belong to the final regions. This final regions have still to be proofed, with regard to simple structural information. Because, it is observed that text height should be larger than 15 pixels to be seen clearly by human, so that regions should be larger than 15 pixel. Furthermore the candidates whose $\frac{\text{width}}{\text{height}}$ is smaller than 1 is discarded as non text region. The region which are now labelled as text regions represent the contained text in the input image. Some results are shown in Figure 2.12.

Figure 2.12: Result Examples

Chapter 3

Experimental Results

First of all this chapter describes several difficulties with regard to performance evaluation in text detection. Furthermore this chapter introduce the ground truth, the matching algorithm and the framework, which were used to evaluate the proposed approach. Lately, the results are presented.

3.1 Difficulties

This section gives a summary of several difficulties regard to performance evaluation, not only in terms of text detection but also in nearly all research areas in computer vision and pattern recognition. The difficulties are well described by Jung *et al.* [JKJ04].

- **Ground Truth Data:** The degree of preciseness is difficult to define. This problem is related to the construction of the ground truth data. The ground truth data for text detection is usually marked by bounded rectangles. However, an algorithm is very accurate and detects text at character level thus will not have a good recall rate, if the ground truth data detect text at text block level.

- **Performance measure:** A decision has to be made on which measures to use in the matching process between results and ground truth data. Usually, is the recall and precision rates used. Furthermore, a method is needed for comparing the ground truth data and the output of the developed approach. There are several comparison possible: pixel by pixel, character by character, or rectangle by rectangle.

- **Application dependence:** The goal of text detection systems can differ, some require that all the text in an input image must be located, while others concentrate on detecting important text.

- **Database** Many researches seek public databases of images containing text, but it is difficult to find a general or domain specific comprehensive database, which has a ground truth data.

- **Output Format:** The output format of different algorithms may be different, which also make it difficult to compare their performance.

3.2 Ground Truth Data

The used ground truth and train database, are the same ones that were used in the ICDAR2003[1] text locating competition. An XML format is used for the detection results and for the ground truth. It is an extended version of the format developed for the ICDAR2003[2] text detection competition organized by Simon Lucas and his team[LPS+05]. Here is an example for a dataset containing the results on one image.

```
<?xml version="1.0" encoding="UTF-8"?>
<tagset>
  <image>
    <imageName>images/image1.jpg</imageName>
    <taggedRectangles>
      <taggedRectangle x="1276" y="900" width="193" height="61" />
      <taggedRectangle x="348" y="844" width="197" height="105" />
      <taggedRectangle x="776" y="812" width="281" height="165"
          modelType="1" />
    </taggedRectangles>
  </image>
</tagset>
```

Listing 3.1: Text detection output in XML

One rectangular bounding box is described by the tag <**taggedRectangle**>. Its geometry is described by x,y,width and height. The attribute **modelType** is a application depended numerical value which encodes the type or class of the object, is only necessary for multiple object recognition and is optional. The ground truth database itself consist of 495 images containing text and 6 without embed text. The total number of ground-truth rectangles in these images is 2261 and the resolution is mostly 1600x1200 but there are also images with a resolution of 338x255 and lower.

[1] http://algoval.essex.ac.uk/icdar/Competitions.html
[2] http://algoval.essex.ac.uk/icdar/Competitions.html

3.3 Implementation

The implementation occurs using Microsoft Visual Studio 2008 Professional and was written in C/C++. For image processing the OpenCv 2.1[3] library was used, which has more than 500 optimized and useful algorithms. Furthermore this library has an implementation of Support Vector Machine and Principal Component Analysis, which is used in the implemented framework. Due to the necessity of an XML parser and writer a simple library was integrated, namely TinyXML[4]. The whole approach was developed/implemented on an personal laptop with Intel(R) Core(TM)2 CPU T5500 @ 1.66GHz processor and 1,00 GB RAM.

3.4 Evaluation Algorithm

The algorithm for the evaluation was proposed by Wolf *et al.* [WJ06]. In short, the performance of a detection algorithm is illustrated intuitively by performance graphs which present object level precision and recall depending on constraints on detection quality. In order to compare different detection algorithms, a representative single performance value is computed from the graphs. The evaluation method can be applied to different types of object detection algorithms. It has been tested on different text detection algorithms, among which are the participants of the ICDAR2003[5] text detection competition. The recall and precision measures can be defined as follows:

$$R_{OB}(\boldsymbol{G}, \boldsymbol{D}, t_r, t_p) = \frac{\sum_i Match_G(\boldsymbol{G}_i, \boldsymbol{D}, t_r, t_p)}{|\boldsymbol{G}|} \tag{3.1}$$

$$P_{OB}(\boldsymbol{G}, \boldsymbol{D}, t_r, t_p) = \frac{\sum_i Match_D(\boldsymbol{D}_i, \boldsymbol{G}, t_r, t_p)}{|\boldsymbol{D}|} \tag{3.2}$$

where $\boldsymbol{D}$ is a vector of detected rectangles and $\boldsymbol{G}$ the ground truth rectangles. Furthermore $Match_G$ and $Match_D$ are functions which take into account the different types of matches and which evaluate to the quality of the match:

$$Match_G(\boldsymbol{G}_i, \boldsymbol{D}, t_r, t_p) = \begin{cases} 1, & \text{if } \boldsymbol{G}_i \text{ matches against a singledetected rectangle} \\ 0, & \text{if } \boldsymbol{G}_i \text{ does not match againstany detected rectangle} \\ f_{sc}(k), & \text{if } \boldsymbol{G}_i \text{ matches against several}(\rightarrow k)\text{detected rectangles} \end{cases} \tag{3.3}$$

[3] http://opencv.willowgarage.com/wiki/
[4] http://www.grinninglizard.com/tinyxml/
[5] http://algoval.essex.ac.uk/icdar/Competitions.html

$$Match_D(\boldsymbol{D}_i, \boldsymbol{G}, t_r, t_p) = \begin{cases} 1, & \text{if } \boldsymbol{D}_i \text{ matches against a single detected rectangle} \\ 0, & \text{if } \boldsymbol{D}_i \text{ does not match against any detected rectangle} \\ f_{sc}(k), & \text{if } \boldsymbol{D}_i \text{ matches against several}(\rightarrow k)\text{detected rectangles} \end{cases} \tag{3.4}$$

where $f_{sc}(k)$ is a parameter function of the evaluation scheme which controls the amount of punishment, which is inflicted in case of scattering, i.e. splits or merges. If it evaluates to 1, then no punishment is given, lower values punish more. In the experiments it set to a constant value of 0.8, which was also used during the ICDAR2003[6] text locating competition. The decision, whether a ground-truth rectangle $\boldsymbol{G}_i$ is matched against a detected rectangle $\boldsymbol{D}_i$ is taken based on the overlap information stored in two matrices $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ introduced by Liang *et al.* [LPH97], which corresponds intuitively to the surface recall and surface precision. The matrices are analysed in order to determine the correspondences between the two rectangle lists. In general, a non zero value in an element with indices (i, j) indicates, that ground truth rectangle $\boldsymbol{G}_i$ overlaps with result rectangle $\boldsymbol{D}_j$. However, the two rectangles are only matched if the overlap satisfies the quality constraints, i.e. if area recall and area precision are higher than the respective constraint:

$$\begin{aligned} &(a) \quad \boldsymbol{\sigma}_{ij} > t_r \\ &(b) \quad \boldsymbol{\tau}_{ij} < t_p \end{aligned} \tag{3.5}$$

where $t_r \in [0, 1]$ is the constraint on area recall and $t_p \in [0, 1]$ is the constraint on area precision. In detail, the different matches are determined as follows:

one-to-one matches: one ground truth rectangle $\boldsymbol{G}_i$ matches with a result rectangle $\boldsymbol{D}_j$ if row i of both matrices contains only one element satisfying 3.5 and column j of both matrices contains only one element satisfying 3.5.

one-to-many matches (splits): one ground truth rectangle $\boldsymbol{G}_i$ matches against a set $\boldsymbol{S}_o$ of result rectangles $\boldsymbol{D}_j, j \in \boldsymbol{S}_o$ if

- a sufficiently large proportion of the ground truth rectangle has been detected (condition 3.5(a) in a scattered version): $\sum_{j \in \boldsymbol{S}_o} \boldsymbol{\sigma}_{ij} \geq t_r$, and
- each contributing result rectangle overlaps enough with the ground truth rectangle to be considered a part of it (condition 3.5(b) in a scattered version): $\forall j \in \boldsymbol{S}_o : \boldsymbol{\tau}_{ij} \geq t_p$

many-to-one matches (merges): one result rectangle $\boldsymbol{D}_j$ matches against a set $\boldsymbol{S}_m$ of ground truth rectangles if

[6] http://algoval.essex.ac.uk/icdar/Competitions.html

- A sufficiently large portion of each ground truth rectangle is detected (condition 3.5(a) in a scattered version): $\forall i \in \boldsymbol{S}_m : \boldsymbol{\sigma}_{ij} \geq t_r$ and
- Each ground truth rectangle has been detected with enough area precision (condition 3.5(b) in a scattered version): $\sum_{i \in \boldsymbol{S}_m} \boldsymbol{\tau}_{ij} \geq t_p$

The case many to many was not taken into account on the ground that it may never happen. For the evaluation the threshold values of $t_r = 0.8$ and $t_p = 0.4$ are used which were also chosen by the ICDAR2003[7] competition. In the creation of graphs there are two different cases therefore the results are two graphs: either the t_r is fixed and t_p is increased or the other way. Furthermore the introduced evaluation algorithm include also a single performance value, either for direct comparison of the performances of different algorithms or to optimize the parameters of the detection algorithm ect. Therefore Wolf *et al.* [WJ06] proposed the proportion of the graph area, which is beneath the performance graphs as a reliable and objective measure. This is equivalent to the mean value of object measures over all possible constraint values. First the area proportion is calculated separately for object recall Equation 3.6 and object precision Equation 3.7

$$\begin{aligned} R_{OV} = & \frac{1}{2T} \sum_{i=1}^{T} R_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, \frac{i}{T}, t_p) + \\ & + \frac{1}{2T} \sum_{i=1}^{T} R_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, t_r, \frac{i}{T}) \end{aligned} \tag{3.6}$$

$$\begin{aligned} P_{OV} = & \frac{1}{2T} \sum_{i=1}^{T} P_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, \frac{i}{T}, t_p) + \\ & + \frac{1}{2T} \sum_{i=1}^{T} P_{OB}(\overline{\boldsymbol{G}}, \overline{\boldsymbol{D}}, t_r, \frac{i}{T}) \end{aligned} \tag{3.7}$$

The parameter T is a granularity parameter which controls the trade-off between the computational complexity of the evaluation algorithm and the precision of the integration approximation. The default value of this parameter is set to T=20.

$$H_{mean} = 2 \frac{P_{OV} R_{OV}}{P_{OV} + R_{OV}} \tag{3.8}$$

The final performance value is the harmonic mean Equation 3.8 of the two measures equation.

[7] http://algoval.essex.ac.uk/icdar/Competitions.html

3.5 Evaluation Software

To evaluate the proposed approach a software is used named DetEval[8], which applies the algorithm described above. DetEval receives as input XML Files with the results of detection as well as the ground truth information, which are structured according to the schema described in Section 3.2. After employing the evaluation algorithm introduced above, the results can be plotted or written into a file. DetEval is available in two versions and can be used under the terms of the GNU. The GUI version is sufficient for most cases and the command line version, which is used in this thesis for the evaluation, allows more control on the evaluation process.

3.6 Results

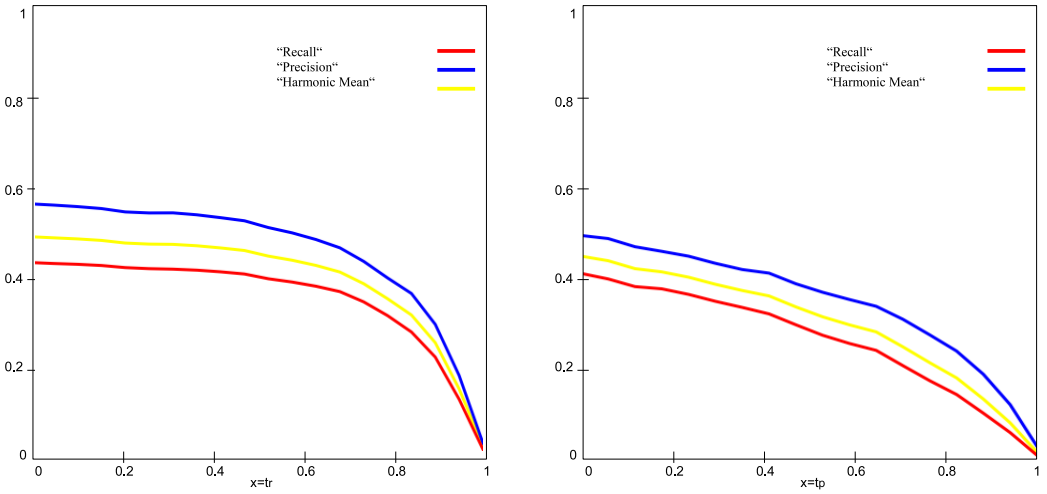
The proposed approach was applied to the database of images, which were used during the ICDAR2003[9] text detection competition. The results were saved in a single XML File, and structured according to the prescribed scheme. The evaluation occurs using the proposed evaluation software with the prescribed metrics and parameters. The Table 3.1 shows the results of the proposed approach and

Algorithm	**Recall**	**Precision**	H_{mean}	**Detected Regions**	**t(s)**
Ashida	41.7	55.3	47.5	1916	8.7
H.W.David	46.6	39.6	42.8	1515	0.3
Wolf *et al.*	44.9	19.4	27.1	3477	17
Todoran	17.9	14.3	15.9	1368	0.3
Proposed	**33.2**	**40.4**	**36.4**	**1180**	**1.2**

Table 3.1: Results with Wolfs Evaluation Algorithm

the algorithms which were evaluated during the ICDAR2003[10] text detection competition. The column labelled t(s) gives the average time in seconds to process each image. The developed method achieves a Recall of 33.2, Precision of 40.4 and H_{mean} value of 36.4. So that the proposed approach would have achieved the third place in the competition. The reasons for the relative bad results can be refer to the last steps of the procedure. From the moment the procedure begins to extract text lines from text blocks many false alarms arise, which are not fully eliminated by the final step in the process cycle. The reason for this behaviour could be, that

[8] http://liris.cnrs.fr/christian.wolf/software/deteval/index.html
[9] http://algoval.essex.ac.uk/icdar/Competitions.html
[10] http://algoval.essex.ac.uk/icdar/Competitions.html



(a) t_r Graph **(b)** t_p Graphs

Figure 3.1: Resulted Graphs

a false alarm appear in all sizes, so that many overlaps can arise in relation to the features. If the goal for the evaluation would be to detect text blocks, the results would be much better. During the experiments it was evident that the algorithm for feature extraction has the potential for specialization in a certain kind of text occurrences, if the textures are more or less similar. For the challenge to classify all types of text, what means to train the SVM with much more images, is possible. But to train with more images could lead to overlaps of the features and whether this leads to better evaluation results is doubtful. Looking at the resulted graphs Figure 3.1 (a) and Figure 3.1 (b) one important property can be read off, the values decrease to zero by fixed t_r and increasing t_p or fixed t_p and increasing t_r i.e. that the resulted regions have in the most cases a greater region than in the ground truth. In addition to this the behaviour show also, that the detected regions are mostly complete detected and not partly. Furthermore the graphs show that in the most cases the detected regions containing text are detected with a fair accuracy. With regard to the constraint t_r it can be red off in Figure 3.1 (a), that over a larger area($\frac{2}{3}$) a value of of about 40 and more is achieved i.e. that in the most images the recall and precision are about the value 40. Additionally, for 23 images a precision and recall of 100 was achieved. Moreover the low precision can be attributed to the large number of false positives which are not totally eliminated by the last step of the proposed approach. The reason for this is, that the most false positive regions represent objects which have similar textures and entails high contrast, for example leaves with sky as background or fences. The

most difficulties for the proposed approach was to detect text on reflective and transparent objects like windows, but also to detect text with similar color like the background. Another behaviour is that the proposed approach can be reduced to a region-based method without the usage of Support Vector Machines, at the expense of precision. The results are shown in Table 3.2. Looking at the results,

Algorithm	**Recall**	**Precision**	H_{mean}	**Detected Regions**	**t(s)**
Ashida	41.7	55.3	47.5	1916	8.7
H.W.David	46.6	39.6	42.8	1515	0.3
Wolf *et al.*	44.9	19.4	27.1	3477	17
Todoran	17.9	14.3	15.9	1368	0.3
Proposed	**39.3**	**23.7**	**29.6**	**2476**	**0.1**

Table 3.2: Results without Classification

good H_{mean} value and a slightly better recall can be read off and the precision decreases drastically to 23.7, the reason for this results are too many false alarms. The detected regions show us that almost a doubled amount of regions as the proposed approach detect and more regions as in the ground truth exists i.e. that the usage of classification eliminates almost 1000 false regions, so that the usage of classification is here necessary, even if the computing time increases by one second. By and large, the proposed approach is a successful method which has potential i.e. it can be improved considerably, for example by a better choice of training images.

Chapter 4

Conclusion

4.1 Summary

The goal of this bachelor thesis was to develop and to evaluate a novel texture-based approach for word detection in digital images. Roughly the developed approach is using Support Vector Machines, Wavelet-Based features and edge projections in differ direction. In more detail it can be divided into the following processes: At first a sliding window with the size of 100x100 iterate over an input image in 100 pixel steps. Based on a horizontal projection of an edge image it is decided, whether an actual window is a possible text region or not. If a window was labelled as a possible text region, texture features are extracted from this region and note to them a classification is performed by an Support Vector Machine. The features are extracted by applying an algorithm using the first level wavelet decomposition and co-occurrence histograms. This algorithm produces 384 features which are reduced to 45 by a principal component analysis. The classification produces text blocks which are embedded in an input image. But the task is to detect individual words, so there are further processes developed. The next steps are text line detection and individual word extraction. The text line detection is executed by an horizontal projection. This detection is necessary to be able to extract individual words, because the word extraction is applied on the detected text lines by an vertical projection. During these steps some false positives arise, which are eliminated in the last stage by a second Support Vector Machine and by simple structural information. Indeed, the last process eliminates not all false positive, because these regions various in size and this behaviour leads to overlap of extracted features and it is the reason for the low precision and recall values. Generally speaking the proposed approach is a possible solution for the task word detection with potential. The best results were achieved on images with text and

simple background. How this approach can be improved is presented in the last section of this thesis.

4.2 Future Work

This section will present ideas how this approach can be improved by further work. First of all, the text line extraction and word detection can be replaced by an Connected Component method. The reason for this is, that in most cases the detected text blocks represent the embedded text in the whole image, so that most of the text block is text which can be distinguish from the background by an Connected Component method. Second, the exact amount of features for the best results can be examined, what should lead to better results. Furthermore, if better texture features can be find they can be replaced.

Bibliography

[COB04] Chen, D.T. ; Odobez, J.M. ; Bourlard, H.: Text detection and recognition in images and video frames. 37 (2004), March, Nr. 3, S. 595–608

[CV95] Cortes, Corinna ; Vapnik, Vladimir: Support-Vector Networks. In: *Machine Learning* (1995), S. 273–297

[FK08] Farhoodi, Roshanak ; Kasaei, Shohreh: Abstract Text Segmentation from Images with Textured ans Colored Background. (2008). `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.91.7060`

[Gll07] Gllavata, Julinda: *Extracting Textual Information from Images and Videos for Automatic Content-Based Annotation and Retrieval*, Fachbereich Mathematik und Informatik der Philipps-Universitaet Marburg, Dissertation, 2007. `http://archiv.ub.uni-marburg.de/diss/z2007/0107/pdf/djg.pdf`

[JKJ04] Jung, Keechul ; Kim, Kwang I. ; Jain, Anil K.: Text information extraction in images and video: a survey. In: *Pattern Recognition* 37 (2004), Nr. 5, S. 977–997

[Jol86] Jolliffe, Ian T.: *Principal Component Analysis*. Springer-Verlag, http://www.springer.com, 1986

[JQXW06] Jiang, Renjie ; Qi, Feihu ; Xu, Li ; Wu, Guorong: Detecting and Segmenting Text from Natural Scenes with 2-Stage Classification. In: *ISDA '06: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*. Washington, DC, USA : IEEE Computer Society, 2006. – ISBN 0–7695–2528–8, S. 819–824

[LPH97] Liang, Jisheng ; Phillips, Ihsin T. ; Haralick, Robert M.: *Performance evaluation of document layout analysis algorithms on the UW data set*. 1997

[LPS+05] Lucas, Simon M. ; Panaretos, Alex ; Sosa, Luis ; Wong, Anthony Tang S. ; Ashida, Kazuki ; Nagai, Hiroki ; Okamoto, Masayuki ; Yamamoto, Hiroaki ; Miyao, Hidetoshi ; Zhu, Junmin ; Ou, Wuwen ; Wolf, Christian ; Jolion, Jean michel ; Todoran, Leon ; Worring, Marcel ; Lin, Xiaofan: X.: ICDAR 2003 robust reading competitions: entries, results and future directions. In: *International Journal on Document Analysis and Recognition - Special Issue on Camera-based Text and Document Recognition 7(2-3)*, 2005, S. 105–122

[PS06] P.S.Hiremath ; S.Shivashankar: Wavelet Based Features for Texture Classification. In: *ICGST International Journal on Graphics, Vision and Image Processing* 6 (2006), S. 55–58

[SDB98] Shim, Jae-Chang ; Dorai, Chitra ; Bolle, Ruud: Automatic Text Extraction from Video for Content-Based Annotation and Retrieval. In: *ICPR '98: Proceedings of the 14th International Conference on Pattern Recognition-Volume 1.* Washington, DC, USA : IEEE Computer Society, 1998. – ISBN 0–8186–8512–3, S. 618

[WJ06] Wolf, C. ; Jolion, J.-M.: Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms. In: *International Journal on Document Analysis and Recognition* 8 (2006), Nr. 4, S. 280–296

[YHGZ05] Ye, Q.X. ; Huang, Q.M. ; Gao, W. ; Zhao, D.B.: Fast and robust text detection in images and video frames. 23 (2005), June, Nr. 6, S. 565–576