Design and evaluation of features that best define text in complex scene images

Navid Mavaddat
School of Computer Science
and Software Engineering
University of Western Australia
Crawley 6009, Australia
navid@csse.uwa.edu.au

Tae-Kyun Kim
Sidney Sussex College
University of Cambridge
Cambridge CB2 3HU, UK
tkk22@cam.ac.uk

Roberto Cipolla
Department of Engineering
University of Cambridge
Cambridge CB2 1PZ, UK
cipolla@cam.ac.uk

Abstract

In this paper we explore features for text detection within images of scenes containing other background objects using a Support Vector Machine (SVM) algorithm. In our approach, the Haar-like features are designed and utilised on banks of bandpass filters and phase congruency edge maps. The designed features with SVM leverages the properties of text geometry and colour for better differentiation of text from its background in real world scenes. We also evaluate the contributions of the features to text detection by the SVM coefficients, which leads to time-efficient detection by using an optimal subset of features.

1 Introduction

The recognition of text has numerous applications such as assisted reading for the visually impaired, machine language translation [1, 2], and machine reading for robotic and automated systems that need to interact in a human oriented environment, such as intelligent vehicle driving systems [3]. Text recognition in cluttered images poses three challenges: firstly to detect areas of text amongst other objects; secondly, the rectification of text and finally, text recognition by OCR. Within the class of text objects in images of natural scenes there is a large intra-class variability. Text can be found in many sizes which are not necessarily related to the size of nearby objects or cues. Text can be of many different fonts, thickness, styles and colours. Though many characters share common features, the character set in most languages is large as a recognisable difference in characters is a necessity for distinguishing between letters. Text in natural scenes could be distorted in imaging due to perspective and affine distortion, a consequence of the imaging process and also as a result of the geometric layout of the text itself. Once the position of the text is identified and its boundaries defined, the text can be rectified to remove all distortions. Subsequently, any common OCR system can read the text. In this work we focus on the detection of text in complex scenes.

There are a number of approaches for detecting text in images. Many studies have focused on text detection on the images of documents taken by a scanner or camera [4]. Such images have very simple backgrounds. Various techniques have been also developed for text detection in cluttered scenes, but their performance typically relies on image segmentation [5]. When text strokes appear narrow or in low contrast to their background, it is hard to obtain good segmented text regions for text detection in the previous methods. Chen and Yuille have tackled challenging scenes where there is text in many different sizes and background is highly cluttered [6]. It has been shown that the Haar-like box features in a boosted classifier yield accurate and fast text detection.

Here we examine a specific method that uses an SVM algorithm and the Haar-features suggested by Chen and Yuille [6], extend them with the inclusion of banks of bandpass filters and edge maps in order to evaluate features that best define text. We show that using the extended features greatly improves the text detection accuracy. Evaluation of the features has been performed by the feature histograms and the SVM coefficients. Our method has been designed with Latin characters in mind but can easily adapted to learn other character systems.

The remainder is organised as follows: Section 2 briefly reviews SVM algorithm. The proposed input channels, feature extraction and selection methods are explained in Section 3,4 and 5 respectively. Section 6 presents the SVM implementation. Results are given in Section 7 and conclusions are drawn in Section 8.

2 SVM algorithm

The SVM algorithm learns a classifier that will allow us to classify text from non-text samples that have been vectorised into an array of n-features. We will describe our process of feature extraction in the following section after we give an overview of the SVM algorithm. Given a set of text and non-text feature vectors we can describe the SVM algorithm as follows: let $\{\mathbf{f}^l, g^l\}_{l=1}^m$ with $\mathbf{f}^l \in \mathbb{R}^n$ and g^l a scalar assuming values of either $+1$ or -1, represent m sample points in an n-dimensional feature space $\boldsymbol{\phi}$. We designate $+1$ to identify text samples, while -1 is for non-text samples. SVM finds an optimum *hyperplane* ϕ that divides the feature space $\boldsymbol{\phi}$ into two regions that separates all the $\mathbf{f}^l$ points with $(g^l = +1)$ from all the other points with $(g^l = -1)$. The hyperplane ϕ can be represented by a normal vector $\boldsymbol{\omega} \in \mathbb{R}^n$ together with a constant b such that $|b|$ is proportional to the distance of ϕ from the origin. The closest sample points on either side of ϕ are referred to as *support vectors*.

To find ϕ such that the distances to the support vectors are maximised, it can be shown that $\frac{1}{2}\boldsymbol{\omega} \cdot \boldsymbol{\omega}$ should be minimised. This is equivalent to maximising the Lagrangian:

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha^l - \frac{1}{2} \sum_{l=1}^{m} \sum_{k=1}^{m} \alpha^l \alpha^k g^l g^k K(\mathbf{f}^l . \mathbf{f}^k) \qquad (1)$$

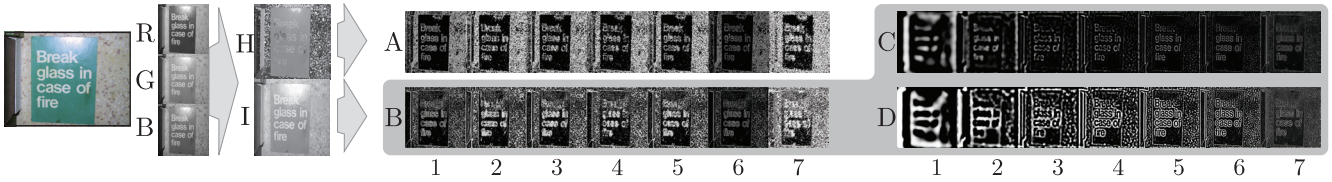

Figure 1: Images for training and testing (left) consist of red, green and blue channels which are then transformed into hue and intensity channels (centre). Filters are applied to the channels to produce 28 new channels (right).

with respect to Lagrangian multipliers α^l ($l \in \{1 \dots m\}, \alpha^l \geq 0$). $K(\mathbf{f}^l \cdot \mathbf{f}^k)$ is known as the *kernel function*. The kernel function can take many forms, for example a *linear kernel* is defined as $K(\mathbf{f}^l \cdot \mathbf{f}^k) = \mathbf{f}^l \cdot \mathbf{f}^k$ while a *gaussian kernel* takes the form $K(\mathbf{f}^l \cdot \mathbf{f}^k) = \exp\left(-||\mathbf{f}^l \cdot \mathbf{f}^k||/2\sigma^2\right)$. Polynomial and other non-linear kernels are possible.

It is evident in equation (1) that the training features set and its labels determine $\boldsymbol{\alpha}$. The value of $\boldsymbol{\alpha}$ from the maximisation determines the hyperplane with

$$\boldsymbol{\omega} = \sum_{l=1}^{m} \alpha^l g^l \mathbf{f}^l \text{ and } b = \boldsymbol{\omega} \cdot \mathbf{f}^l - g^l + \frac{\alpha_l}{2C} \text{for all } C \geq \alpha^l > 0. \quad (2)$$

The constant C is introduced as a *slack variable* that defines the degree of allowable classification error.

Any test point $\mathbf{f}^t$ can be assigned a positive or negative class label by determining which side of the hyperplane it lies in.

3 Input channels

We propose a 'patch-based' text detector to identify regions of high text likelihood in an image. We use a multi channel approach to exploit geometric and textural characteristics of text such as edges, corners, strokes and the curves that make up the geometry of text. The apparent 'texture' of text in comparison to its background is another useful characteristic. This is a three stage process consisting of: sample preparation, feature extraction and finally classification.

To make use of these characteristics we filter the training database images by a set of 28 filters. We refer to each filtered image as an *input channel*. The input channels we have utilised are summarised in table 1 and an example of a filtered image is shown in figure 1.

The input channel set consists of both banks of bandpass filters and banks of phase congruency edge maps. Bandpass filters are used to promote the prominence of text stroke regions in the input images.

Phase congruency is used because as an edge detection algorithm it is particularly robust against changes in illumination and contrast [7], both of which are highly variable in images of text. Local maximal phase congruency in images indicates edges which are a highly significant feature in text. During the calculation of maximal phase congruency, directional and minimal phase congruency can also be calculated with minimal further computation. Minimal phase congruency indicates corners which are a fundamental characteristic in text, while directional phase congruency maps assist by giving further information about text stroke qualities.

It should be noted that colour images are considered in terms of both hue and intensity. Text is mainly defined against its background. The contrast between foreground and background is often manifested as a hue difference instead of, or in addition to, intensity differences.

Table 1: Input channels

Channel A1	Phase congruency maxima of hue values
Channels A2–A5	Directional phase congruencies of hue values (four directions – $\{\uparrow, \nwarrow, \leftarrow, \swarrow\}$)
Channel A6	Phase congruency minima of hue values
Channel A7	Phase congruency orientations (in radians) of hue values
Channels B1–B7	As above but of intensity instead of hue values
Channels C1–C7	7 bands of bandpass filters applied to intensity values
Channels D1–D7	As above but applied to $(1 - intensity)$ values

4 Feature extraction

In the next step we extract and vectorise a set of features for each sample patch. To do this we apply Haar-like box filters to each of the sample patches. We make use of the

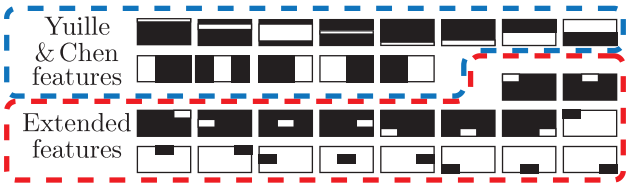

Figure 2: Haar-like box features.

box features suggested by Chen and Yuille [6] but extended with the inclusion of further filters designed to extract less prominent text characteristics. A set of box filter responses is determined from each of the 28 input channels extracted for each sample.

The 164 features are derived from the 31 box filters shown in figure 2. The feature set for each channel is listed in table 2.

Haar-like features are widely used and their efficacy is well documented. We have chosen to use means of box filters as they give a good representation of the response of the channel filters in various blocks of the sample patch. In addition they are very quick to calculate when utilising integral image techniques [8]. Standard deviation features are used as they are an alternative representation of the response of the channel filter blocks of the sample patch. Thus for each sample patch both an integral image and an

Table 2: Feature definitions

Features 1-24	Differences of mean and standard deviation features based on Yuille and Chen box features
Features 24-82	Differences of mean and standard deviation features of 18 blocks, denoted as 'Extended features' in figure 2 (intended to help learn localisation)
Features 83-164	The negative values of features (1-82)

integral square image is calculated for each of its channels ($28 \times 2 \times N^{o}\ samples$ in all). Though requiring more computation than mean features, quick calculation can still be performed by using integral squared images. Since standard deviation is defined as $\sum_{i=1}^{N}(x_i - \bar{x})^2$ and $\sum_{i=1}^{N}(x_i - \bar{x})^2 \equiv [\sum_{i=1}^{N} x_i^2] - N\bar{x}^2$ we can make its calculation more efficient by pre-calculating $\sum_{i=1}^{N} x_i^2$ as an integral image from squared pixel values.

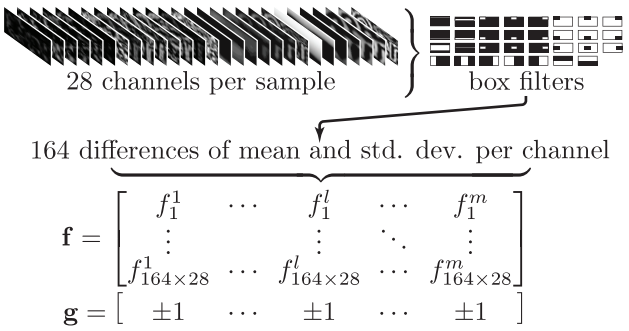

Figure 3: Feature responses are derived from sample patches taken across 28 channels. From the patches Haar-like box filters yielding mean and standard deviations. The differences of combinations of boxes results in 164 feature responses per channel. The total number of features calculated for each sample patch is 4592.

5 Feature Evaluation

The benefit of particular features or channels of features can be evaluated in a number of ways. We will consider two of these: feature histograms and SVM 'weightings'.

To evaluate a single feature in isolation, we can create histograms of the positive and negative responses of the test set samples for that feature. By taking the Difference Root Mean Square (DRMS) of the positive and negative histograms, it is possible to gauge the relative separability of a particular feature in comparison to other features. Figure 4 shows the DRMS values of the features calculated from our training dataset. Since SVM allows features to contribute in combination and not just individually, it is useful to evaluate the contribution of particular features to the SVM classifier as a whole.

We define a coefficient

$$W_i = \sum_{l=1}^{m} \alpha^l |f_i^l|. \tag{3}$$

which provides an indicator of the contribution of each particular feature dimension i to the definition of plane ϕ.

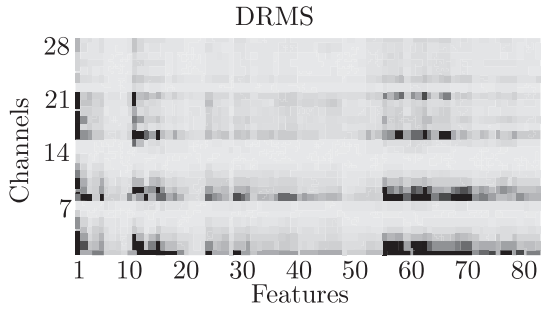

Figure 4: DRMS of features arranged per channel (features 1-82 of each channel shown). Darker boxes indicate greater separability.

Therefore features with high values of W_i play a larger role

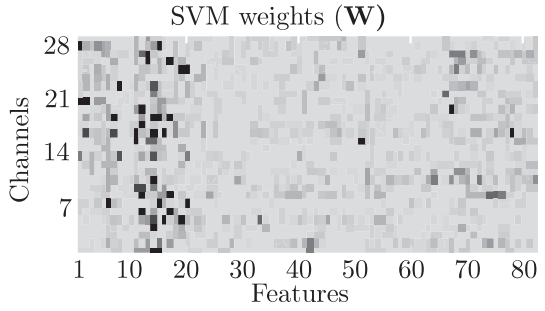

Figure 5: SVM weightings arranged per channel (features 1-82 of each channel shown). Darker boxes indicate greater separability.

in determining the classifier and can be considered more useful features. In figure 5 we have arranged the features in channels for easy comparison. We observe that all channels make contributions to the final classifier, whereas some box filter types are weaker than others. It can be noted that standard deviation features contribute more to the classifier than the mean features.

6 SVM implementation

For the implementation of text detection using SVM we require a database of labelled training data. In our implementation, the database we use consists of images with text regions labelled by bounding boxes. The bounding boxes are parallel to the image edges and thus may contain areas of non-text if the text is rotated with respect to the bounding box.

Positive (text) and negative (non-text) patches are sampled from the training database images. Positive patches are gathered by taking a sliding window across the bounding boxes of text with a fixed overlap (we have fixed our overlap to 30%). Negative patches are sampled randomly across the image at various randomly determined sizes, but do not contain or intersect any text-bounding-box beyond a defined overlap area (see figure 6). Subsequently the feature responses are normalised to bring the mean of each feature to 0 and its standard deviation to 1. These normalised features are fed to the SVM algorithm along with their class (text or non-text) labels. We use the SVM algorithm discussed in section 2 to create classifier model from the training data. In our work we make use of a linear kernel, as it is quickly computed and as we will show later, together with

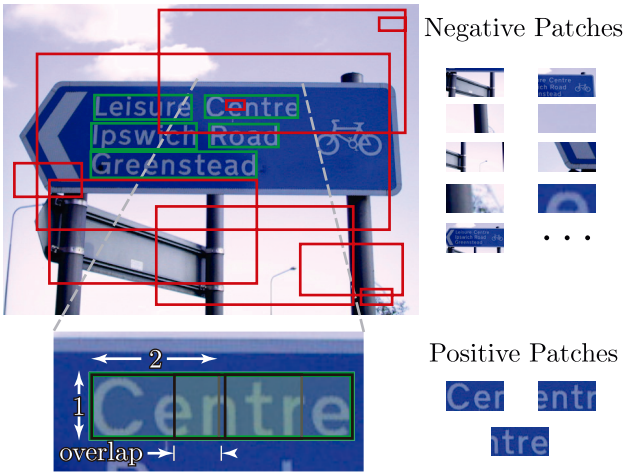

Figure 6: Sampling of positive and negative patches

the use of a soft margin, we can find a good separation for our data. The model can now be used to classify test image patches as text or non-text. The training and testing stages we have discussed are summarised in figure 7.

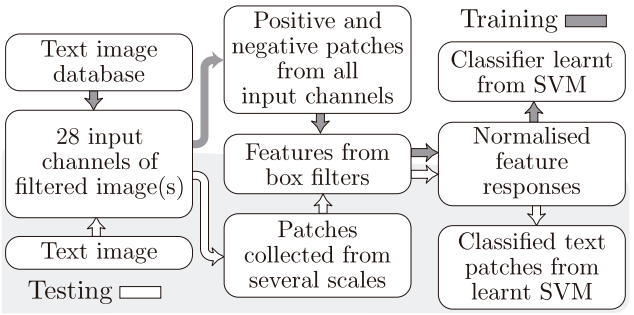

Figure 7: Training and testing stages in the patch based text detector.

7 Results

Training data are obtained from a database of images containing text as used by the ICDAR reading competition [9]. 125 images are randomly chosen from the training set and used for training. All 250 of the testing set images are used for testing. We evaluate the efficacy of our trained

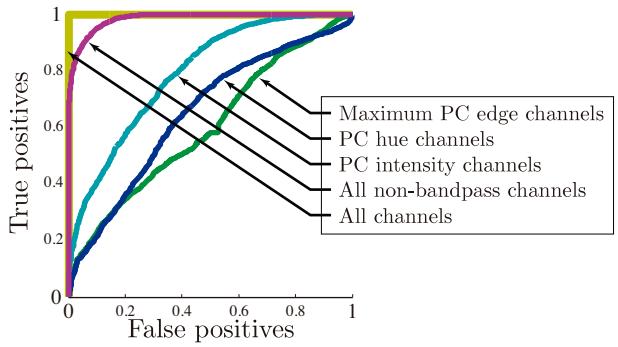

Figure 8: ROC curves for combinations of channels generated wholly from the test data using our trained model.

classifier by examining the Receiver Operating Characteristic (ROC) curve. The ROC curve is a plot of true positives versus false positives as the decision threshold is varied and is calculated from classifying patches taken from the test as distinct from the training dataset. The test patches were extracted in the same manner as the training patches. As we can see in figure 8, by training the classifier with different combinations of channels, it is evident that as we increase the number of channels used the quality of the classifier improves significantly. The maximum phase congruency channel can be considered similar to the edge detected input images used by Chen and Yuille [6]. Figure 9 shows the correctly classified text patches, taken from our testing dataset, using firstly the multi-channel approach we have proposed and then only the single maximum phase congruency channel.

Figure 9: Detected text patches, shown by white bounding boxes, using the multi-channel classifier (left) and single channel, maximum phase congruency classifier (right).

8 Conclusion

We have implemented a text patch classifier using filters and features specifically designed to extract text components. Our classifier was learnt through an implementation of the SVM algorithm. We can conclude that targeted multiple image channels improve text classification noticeably. By observing the SVM weightings we can see that features across most channels contribute to the final classifier and thus assist in the detection and localisation of the text we are seeking to recognise.

References

[1] Myers, G., Bolles, R., Luong, Q.T., Herson, J.: Recognition of text in 3-d scenes. In: Fourth Symposium on Document Image Understanding Technology, Maryland, Columbia (2001) 85–99

[2] Gao, J., Yang, J., Zhang, Y., Waibel, A.: Text detection and translation from natural scenes. Technical report, Carnegie Mellon University (2001)

[3] General Motors Corporation: Opel eye camera reads signs, improves safety. Website (2008) http://media.gm.com.

[4] Doermann, D., Liang, J., Li, H.: Progress in camera-based document image analysis. In: DAR '03. Proceedings. 7th International Conference on. (2003) 606–616

[5] Jiang, R., Qi, F., Xu, L., Wu, G.: Detecting and segmenting text from natural scenes with 2-stage classification. ISDA '06 **2** (Oct. 2006) 819–824

[6] Chen, X., Yuille, A.: Detecting and reading text in natural scenes. In: CVPR 2004. Volume 2. (2004) 366–373

[7] Kovesi, P.: Edges are not just steps. In: ACCV 2002, Melbourne Australia (2002) 822–827

[8] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR 2001. Volume 1. (2001)

[9] : ICDAR robust reading competitions image database (2003) http://algoval.essex.ac.uk/icdar/Datasets.html.