University of Groningen

Multi-script text versus non-text classification of regions in scene images

Sriman, Bowornrat; Schomaker, Lambert

Published in:
Journal of Visual Communication and Image Representation

DOI:
10.1016/j.jvcir.2019.04.007

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Final author's version (accepted by publisher, after peer review)

Publication date:
2019

Link to publication in University of Groningen/UMCG research database

Citation for published version (APA):
Sriman, B., & Schomaker, L. (2019). Multi-script text versus non-text classification of regions in scene images. *Journal of Visual Communication and Image Representation*, *62*, 23-42. https://doi.org/10.1016/j.jvcir.2019.04.007

Copyright
Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment.

Take-down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): http://www.rug.nl/research/portal. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Download date: 04-06-2022

Accepted Manuscript

Multi-script text versus non-text classification of regions in scene images

Bowornrat Sriman, Lambert Schomaker

PII: S1047-3203(19)30140-3
DOI: https://doi.org/10.1016/j.jvcir.2019.04.007
Reference: YJVCI 2531

To appear in: *J. Vis. Commun. Image R.*

Received Date: 15 November 2018
Accepted Date: 13 April 2019

Please cite this article as: B. Sriman, L. Schomaker, Multi-script text versus non-text classification of regions in scene images, *J. Vis. Commun. Image R.* (2019), doi: https://doi.org/10.1016/j.jvcir.2019.04.007

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Contents lists available at ScienceDirect
Journal of Visual Communication and Image Representation
journal homepage: www.elsevier.com/locate/jvci

Multi-script text versus non-text classification of regions in scene images

Bowornrat Sriman[a,*], Lambert Schomaker[a,1]

[a]*Artificial Intelligence, University of Groningen, Bernoulliborg, Nijenborgh 9, 9747 AG, Groningen,The Netherlands*

ARTICLE INFO

Article history:

Keywords:
Text detection in scene image,
Text/non-text classification,
Color features,
Color histogram autocorrelation

ABSTRACT

Text versus non-text region classification is an essential but difficult step in scene-image analysis due to the considerable shape complexity of text and background patterns. There exists a high probability of confusion between background elements and letter parts. This paper proposes a feature-based classification of image blocks using the color autocorrelation histogram (CAH) and the scale-invariant feature transform (SIFT) algorithm, yielding a combined scale and color-invariant feature suitable for scene-text classification. For the evaluation, features were extracted from different color spaces, applying color-histogram autocorrelation. The color features are adjoined with a SIFT descriptor. Parameter tuning is performed and evaluated. For the classification, a standard nearest-neighbor (1NN) and a support-vector machine (SVM) were compared. The proposed method appears to perform robustly and is especially suitable for Asian scripts such as Kannada and Thai, where urban scene-text fonts are characterized by a high curvature and salient color variations.

© 2019 Elsevier B. V. All rights reserved.

ACCEPTED MANUSCRIPT

1. Introduction

Text detection and recognition provides useful enrichment in many computer-vision applications. The function has been advocated as an aid for persons with a reading impairment or tourists needing automatic translations. It can also provide a means to improve GIS systems, especially for pedestrians, using visual landmark recognition for navigation. This function is also important in Google Streetview for providing street name and house number recognition, and context retrieval on image/video-based applications. In pedestrian navigation support systems, finding a particular path towards the actual final goal within a room radius in a large building such as an office or hospital is becoming increasingly important but is still unsolved. To recognize items in a complex scene image captured by digital cameras or smart devices entails several problems in image processing, due to the variety of color patterns, blur, noise or other image distortions.

Specifically, finding text embedded in photographs of urban/natural scenes is still a difficult pattern recognition problem. This is because there are often complex background images with similar colors and gradients to the target text elements. Contrary to paper-based optical character recognition (OCR) problems where the background consists of an evenly distributed color and non-salient texture, the background image content in scene images is highly complicated. Additionally, there are various text styles (fonts), multiple colors and scale, and perspective variations both in Western and Asian scripts. There is also much more text than just street numbers. Advertisements may be presented as three-dimensional physical text objects in a multitude of shape variants. Texts in a scene image can be viewed from different angles, deforming the observed

*Corresponding author:
e-mail: B.Sriman@rug.nl (Bowornrat Sriman), l.r.b.schomaker@rug.nl (Lambert Schomaker)

ACCEPTED MANUSCRIPT

objects and in addition, shadow and poor lighting conditions have a big influence, if the goal is to capture the text information in a real-world scene. However, despite these difficulties, recognizing the text strings in a real scene provides highly useful, uncomplicated and explicit information on the ambient environment to humans and AI systems. Therefore, classifying text versus non-text in a natural scene is an important goal.

There are various classification techniques to discriminate texts from complicated backgrounds such as artificial neural networks (ANN), nearest-neighbor (1NN), support vector machine (SVM) and the deep convolutional neural network (DCNN). Recently, CNN or deep learning has evolved into a powerful technique used in image analysis, e.g., scene text detection, classifying objects and object recognition. However, there is a complex modeling process; for instance, multi-digit number recognition from Street View requires an astonishing number of 11 network layers [13]. Deep learning therefore requires not only a huge amount of labeled sample data but also huge computing resources. Therefore, in order to be less time-consuming, the size of the raw input image from the camera is usually reduced [14]. It should be noted that the current successes of CNNs start with the availability of a correctly cropped sub-image of usually approximately 256×256 pixels. This represents the width of 16-32 OCR-legible characters only. In our application, however, the input image will be a complete scene of, e.g., 2-41 megapixels. A full convolution of a 256^2 pixel patch over such a large image is computationally too demanding. Fast and efficient prior detection of relevant image regions will be required in convenient real-life applications. In simple terms, one needs to be able to detect text (FG) and non-text (BG) blocks in a computationally effective and convenient manner, before the stage of text recognition (OCR) itself.

Local, structural shape features such as SIFT and SURF are an important solution for the detection of relevant text regions. Such local features can be computed easily and have been shown to be well suited for matching and recognition as well as for many other applications where occlusion, background clutter, and other content variations occur. To compute points of interest (POI) in the image, it has been found that the Harris-Laplace (HL) detection method provides a larger portion of salient image patches than SIFT [29]. HL-detected POIs identify rich local information allowing higher-level object attributes, e.g., descriptions and color, to be completed for that region. However, whereas HL appears to be more useful for POI (keypoint) detection, the SIFT descriptor may be very useful for additionally describing a region, due to the scale and orientation invariance, if it can effectively discriminate between foreground and background regions.

However, images of a natural scene are typically colorful and have much more texture than scanned printed text documents, which usually contain a white background and dark foreground. This observation seems to imply that some color features, e.g., the histogram of real-valued color features in different color spaces should be used to identify FG/BG properly. In addition, the illumination variation, shadows, and reflections within a natural scene, will influence the performance of such a color-based image classification. For example, our human visual system will compensate for the observed differences in colors of the same object when the object is struck by light, [33] but it is still a challenge to realize color constancy in an algorithm.

When an image is captured, it is converted from analog signals to a digital image, in a quantization process in order to convert the intensity of light to a number. There have been several color intensity representations used in various application fields, since the development of color theory [17, 34]. Nevertheless, the most commonly used representation uses 256 intensity levels in the range 0-255 for pixel value for each of the colors red, green and blue. Due to the complicated relation between lighting conditions and reflective properties of text and background material, some transform in the raw RGB values is needed to realize color constancy. To illustrate this, a text sign was captured with a smartphone at different times of the day, i.e., in the morning and evening, as shown in Fig. 1. We experimented with some feature schemes, which did not produce a promising outcome [29]. The word image at the top was taken on a sunny morning whereas the photo at the bottom was taken in the evening under cloudy conditions. For each photo, the histogram of R|G|B color intensities $I\epsilon[0, 255]$ is computed. The histograms of the images taken during daytime and evening are noticeably different (Fig. 1). What is needed is a change that retains the useful color information better than RGB histograms, disregarding the irrelevant lighting condition.

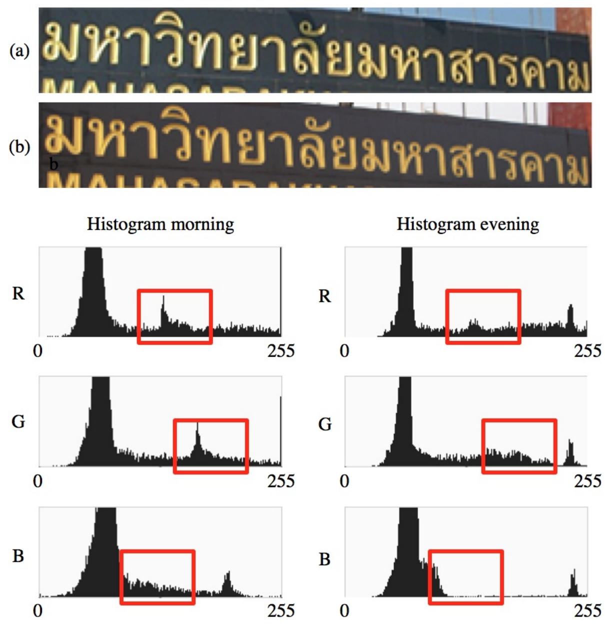


Fig. 1: Illustrates one text under sunny conditions vs evening. Color histogram in R, G and B for a text object photographed in the morning (a) and in the evening (b). The differences are highlighted in corresponding rectangles on the left and the right.

Autocorrelation is a form of time series analysis. It describes the correlation between data sequences in a single series of sample values, at several delays in time. Using autocorrelation, the

ACCEPTED MANUSCRIPT

position of salient elements (e.g., peaks) in time is made irrelevant, whereas the information about their shape is retained. Therefore, it was hypothesized that it would be possible to represent color information in an intensity-invariant manner, by using the autocorrelation functions (ACFs) for the histograms of color intensity in red, green and blue, or even in other color-coding schemes. Although ACFs have been used in image processing [4, 31], we believe this is a novel approach. Summarizing, we expect the heuristic of autocorrelation in time series analysis to reduce the illumination problem in FG/BG detection in natural scenes. After applying ACF to the histograms, the graphs of the images taken at different times appear much more similar (Fig. 2). On the basis of such observations we expect that the autocorrelation function can be used to attain color constancy, at least to an important extent.

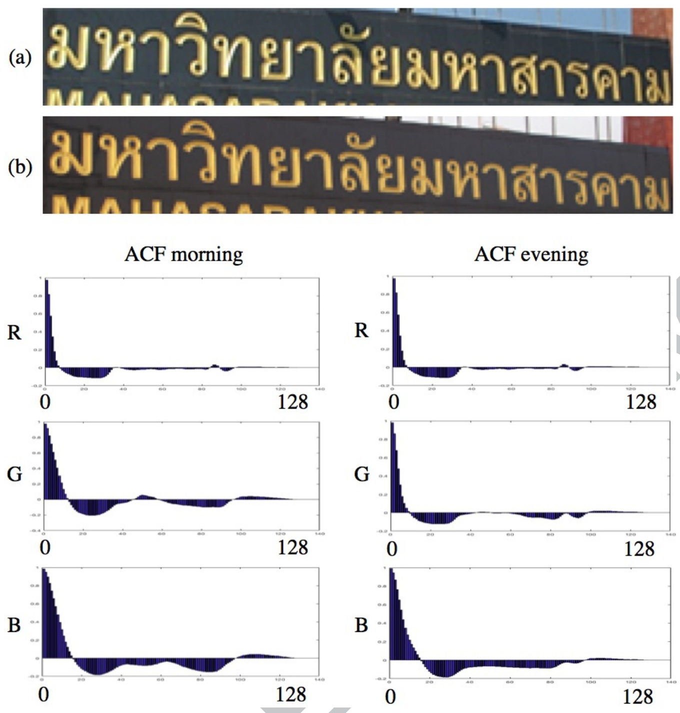


Fig. 2: Illustrates a text under sunny conditions vs evening.

In order to test whether this expectation holds, we present text/non-text region classification, using the image features (i.e., color autocorrelation histogram (CAH)) and compare it to standard SIFT. The basic idea is to obtain the image features around a salient point of interest (POI) by using the Harris-Laplace (HL) detector in order to find the prominent characteristics (e.g., color and descriptor) of the images around a POI, so a region of interest (ROI) can be defined. All the detected regions will be represented by features extracted from this region. So, at each POI, a feature vector is extracted using the global feature (CAH) and local feature (SIFT) in order to represent the region information. Each region can be divided into the foreground (text, FG) class or the background (non-text, BG) class. However, in fact, this dual-class definition is a simplification. Unlike the situation of ordinary document analysis, the BG class is a container, in which both simple signs or plate colors are represented as well as complicated natural or urban scenes. Therefore, as we have suggested elsewhere [29], explicit modeling is required: intensity differences are not a sufficient heuristic for FG/BG separation. These feature vectors should be gathered into categories. The bag-of-word model was first reported to use the frequencies of words from a dictionary for document representation [27]. Today, a similar bag-of-visual-words (BOVW) method is often used for scene image processing. It is convenient and effective in many computer vision applications such as action recognition [16, 28], image classification and character recognition. To obtain the codebooks or BOVW, all feature vectors are grouped into clusters using the k-means algorithm. Due to the visual complexity of scenes, *k* may be a large number. There are several criteria to be involved, so the best selection could predict the best result.

The contributions are investigated by the optimum performance of the experiments, which are designed as follows:

(1) In order to study the impact of different conditions for CAH based on RGB color, we explore the optimal parameter values for: i) patch sizes, ii) autocorrelation types, iii) distance types, iv) normalization types, and v) codebook sizes which require a computationally intensive grid search. The best result will then be chosen to create the CAH classifier and SIFT classifier.

(2) There are many kinds of color spaces; therefore, nine color spaces using the selection criteria for creating CAH feature are compared to get the best color classification result.

(3) In the stage of comparing text/non-text classification, we want to test the basic features first, such that e.g., CNN-based methods can be contrasted with this in later research. So, CAH and SIFT features are compared for accuracy by two different model classifier methods such as an ordinary nearest neighbor (1NN) and a proven method such as SVM.

(4) Finally, we attempt to exploit the relative benefits of the CAH and SIFT classifiers by adjoining the feature vectors and evaluating the effect on text/non-text classification.

The remainder of this paper is organized as follows. In Section 2, we review related work on the problem of character recognition, and text/non-text classification. In the next stage, we describe and explain the architecture of our proposed method regarding feature extraction following by model creation in Section 4. Then, we present extensive experimental evaluations for getting the optimum control parameters and discussion in Section 5 and two classifiers are described in Section 6. The experimental evaluations are reported in Section 7 and finally we conclude our work in Section 8.

2. Related work

Not all regions in a scene image are equally informative. To identify texts in images, the concept of 'points of interest' plays an important role. The idea is to detect some critical points in the image. These may include corner points, edges and so on. To describe the characteristics of these points, several approaches have been proposed for extracting their features using local descriptors, e.g., SIFT and Harris-Laplace (HL). Azad and et. al. [3] presented the features that combine the Harris corner detector with the SIFT descriptor for real-time localization

ACCEPTED MANUSCRIPT

and recognition of textured objects. The proposed approach can construct the features within approximately 20ms computing time by recognizing an image of size 640×480 pixels and localizing a single object at 30Hz of frame rates. Wang et al. [35] and Weixing et al. [37] used HL to detect and locate image features for automatic image registration based on the effectiveness of scale-invariance of HL. However, the multi-scale approaches have a common defect. Therefore, based on the scale-space of HL together with the reliability of SIFT descriptors, Zhang et al. [43] improved the algorithm to remove the redundant points detected by the original Harris-Laplace. Gong et al. [12] implemented the application to target tracking using the Harris-Laplace corner to localize the target, and the results promise the feasibility of the proposed method and precisely localize the target; so far it has been used to make an intelligent transportation system.

In addition, a color-based method is a crucial technique for text localization, face recognition, as well as object segmentation. Kwok et al. [18] investigated the separation of foreground and background objects based on the selected distribution with the maximum entropy that affects the aerial images of planted fields. The images are transformed from each RGB color to YIQ, YUV, $I_1I_2I_3$, HSI, and HSV color spaces. The result showed that the method was appropriate for segmentation of color images. Nevertheless, it had a bias effect on grayscale images. Color space is also beneficial for segmenting the image, e.g., using chromatic and achromatic information in XYZ color signals and separately smoothing them through anisotropic diffusion [23]. The results of these experiments have confirmed the effectiveness of this approach.

One crucial step after detecting and localizing potential text fragments in images is to determine the possibility of these fragments (so-called text candidates) to be part of actual texts or non-text objects. Block-based classification is one of the frequently-used techniques that have been reported in the literature. Blocks (rectangular image regions) are generated around the point-of-interest candidates before a classification is performed to classify the blocks into text and non-text blocks. Many previous studies have presented a classification technique to verify text blocks in scene images. Jiang et al. [39] proposed verifying a list of connected components (CCs) (derived from a color clustering algorithm) to be texts and non-texts by a 2-stage classification: coarse classification and then precise classification. The coarse classification included a series of cascaded classifiers, considered five features (geometric, shape regularity, edge, stroke and spatial coherence features) and two thresholds to discard non-text CCs. All accepted CCs from the previous step, and then were judged by the precise classification using SVM. The experiment showed that classifying text provided a more explicit result than non-text. The alternative technique for classifying candidate text regions is two-layer classification presented by Zhu et al. [44]. The first layer is to compare the similarity score of CC region blocks, which can filter out most repeat backgrounds. The second layer is an SVM classifier using a histogram of gradients (HOG) descriptor. The experiments showed that the first layer classified non-text as 63% and text as 97.3%. After the second layer was applied for verifying the text classification, the result increased by approximately 0.4%.

Minetto et al. [25] proposed a technique to classify a box of the text-line candidates of a GPS-tagged high-resolution digital photo of a city, which applied the T-HOG descriptor to generate a descriptive feature. The classifier classified the candidates into text and non-text regions. The classifier was constructed with a multi-cell histogram of oriented gradients (HOG) of Dalal and Triggs [7]. It was used to analyze the differences in font sizes and ignore irrelevant texture inside characters.

The approach of characterness cues presented by Li [19] is used to measure the unique properties of characters. The feature includes three property cues: stroke width (SW), perceptual divergence (PD), and a histogram of gradients at edges (eHOG), which are represented by region r. The region r is then used to compute the probability of characters and backgrounds by the Naive Bayes model to observe the distribution likelihood of characters and non-characters for each cue. The observation showed that in the PD distribution, characters tended to have a higher contrast than non-characters, which is different to both SW and eHOG. The evaluations of the proposed methods show that SW is applied to indicate characters and non-characters. However, it is more effective when all cues are combined.

Graphics and scene text discrimination in video document analysis are also challenging and interesting problems because of the clarity and separability of graphics and scene text in video frames. Therefore, Xu et al. [40] presented a method for classifying graphics texts and scene texts under the hypothesis that graphics texts are arranged at almost the same location and have a uniform color with a plain background, against scene texts that have a non-uniform color and cluttered background. They use the principle of deviation of the movement of the text block for a few seconds to determine whether it is a scene or graphic text. If there is a high deviation, it is a scene text. On the other hand, graphic text usually has a low deviation in position. The result of the proposed method guarantees that the classifying graphics and scene texts are correctly classified. However, there may be a problem with static scene texts that are classified as graphic texts.

Recently, Zhu and Zanibbi [42] proposed a method for scene-text detection with a feature learning-based convolutional neural network called Text-Conv and cascaded classification. In the feature learning stage, they utilized and minimized some of the equations in Coates's et al.s [5] algorithm. The algorithm provided the convolution masks, a number of $k = 1,000$, which were used for both the coarse and fine detectors stages. This stage produced many patches, and the patches were classified as text/non-text using the confidence-rated AdaBoost. The experiments regarding the Area Under Curve of Precision/Recall on ICDAR2013 [5] showed that the detection hotmap obtains 71.2%, while Coates et al. obtained 62%.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

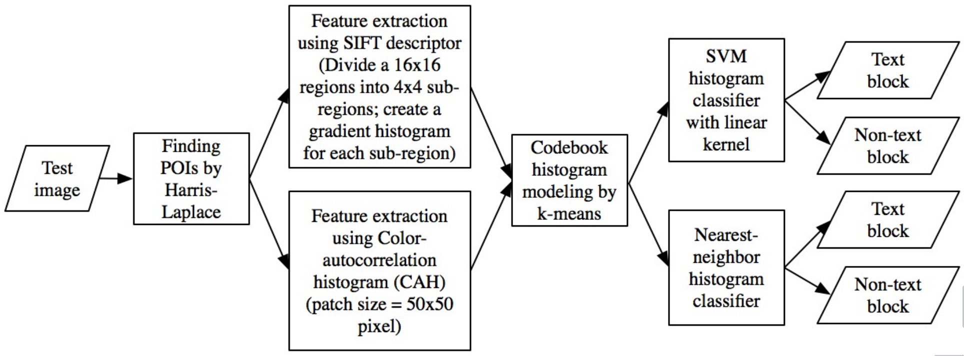

Test image
Finding POIs by Harris-Laplace
Feature extraction using SIFT descriptor (Divide a 16x16 regions into 4x4 sub-regions; create a gradient histogram for each sub-region)
Feature extraction using Color-autocorrelation histogram (CAH) (patch size = 50x50 pixel)
Codebook histogram modeling by k-means
SVM histogram classifier with linear kernel
Nearest-neighbor histogram classifier
Text block
Non-text block
Text block
Non-text block

Fig. 3: Overview of the proposed classification method.

3. Proposed approach

For text/non-text classification, the proposed method first defines candidate text and non-text blocks from the ground truth, which will be used for generating models and test images. First, we need to locate the points of interest (POIs) in the images by, e.g., SIFT or Harris-Laplace (HL) (Section 3.2). The information of a POI (keypoint) location (x,y) is crucial to describe the image context around that detected point. The patch size of the regions of interest (ROIs) around a POI will be determined empirically (Section 5.1). In order to compare the capabilities of different feature-extraction techniques, we extract, at each POI, the SIFT descriptors (N_{dim} = 128) and the color autocorrelation histogram (CAH). The area used to compute the CAH features is derived from an expanded region around the key-point position obtained from the HL detector. The image patch will be characterized by a color histogram of color space intensities, e.g., $R|G|B$ or $Y|U|V$. To suppress average lighting-condition variations, the autocorrelation function of such histograms is computed (Eq. 1).

$$r(\tau) = \int_{-\infty}^{+\infty} x(t)x(t-\tau)dt \qquad (1)$$

lag	1	2	3	4	5	6	7	8
Xt	0	0	0	1	2	3	4	5
Xt-1	0	0	1	2	3	4	5	4

lag	9	10	11	12	13	14	15
Xt	4	3	2	1	0	0	0
Xt-1	3	2	1	0	0	0	0

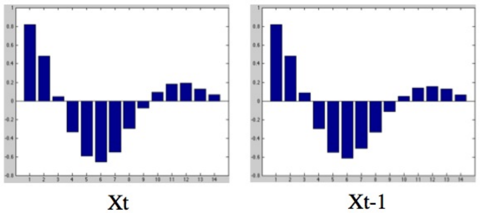


Fig. 4: The Autocorrelation histogram of X_t and X_{t-1}.

If the data are completely random, the autocorrelation value should be close to zero for all time lags; for instance, Fig.4 shows the autocorrelation plotting at time t, which should not be significantly different from the point at time $t-1$ and so on, with a peak at $\tau = 0$. Color histograms may contain multiple intensity peaks, the position of which (i.e., the lighting) may not be important, while the presence of repetitions is significant.

In the proposed method, therefore, the three channels of the color space intensity histograms are shifted by autocorrelation. To reduce the dimensionality of the vector and simplify the calculation, each color channel is quantized to 128 levels and then the acf vectors of the three channels are adjoined into a vector of M_{dim} = 384. Additionally, there are various raw features derived from both the SIFT and CAH features. The BOVW technique with k-means clustering is able to reduce a huge feature-vector set into a number of k representative groups, i.e., clusters. The cluster centroids represent prototypical keypoints (PKP) for both FG and BG classes separately and are used as a code book. This allows a count of the PKPs present in an image region, for both classes. Subsequently, the resulting histogram can be classified. We will test two classifiers: SVM and nearest neighbor (1NN). The diagram of the proposed classification technique is depicted in Fig. 3.

3.1. Candidate text/non-text region

The method proposed in this paper first generates candidate regions (blocks), possibly containing text (or requiring non-text) to be processed by later image analysis tools such as character extraction and recognition. To acquire text/non-text blocks, the text region can be easily derived from the ground truth, while the non-text regions need to be established automatically, with the constraint of getting sizes similar to the text regions as explained in Algorithm 1.

3.2. Localization of Point of Interest

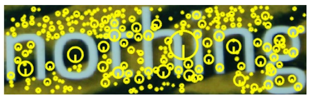
A scene image usually includes illumination, texture, color and a cluttered background. Extracting a global feature from the complicated topology of the image may not provide optimal and robust results, in contrast to using local features [10]. The objective of using local features is to create precise descriptors of individual local image structures, concerning a point, edge, or corner. A local descriptor is used to indicate a small patch of pattern in an image that varies from its neighboring regions. All multiple patches are used to match an image. The matching result represents the properties of the region such as illumination, texture, and color. Good local descriptors can cover information

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

of the image even if the image scale is changed; they are also invariant to image transformations, and more robust to partial occlusion than global descriptors.

Although localization of structural features by SIFT is robust against image transformations and small geometric distortions, it sometimes provides an insufficient number of keypoints in order to obtain higher-level object attributes [29] as shown in Fig. 5. Figure 5 demonstrates the number of POIs (the yellow point) detected by SIFT and HL in the same image (Fig. 5(a) shows that SIFT using the DoG algorithm provides 217 keypoints, while HL gives 421 keypoints in Fig. 5(b)). The Harris-Laplace (HL) detector appears to be more effective in POI (keypoint) detection than SIFT. Therefore, this paper utilizes the HL detector to detect the points of interest (POIs) in FG/BG images.


(a) DoG of SIFT (267 points)

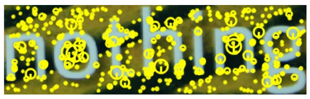

(b) Harris-Laplace (HL) (450 points)

Fig. 5: Comparison of keypoint detection described for two local feature descriptors: SIFT (ratio = 1) and Harris-Laplace.

The Harris-Laplace detector, proposed by Mikolajczyk and Schmid [24], has been shown to have a better performance in repeatability, localization and scale variation than other detectors. Its advantage is in providing a higher accuracy in the location and scale of the interest points with a reduced computational complexity. The Harris-Laplace detector process starts with localizing points in scale-space by the multi-scale Harris function on the image derivative P(**x**) where **x**=(x,y) using the autocorrelation matrix $\mu(\mathbf{x}, \sigma_I, \sigma_D)$. The matrix describes the gradient distribution in the local maxima of the 8-neighborhood of point **x**, which is defined by Eq. (2).

$$\begin{aligned}\mu(\mathbf{x}, \sigma_I, \sigma_D) &= \begin{bmatrix} \mu_{11} & \mu_{12} \\ \mu_{21} & \mu_{22} \end{bmatrix} \\ &= \sigma_D^2 g(\sigma_I) * \begin{bmatrix} P_x^2(\mathbf{x}, \sigma_D) & P_xP_y(\mathbf{x}, \sigma_D) \\ P_xP_y(\mathbf{x}, \sigma_D) & P_y^2(\mathbf{x}, \sigma_D) \end{bmatrix}\end{aligned} \quad (2)$$

Given the integration scale represented by σ_I where $I = 1..n$ and $\sigma_n = \xi^n \sigma_0$, where the scale factor between successive levels $\xi = 1.4$, the local scale is $\sigma_D = s\sigma_n$ where the constant vector $s = 0.7$, α represents practical value, and the matrix $P_x(\mathbf{x}, \sigma_D)$, $P_y(\mathbf{x}, \sigma_D)$ is the derivatives computed using the Gaussian window size σ_I ($g(\sigma_I)$) in the x and y-direction at point **x**. Therefore, when the Harris measure combines the trace and the determinant of the second moment matrix, it will be changed to the function in Eq. (3).

$$P(\mathbf{x}) = det(\mu(\mathbf{x}, \sigma_I, \sigma_D)) - \alpha trace^2(\mu(\mathbf{x}, \sigma_I, \sigma_D)) \quad (3)$$

It then selects appropriate scales, which were extensively studied by Lindeberg [20]. The scale selection concept is to select a characteristic scale by searching a local extremum over scales in order to reduce the set of interest points using Laplacian-of-Gaussian (LoG) in Eq. (4). Determined, P_{xx}, P_{yy} are the second image derivatives in horizontal and vertical directions, respectively. The characteristic scale at position **x** is defined by the scale σ_n which hits the maximum of $|LoG(\mathbf{x}, \sigma_n)|$.

$$|LoG(\mathbf{x}, \sigma_n)| = \sigma_n^2 \left|P_{xx}(\mathbf{x}, \sigma_n) + P_{yy}(\mathbf{x}, \sigma_n)\right| \quad (4)$$

Salient positions in the scale space are detected by computing a multi-scale stack of the Harris corner indicator $\{H(\mathbf{x}, \sigma_k, \gamma\sigma_k)\}_{k=1}^n$, where γ is a scalar. For each scale σ_k, the local maxima of the Harris indicator $H(\mathbf{x}, \sigma_k, \gamma\sigma_k)$ are computed by $\{(\hat{\mathbf{x}}, \sigma)\} = argmaxlocal_{\mathbf{x}} H(\mathbf{x}, \sigma_k, \gamma\sigma)$. Then, select points at which the normalized LoG is maximal across scales and the maximum is above a threshold ($\hat{\sigma} = argmax_{\sigma} |LoG(\hat{\mathbf{x}}, \gamma\sigma)|$). The spatial location $\hat{\mathbf{x}}$ which do not hit a scale maximum, is given up.

3.3. Feature Extraction

The POIs or keypoints obtained from the previous process will be used to compute dedicated features. Around each keypoint location (Fig. 6(a) and Fig. 6(c)), two types of feature are computed, i.e., the SIFT descriptor and a color autocorrelation histogram (CAH). Suppose there is a candidate FG image with HL-detected keypoints as POIs in Fig. 6. At each extracted candidate keypoint (Fig. 6(a)), a SIFT descriptor will be computed (Fig. 6(b)). At the same time, the expanded region around the POI candidate will be cut out of the image, i.e., be 'cropped' with a width equal to the length in Fig. 6(c). The small, cropped region of M×M pixels will be called ROI (region of interest) in the sequel. Then the color autocorrelation histogram feature (CAH) will be computed from this ROI (Fig. 6(d)). As a result, the feature description addresses both structural shape aspects (SIFT) and color information (CAH).

3.3.1. Scale Invariant Feature Transform (SIFT)

SIFT is an efficient local feature descriptor proposed by Lowe [22], which has been shown to be useful in extracting dominant image features regardless of translation, scaling and projective transformations on the salient visual element. SIFT consists of four major stages: 1) scale-space extrema detection, 2) keypoint localization, 3) orientation assignment, and 4) computing the keypoint descriptors. The scale space of an image is determined by $L(x, y, \sigma)$ obtained from the convolution of an image $I(x, y)$ with the Gaussian filter $G(x, y, \sigma)$ with a small σ value (Eq. (5)).

$$\begin{aligned} L(x, y, \sigma) &= G(x, y, \sigma) * I(x, y) \\ G(x, y, \sigma) &= \frac{1}{2\pi\sigma^2} e^{-((x^2+y^2)/2\sigma^2)} \end{aligned} \quad (5)$$

Extracting the SIFT feature requires the difference between two Gaussian-convolved images to be computed, at different

ACCEPTED MANUSCRIPT

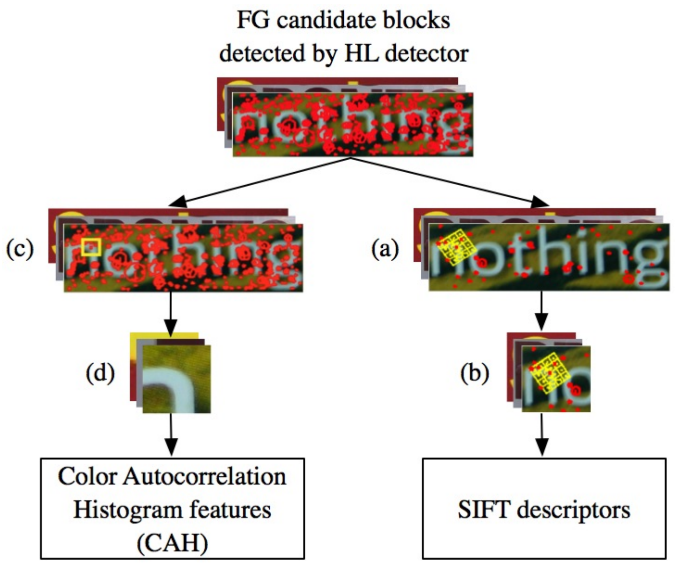

FG candidate blocks detected by HL detector
(c)
(a)
(d)
(b)
Color Autocorrelation Histogram features (CAH)
SIFT descriptors

Fig. 6: Extracting features at each keypoint using SIFT and CAH.

scales, by using difference of Gaussians (DoG), which is separated by a constant multiplicative factor $k = \sqrt{2}$ in Eq. (6). In order to obtain accurate keypoint localization, the extrema (min and max) need to be computed by comparing a pixel to its neighbors in the filtered image.

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{6}$$

Therefore, we extract local features of various resolutions and scales at the POIs, which are detected by the Harris-Laplace keypoint detector (represented by the red circles in Fig. 7(a)). Fig. 7(b) shows an example of POI features extracted by SIFT features, which consist of a coordinate (x, y), scale, and orientation in adjacent 4×4-pixel sub regions of a 16×16-pixel square. The descriptors are computed using the gradient magnitude and orientation around the keypoints, i.e. eight values per sub region. Then the 16×8 = 128-dimensional feature descriptor of this keypoint is complete, as demonstrated in Fig. 7(c).

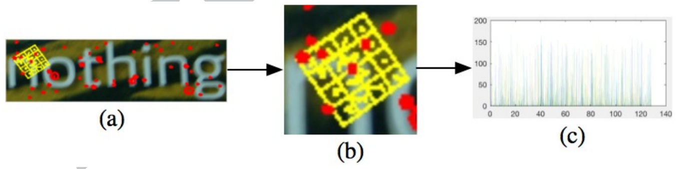


Fig. 7: An example of generating features at each keypoint using SIFT descriptors.

3.3.2. Color Intensity Histogram by Time Series Analysis

The text and non-text blocks appearing in nature scenes are usually decomposed in different color channels. Color is a determinant of the image that plays a vital role in image-retrieval systems. Color or hue is often applied in scene-image processing because color can be highly informative in object classification of both natural objects (e.g., fruits, flowers) and manmade artefacts such as colorful advertisement texts and street signs. Furthermore, color also facilitates the distinction in homogeneous image regions, such as blue sky, the yellow of banana, or the green of leaves. In the image-processing process, a system needs to be able to associate the measured color intensities with physical properties. This is not easy, due to the interaction between the color properties of the incident light and the light that is reflected by an object (metamerism). There are many color spaces that have been used in color analysis [32]. In order to choose the most accurate color space for the purpose of image classification in this paper, a set of the well-known color spaces RGB, $C_1C_2C_3$, $L_1L_2L_3$, $O_1O_2O_3$, HSI, HSV, $I_1I_2I_3$, YUV and YIQ is analyzed.

1. RGB color images are generally contained in the format of the RGB color space Eq. (7), which is blended from the three primary colors red, green and blue. Different proportions of the three colors can make more colors. These colors are used for light display systems such as television, computers, cameras, and projectors.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} R/(R+G+B) \\ G/(R+G+B) \\ B/(R+G+B) \end{bmatrix} \tag{7}$$

2. $C_1C_2C_3$ is a normalized RGB, which has invariant characteristics. The $C_1C_2C_3$ color space is appropriated for photometric color invariants for matte, dull surfaces [11]. It is independent of the changes in orientation, illumination direction, and illumination intensity; the color is determined by Eq. (8).

$$\begin{bmatrix} C1 \\ C2 \\ C3 \end{bmatrix} = \begin{bmatrix} tan^{-1} \times (R/max\{G, B\}) \\ tan^{-1} \times (G/max\{R, B\}) \\ tan^{-1} \times (B/max\{R, G\}) \end{bmatrix} \tag{8}$$

3. $L_1L_2L_3$, proposed by Gevers and Smeulders [11], was presented to determine the direction of a triangular plane in the RGB space, or it represents normalized square color differences as described in Eq. (9).

$$\begin{bmatrix} L1 \\ L2 \\ L3 \end{bmatrix} = \begin{bmatrix} (R-G)^2/((R-G)^2 + (R-B)^2 + (G-B)^2) \\ (R-B)^2/((R-G)^2 + (R-B)^2 + (G-B)^2) \\ (G-B)^2/((R-G)^2 + (R-B)^2 + (G-B)^2) \end{bmatrix} \tag{9}$$

4. $O_1O_2O_3$, the opponent color space, is color separated into three channels that are channel $O_1O_2O_3$. Channel O_1 and O_2 represent the chromatic information, while O_3 represents the intensity information in Eq. (10).

$$\begin{bmatrix} O1 \\ O2 \\ O3 \end{bmatrix} = \begin{bmatrix} (R-G)/\sqrt{2} \\ (R+G-(2 \times B))/\sqrt{6} \\ (R+G+B)/\sqrt{3} \end{bmatrix} \tag{10}$$

5. HSI, the hue-saturation-intensity color model, is human intuition and commonly used in image processing. The HSI in Eq. (11) represents three elements: hue (H) describes its color in the range [0°,360°], saturation (S) reflects the color purity,

ACCEPTED MANUSCRIPT

and intensity (I) is black and white. HSI is generated via the nonlinear transformation of the RGB color space, in which the intensity and saturation of manipulated pixels is limited in the range of [0,1].

$$\begin{bmatrix} H \\ S \\ I \end{bmatrix} = \begin{bmatrix} arctan(\sqrt{3}(G-B)/(2R-G-B)) \\ 1 - min(R,G,B)/I \\ (R+G+B)/3 \end{bmatrix} \quad (11)$$

6. HSV is a color model comprising of hue, saturation, and value, that describes color in terms of its shade and brightness. Hue is the value of the primary colors (red, green and blue). Value is the brightness of the colors, which can be measured by the intensity of the brightness of each color. Therefore, HSV in Eq. (12) can expediently make the brightness channel more vivid than other color models.

$$\begin{bmatrix} H \\ S \\ V \end{bmatrix} = \begin{bmatrix} arctan(\sqrt{3}(G-B)/(2R-G-B)) \\ 1 - 3 \times min(R,G,B)/(R+G+B) \\ max(R,G,B) \end{bmatrix} \quad (12)$$

7. $I_1I_2I_3$ color space is obtained through the deceleration of the RGB color components using the dynamic K.L. transformation by Ohta et al. [41]. The three orthogonal color spaces are according to Eq. (13).

$$\begin{bmatrix} I1 \\ I2 \\ I3 \end{bmatrix} = \begin{bmatrix} 0.33 & 0.33 & 0.33 \\ 0.5 & 0.0 & -0.5 \\ -0.25 & 0.5 & -0.25 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (13)$$

8. The YUV color space in Eq. (14) is used to color video standards. It most added on phase alternating line (PAL) and sequential color with memory (SECAM) television. It consists of the luminance (Y) component, which determines the brightness of the color, while the two components (U and V) represent the color itself (the chrominance).

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.3 & 0.59 & 0.11 \\ -0.15 & -0.29 & 0.44 \\ 0.6 & -0.51 & -0.1 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (14)$$

9. YIQ is implemented for TV broadcasting, which the primary objective is working with black and white television. The composition of YIQ is represented by Y, I and Q, which is an encoded color signal of the image. The Y represents the luminance information, which is the signal used for black and white TVs. The chrominance information is contained in the components I and Q. The procedure of YIQ is demonstrated in Eq. (15).

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.3 & 0.59 & 0.11 \\ 0.6 & -0.28 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (15)$$

In order to obtain a usable degree of independence from lighting conditions, we propose to use an autocorrelation on the histograms of the color channel intensities. The autocorrelation-based CAH feature should represent color distribution in a manner which is not directly determined by the general intensity level, while retaining relative color information. There are different practical methods for computing an autocorrelation function. Some of them operate via the time domain, whereas others are based on the inverse Fourier transform.

Additional implementation details involve normalization and handling of circularity and/or boundary effects. We used three methods: the autocorrelation function (ACF) in Fig. 8, the inverse fast Fourier transform (IFFT) in Fig. 9, and the cross-correlation (XCORR) in Fig. 10 (matlab, [26]), in order to find the optimal method for discriminating between foreground and background regions. Each patch region is separated into three channels of a given color space intensity histogram, for example, $R|G|B$ color space, or $H|S|V$, etc. The resulting histogram is then converted by one of these three kinds of autocorrelation. In order to reduce computational complexity, the histogram has reduced dimensions (128 levels). Finally, the autocorrelation vectors of the three channels are adjoined into an $M_{dim} = 384$ vector to represent the complete CAH feature.

4. Codebook Histogram Modeling

The bag-of-features model has been widely used in computer vision and video analysis, e.g., for separating objects from cluttered backgrounds, for face and character recognition, and for image classification. In this paper, the bag-of-visual words (BOVW) is deployed to create a combined codebook histogram of color autocorrelation histogram (CAH) features and SIFT features. When all the training images have been processed, yielding the CAH features and SIFT descriptors, the codebook generation is performed through clustering as depicted in Fig. 11.

In order to obtain an optimum codebook for CAH, there are design decisions as regards the number of elements k of the codebook histogram and the use of distance function. The mobile device platform poses restrictions on memory use and computing efficiency. In addition, it is essential to determine the optimal local patch size to determine the level of detail to describe a cropped region. Various color spaces need to be evaluated for the CAH feature. Furthermore, a proper normalization method is necessary for calculating data in an adjoined vector with values in a comparable range. The relative performance of CAH and SIFT descriptors needs to be evaluated. Consequently, in order to choose the optimal parameter values, all the conditions in Section 5.1 need to be compared in terms of the performance of text versus non-text classification. We will use statistical testing (ANOVA) for the selection process.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

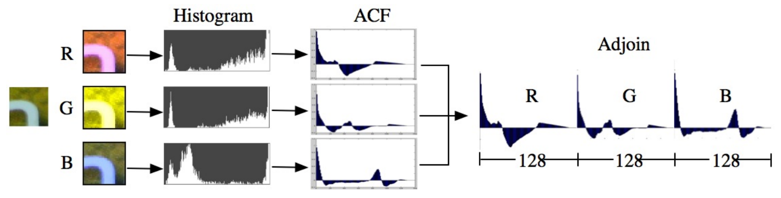


Fig. 8: The process of feature extraction on an image patch based on the color intensity histogram with the autocorrelation function (ACF).

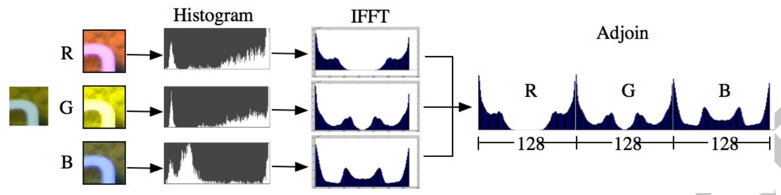


Fig. 9: The process of feature extraction on an image patch based on the color intensity histogram with the Inverse fast Fourier transform (IFFT).

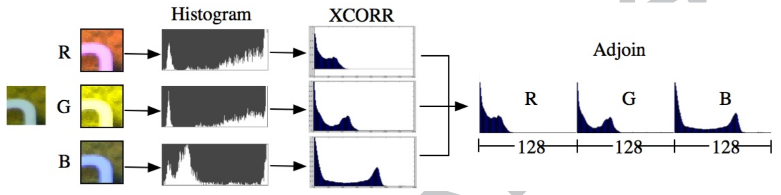


Fig. 10: The process of feature extraction on an image patch based on the color intensity histogram with the cross-correlation (XCORR).

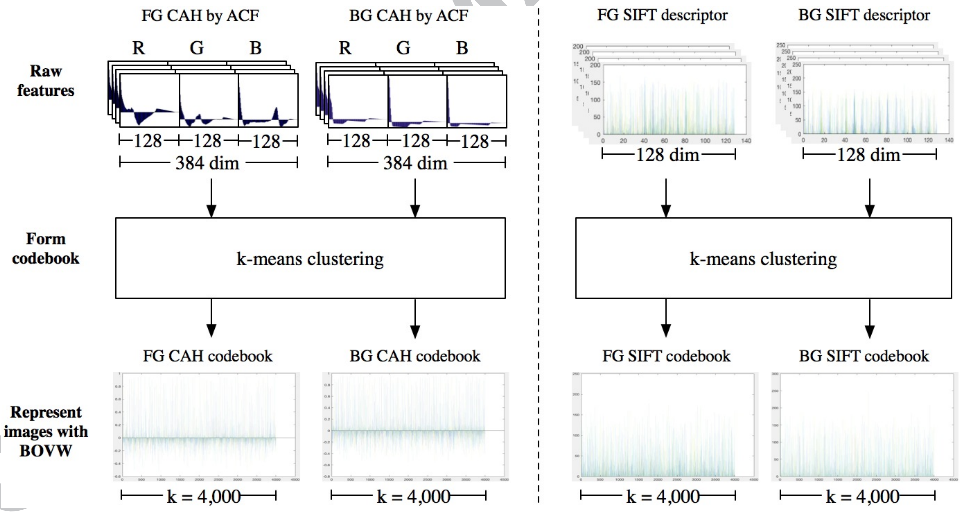


Fig. 11: The process of FG/BG codebook creation using color autocorrelation histogram (CAH) features (left-hand side). The FG and BG codebooks of SIFT features are realized by using k-means with k = 4,000 in order to be comparable with CAH (right-hand side).

Algorithm 1 cropBg

```
1: read total text ground truth of a training image.
2: while ~ $eof$ do
3:     compute size of each text ground truth.
4:     compute the largest rectangle area except text ground truth of a training image.
5:     randomly select rectangular areas from the largest rectangular area, to equal the size of the corresponding text ground truth.
```

4.1. K-means Clustering

The concept of clustering is to find similar characteristics between data without defined data classes or when the exact number of groups is unknown, in order to group similar objects into clusters. The process is called an “unsupervised learning algorithm”. K-means by Lloyds algorithm [21] is an unsupervised learning algorithm that is used for solving clustering problems in the cluster analysis. The concept of *k*-means is developing a lexicographic classification for a huge sample of data. Its purpose is to partition an N-dimensional population into the basis sample number of clusters (assume *k* clusters).

The process of *k*-means is as follows:

ACCEPTED MANUSCRIPT

1. Set s as a dimension of feature descriptors d_1, d_2, ..., d_s, where $d_{1..s}$ are members of a set of keypoints (kp).
2. Suppose that we have n keypoint (kp) vectors kp_1, kp_2, ..., kp_n all from the same class, and let k be the number of clusters to be grouped, where $k < n$.
3. Define the initial clusters by using random m keypoints as a partition.
4. Compute the cluster mean of each individual group.
5. Calculate the distance between cluster members and all cluster means (centroids) to find the nearest centroid of each member.
6. Iteratively relocate members to an appropriate cluster based on calculated distance until clusters are explicitly separated.

Therefore, we run k-means several times with different numbers of desired representative vectors (k) and different sets of initial cluster centers. We select the final clustering giving the lowest empirical risk in categorization.

5. Datasets and Training Process

We assess the performance of classification on three benchmark datasets including Western and Asian scripts. Data characteristics are urban/natural scenes with embedded text image information, in general with highly variable color, contrast, and gradient information. Overall images often have a complex background with similar colors and gradients as the text scale, font variations, and orientation. Also, shadow and lighting conditions are variable. Text styles (fonts) are multi-color and highly variable. The image regions containing text are provided as ground truth labels. The three proposed word scene classification methods are described as follows.

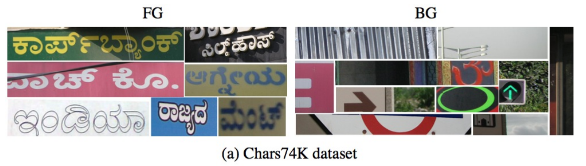
The first dataset is the Chars74K image dataset (in Fig. 12(a)) [9], which uses Kannada script that is similar to Thai script. The Kannada contains 390 text images with a minimum size of 37×60 pixels and maximum of 300×27 pixels, and the average height of the characters is 93.77 pixels. The dataset will be used to tune the control parameters and feature method.

The second is the ICDAR2015 [15], a well-known dataset, containing 1,943 Western word images (in Fig. 12(b))). It comprises an image size of 15×12 pixels for minimum and maximum of 1,735×833 pixels, and the average height of the characters is 87.85 pixels.

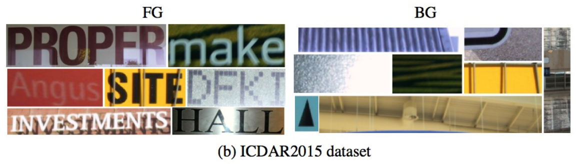
The third dataset is the Thai scene image dataset (TSIB) [30] captured by a smartphone (in Fig. 12(c)). All the words appearing in images are manually annotated. This dataset contains 10,400 word images. The minimum and maximum sizes are 8×11 pixels and 1,485×415 pixels, respectively. The average height of the characters is 84.45 pixels. A random selection of 5,200 text images will be made to conduct the experiments.

5.1. Training Process

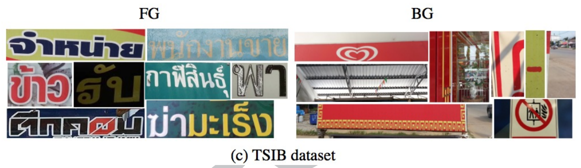
When computing color feature images in different color spaces, there are various qualities and different discriminating capabilities of the images since the color is mixed up with different elements. Each color element or space, e.g., the RGB color model, is an additive model, which comprises



(a) Chars74K dataset



(b) ICDAR2015 dataset



(c) TSIB dataset

Fig. 12: Example text (FG) and non-text (BG) images from (a) the Chars74K, (b) ICDAR2015, and (c) TSIB data sets.

the elements red, green, and blue (that are used to select the optimum criteria). At each step of the patched regions, the novel color feature using autocorrelation on the single-channel is calculated for the pixels in that region and used to compute the intensity histograms. During the stage of exploring the control parameters and feature methods in order to study the impact of different conditions for CAH based on RGB color, we explore the optimal parameter values for: i) patch sizes, ii) autocorrelation types, iii) distance types, iv) normalization types, and v) codebook sizes which requires a computationally intensive grid search. All the experiments were tested on the Peregine cluster, which has 24 cores @ 2.5 GHz (two Intel Xeon E5 2680v3 CPUs). The details of the conditions are defined as follows.

1. patch sizes: 20×20, 30×30, 40×40, 50×50, and 60×60 pixels,
2. autocorrelations: XCORR, IFFT, and ACF,
3. distances: cosine, correlation, Euclidean, city block, Spearman, and Chebychev distances,
4. normalisations:
   i) (x-xmin)/(xmax-xmin),
   ii) (x-mean(x(:)))/std(x(:)),
   iii) (|x|-xmin)/(xmax-xmin),
5. k in k-means clustering:
   1,000, 2,000, 3,000, 4,000, 5,000, 6,000, and 7,000.

Experiment 1: This experiment was conducted to obtain the optimum control parameters. In the initial testing, we use the training set of Kannada script in the Chars74K dataset, which was divided into 10 folds. The experiment starts with all the training images being extracted to get CAH feature descriptors, which are collected into a file and labeled separately with FG

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

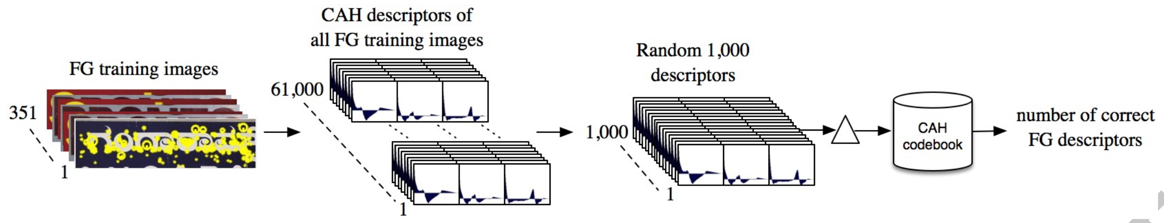

FG training images
351
1
CAH descriptors of all FG training images
61,000
1
Random 1,000 descriptors
1,000
1
CAH codebook
number of correct FG descriptors

Fig. 13: The process of computing to find the optimum criteria.

and BG. Then 1,000 FG and BG descriptors are selected randomly and matched to the FG and BG codebooks. The matching results show the probability of correct matching as illustrated in Fig. 13. A selection of the 105 matching results were analyzed by the ANOVA statistics. The analysis was computed 5 times (525 in total). We selected the best ANOVA result to get effective criteria that perform proper classification to be used in the next step.

Computing the difference of the training set; the result is illustrated in Table 1. It shows that the mean difference is not significant: 0.243 ($p \gg 0.05$), and therefore the data in the groups have no distribution difference.

Table 1: ANOVA statistic for the average results of the training set of 5-fold.

Fold	N	Mean	SD	Std. Err	95% CI Low	95% CI Up	Min	Max
1	105	97.80	2.43	0.24	97.33	98.28	85.40	99.95
2	105	97.90	2.21	0.22	97.47	98.33	85.30	99.80
3	105	98.07	2.20	0.21	97.65	98.50	86.50	99.95
4	105	97.96	2.35	0.23	97.51	98.42	86.90	99.90
5	105	**98.44**	1.51	0.15	98.15	98.74	89.10	99.95
Total	525	98.04	2.17	0.09	97.85	98.22	85.30	99.95

The area of the region of interests (ROIs) or patch size (derived from the extension of the POI by HL) was included for testing in the experiments. From Table 2, the mean difference is tested and it is found that the difference is significant at $p \ll 0.000001$, and the window size affects the accuracy, the wider window, and the higher accuracy. This is because the average height of the characters in the three datasets is 88.69. Hence, the window size must be large enough, e.g., the patch size should be higher than half the height of the character. Therefore, the patch size 50×50 is selected.

The area of the region of interests (ROIs) or patch size (derived from the extension of the POI by HL) was included for testing in the experiments. From Table 2, the mean difference is tested and found the difference is significant at $p \ll 0.000001$, the window size effects to the accuracy, the wider window, the higher accuracy. Since, the average height of the characters in the three datasets is 88.69. Hence, the window size is to be large enough, e.g., patch size should higher than the half height of the character. Therefore, patch size 50×50 is selected.

As mentioned earlier, selecting the type of autocorrelation is a crucial step in getting the promising result; therefore, a comparison of the results of the autocorrelation techniques: IFFT, XCORR, and ACF are demonstrated in Table 3. The mean difference is significant at $p \ll 0.000001$. Thus, the ACF algorithm affects the accuracy.

There are many distance metrics that can be used in creating a model. In order to delineate the relation between average accuracy and the distance metrics, ANOVA computes the mean difference and shows that it is significant at $p \ll 0.000001$, and that the city block distance algorithm relates to the accuracy (Table 4).

Table 2: Evaluation of accuracy over patch sizes.

Patch Sizes	N	Mean	SD	Std. Err	95% CI Low	95% CI Up	Min	Max
20×20	105	96.49	3.12	0.30	95.89	97.09	85.30	99.00
30×30	105	97.70	2.03	0.20	97.31	98.09	89.75	99.65
40×40	105	98.48	1.36	0.13	98.22	98.74	91.75	99.80
50×50	105	**98.61**	1.55	0.15	98.32	98.91	91.90	99.90
60×60	105	98.90	1.34	0.13	98.64	99.16	92.00	99.95
Total	525	98.04	2.17	0.09	97.85	98.22	85.30	99.95

Table 3: ANOVA statistic for training set of three types of CAH.

Autocorrelation	N	Mean	SD	Std. Err	95% CI Low	95% CI Up	Min	Max
IFFT	175	98.01	1.30	0.10	97.82	98.21	93.65	99.70
XCORR	175	97.53	3.00	0.23	97.08	97.97	85.30	99.90
ACF	175	**98.57**	1.71	0.13	98.31	98.83	89.10	99.95
Total	525	98.04	2.17	0.09	97.85	98.22	85.30	99.95

Table 4: Evaluation of accuracy over distance functions for 1NN.

Distances	N	Mean	SD	Std. Err	95% CI Low	95% CI Up	Min	Max
cosine	275	98.37	1.22	0.07	98.22	98.51	93.65	99.95
correlation	132	98.36	1.80	0.16	98.05	98.67	90.55	99.95
Euclidean	59	98.82	0.85	0.11	98.59	99.04	94.85	99.90
city block	16	**98.84**	1.39	0.35	98.10	99.58	94.50	99.95
Spearman	43	93.56	3.80	0.58	92.39	94.73	85.30	99.75
Total	525	98.04	2.17	0.09	97.85	98.22	85.30	99.95

The normalization method is used to handle values in the same range. The three normalization algorithms: *i* (*), *ii* (**), and *iii* (***) (in Section 5.1) are taken into account. From the empirical experiments, the mean difference is significant at $p \ll 0.000001$; thus, algorithm number *ii* matters for the accuracy (Table 5).

Table 5: Evaluation of accuracy over normalization methods.

Normalizations	N	Mean	SD	Std. Err	95% CI Low	95% CI Up	Min	Max
i (*)	489	98.24	1.73	0.08	98.09	98.40	85.40	99.95
ii (**)	16	**98.83**	0.79	0.20	98.41	99.25	97.05	99.80
iii (***)	20	92.36	4.08	0.91	90.45	94.27	85.30	99.80
Total	525	98.04	2.17	0.09	97.85	98.22	85.30	99.95

* i) (x-xmin)/(xmax-xmin)
** ii) (x-mean(x(:)))/std(x(:))
*** iii) (|x|-xmin)/(xmax-xmin)

To verify the codebook size of each FG and BG, we varied the size of codebook between 1,000 - 7,000 and increased the

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

step of the codebook by 1,000 as shown in Table 6. The mean difference is significant at $p \ll 0.000001$, which means that the size of codebook affects the accuracy. A higher size of codebook gives a higher accuracy. However, to make the processing time more applicable, but still productive, we chose 4,000 for generating the codebook size of both FG and BG.

Table 6: Evaluation of codebook size for SIFT and CAH descriptors.

Clusters	N	Mean	SD	Std.	95% CI		Min	Max
					Low	Up		
1,000	75	95.35	3.09	0.36	94.64	96.06	85.30	98.80
2,000	75	97.06	2.04	0.24	96.59	97.53	89.95	99.25
3,000	75	98.21	1.74	0.20	97.80	98.61	86.90	99.70
4,000	75	**98.53**	1.41	0.16	98.21	98.85	90.55	99.75
5,000	75	98.77	1.31	0.15	98.47	99.08	90.55	99.95
6,000	75	99.10	0.92	0.11	98.89	99.31	93.15	99.95
7,000	75	99.24	0.58	0.07	99.10	99.37	97.45	99.90
Total	525	98.04	2.17	0.09	97.85	98.22	85.30	99.95

6. Classifier Design

Consequently, the optimal parameters tested by ANOVA in Section 5.1 are used to establish a color autocorrelation histogram model (CAH). The CAH model is used to determine the validity of the FG/BG classification. In this paper, the model is constructed from two classifier techniques: nearest neighbor (1NN) and support vector machine (SVM), for comparison purposes.

6.1. Nearest Neighbor Voting

The nearest neighbor (1NN) is a simple algorithm that can be used for classification even on the basis of a few examples. The advantage of this approach is that it is not complicated. However, the computation time can vary depending on the size of the reference dataset. The goal is to compare the similarity between unknown objects and neighboring candidates, by calculating the distance in some feature spaces. The closest neighbor indicates the class of a query object.

Foreground and background class, shown in Algorithm 2, are defined and denoted as “fg” and “bg”. Each class has an associated codebook, which is represented by prototypical keypoints (PKPs). Given an image (I), PKPs are detected. Each PKP is classified into either fg or bg. The final decision is made based on the majority type of the PKPs voting scheme. However, since this voting scheme appears to be highly simplistic, it was assumed that a richer description of images should be used to take the decision. In the next section, we will introduce a codebook histogram method, using SVM for classification.

6.2. Support Vector Machine (SVM)

In this approach, image regions are characterized by a distribution (histogram) of prototypical keypoints; again, there are two codebooks, one for FG and one for BG. Instead of using a sample vote technique, image descriptions (i.e. codebook histogram) will be compared vectorially, using a support vector machine. Support vector machines (SVMs) [6] in machine learning is a supervised learning model, which is able to handle data analysis and classification problems. The algorithm is based on the principle of finding the coefficients of the equation to create a dividing line of data that is sent to the training process by focusing on the best dividing line of data. SVM with the linear kernel is indeed one of the simplest classifiers with a lower risk of overfitting than non-linear kernels such as polynomial or RBF. Since we perform a grid search over many parameters, using the linear SVM is applicable for the task. SVM can be used to identify patterns or groups of data; the data are divided into two sides by the hyperplane. Initially, the hyperplane is usually formed in the linear model also used in this paper, and the appropriated linear model is selected by SVM with the maximum distance between the two classes (the so-called functional margin). The margin means the maximal width of the slab parallel to the hyperplane that has no interior data points.

Given a set of feature X $= (x_1, y_1), ..., (x_n, y_n)$ when $x \in R^m$ where n is a number of sample data, m is the dimension of input data, x is the feature vector, and y is a group of data which includes $\{-1, 1\}$. Then set the linear equation to establish a straight line on the hyperplane in order to separate the data into two groups which are represented by the value of y. The data that determines the slope and plane formed on the hyperplane is derived from the pair of (w, b), where w is the slope value and b is the constant value (Y-axis value). Defining the equation identifies which part of the group is located on the hyperplane as shown in Eq. (16) and Eq. (17). Applying all the conditional equations to the geometric analysis by considering where the data are grouped in accordance with SVM is described in Eq. (18).

$$w^T x + b \geq y, \ where\ y = 1 \tag{16}$$

$$w^T x + b \leq y, \ where\ y = -1 \tag{17}$$

$$y(w^T x + b) - 1 \geq 0 \tag{18}$$

However, sometimes it is not possible to distinguish all the data correctly. Therefore, reducing these errors can be a crucial step in improving the classification performance. Thus, it is necessary to define a variable to accept the error value to SVM. The reconstruction of SVM is explained by Eq. (19). The formula describes the vector of the weight values of w by trying to reduce the value in the first term of the equation to the smallest value. Given C is a variable that can be customized to adjust the desired error value as low as possible. The measurement of errors that deviate from the correct position is defined by ξ_i or slack variable.

$$\phi(w, \xi) = \tfrac{1}{2} w^T w + C \textstyle\sum_{i=1}^{n} \xi_i \tag{19}$$

To train an SVM model, there are several hyperparameters to be considered. For instance, the different values for parameter C will yield different performance results. Using the fitcsvm() procedure, we determined the following values for C. This paper used $C = 0.3161$ for the color model and $C = 0.0010$ for the SIFT model of the TSIB dataset. For the Chars74K dataset with the Kannada script, $C = 981.45$ was utilized for the color model and $C = 332.51$ for the SIFT model, and $C = 0.0010$ was

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

utilized for the color model and $C = 0.0011$ for the SIFT model of the ICADAR2015 dataset.

7. Experimental Results

In this section, we demonstrate the result on the three benchmark datasets. We designed the experiments in order to investigate the influence of various color spaces, color space against SIFT, as well as classifier methods on classification performance. Furthermore, we take advantage of the SIFT feature to remind the object description and color autocorrelation histogram to compute the color distribution of an object, and to create adjoined features and test for classifying text and non-text blocks. To evaluate the performance of the proposed method, we use the precision, recall and f-measure. Here precision $p = TP/(TP + FP)$, and recall $r = TP/(TP + FN)$ where TP is the set of true positive classification while FP is Type I error and Type II error for FN. Therefore, the definition of the f-measure is ($f = 2/(1/p + 1/r)$).

Experiment 2: The image-patch regions of interest are represented by a histogram of color autocorrelation (CAH features). It is interesting to observe the classification performance of CAH features. There are nine color spaces using the selection criteria from Section 5.1 for creating CAH feature, to be assessed. From the selective measurements it can be concluded that the optimal patch region size of 50×50 pixels is optimal for describing the foreground (FG) or text elements. Since the average height of a character in the three datasets is 88.69 pixels, hence the found window size is apparently large enough. For example, as a rule of thumb, the patch size should be higher than half the height of the character. Smaller patch sizes may not indicate clearly whether a region of interest concerns FG text or BG scene content.

Another important factor is autocorrelation, which was computed using three different autocorrelation algorithms: IFFT, XCORR, and ACF. Each color channel is quantized with a similar level to 128 dimensions and is adjoined into 384 dimensions to compare the classification results. The ACF is normalized by mean and variance, or in other words has “autocovariance function”. The ACF pattern has a large spike at lag 1 followed by a decreasing wave after a few lags that alternate between positive and negative correlations. Then, a multiple-dimension IFFT is a symmetric pattern or a complex conjugate; it has considerable computational advantages over the convex optimization method [36]. On the other hand, XCORR uses cross-correlation which computes the normalized autocorrelation sequence to lag 255 by ‘coeff’ followed by shifting an odd position to lag 128. Therefore, the sequence is so that the autocorrelations at zero lag equal to 1. We qualitatively compare the efficiency of three autocorrelations in Section 5.1 (Table 3). The result shows that the ACF provides better classification results than IFFT or XCORR concerning both mean and maximum values by ANOVA. On the basis of this result the ACF method seems to be the most appropriate to compute a color autocorrelation histogram feature.

To demonstrate the effectiveness of the distance functions in order to create the 1NN classifier, six distance types are examined. In ascending order of the mean, the Spearman gives the lowest mean at 93.56%, followed by the correlation, cosine, and Euclidean at 98.36%, 98.37%, and 98.82%, respectively. The best distance measure is the city block (Manhattan) function with a mean score of 98.84%. Therefore, city block is selected for applying 1NN matching. In addition, the normalization is also advantageous to handle values in a comparable range: the best method is ii(std), yielding a mean performance of 98.83%.

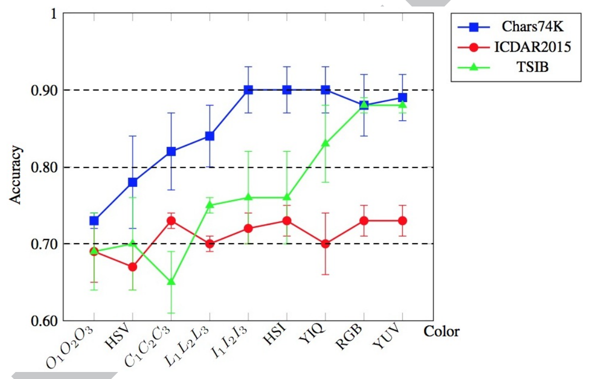


Fig. 14: The accuracy of FG and BG classification with a 1NN classifier using the different types of color spaces, for the three datasets. Color scheme *RGB* and *YUV* appear to perform well on sets Chars74K and TSIB, while the performance on ICDAR2015 is low, for all the color schemes.

The final factor is the size of codebook, which can greatly affect the performance. We tested the codebook size 1,000 to 7,000 for the range of a thousand. A higher codebook size gives a higher accuracy, as depicted in Section 5.1 (Table 6). However, if the codebook size is too large, several related region features will be matched to different visual words and take a lot of time. In contrast, if the codebook size is very small, many unrelated regions will be matched to the same visual words. Therefore, to make the processing time-saving but still effective, we choose 4,000 for creating the codebook size of FG and BG.

The experimental results of FG and BG classification with 1NN classifier of three datasets on the criteria in Section 5.1 can be demonstrated in Fig. 14. From the figure we observe that there are three color spaces; HSI, $I_1I_2I_3$, and YIQ provide 90% high accuracy for Chars74K. On the other hand, the ICDAR2015 needs four colors: $C_1C_2C_3$, HSI, RGB, and YUV, which gives 73% accuracy. Finally, RGB and YUV color are appropriated for the TSIB dataset with performance accuracy at 88%.

Experiment 3: Detecting POIs by HL is more useful for POIs (keypoint) detection than SIFT (Fig. 5) because a large number of POIs would obtain higher-level object attributes, e.g., descriptions and color. Therefore, this paper utilizes the HL detector to detect the points of interest (POIs) of FG/BG images. Each POI is an extracted feature in 2 types: SIFT and

ACCEPTED MANUSCRIPT

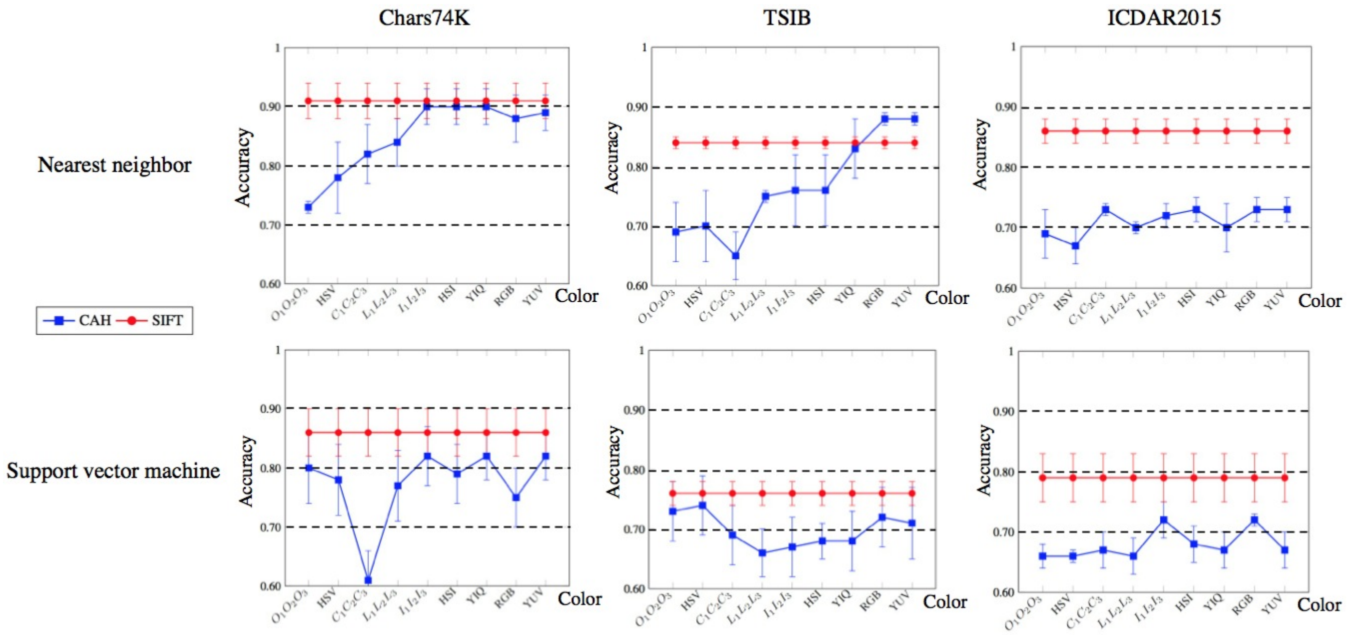


Fig. 15: The results of FG and BG classification with the 1NN classifier compared to the SVM classifier using the different types of color spaces for the three datasets. The performance of SIFT is better than the color feature (CAH).

Algorithm 2 1NN voting (v, pkp)

```
1:  $correctFg = 0; correctBg = 0;$
2:  $cb = \{pkp_1, pkp_2, .., pkp_m\};$
3:  $cf = \{v_1, v_1, .., v_n\};$
4:  $[ncb, ncf] = normalize(cb, cf);$
5:  $sepCB = size(cb, 1)/2;$
6:  $[D, I] = pdist2(ncb, ncf, distance,' Smallest', 1);$
7:  $firstHalf = sum(I \leq sepCB);$
8:  $secondHalf = sum(I > sepCB);$
9:  if $firstHalf > secondHalf$ then
10:     if $GT ==' fg'$ then
11:         $correctFg = correctFg + 1;$
12:         $decision =' fg'$
13: else
14:     if $GT ==' bg'$ then
15:         $correctBg = correctBg + 1;$
16:         $decision =' bg'$
17: return $decision$
```

CAH. Extracting the feature descriptor using SIFT gives 128 dimensions at each POI; on the contrary, extracting features in different color spaces after applying ACF at each ROI has different characteristics but similar dimensions to 384. At this stage of comparing text/non-text classification, we want to test the basic features first, so that they, for example CNN-based methods, can be contrasted with this in later research.

The accuracy of CAH and SIFT features is compared using two different model classifier methods: the ordinary nearest neighbor (1NN) and a proven method (SVM) as demonstrated in Fig. 15. Figure 15 describes the results of FG and BG classification with the 1NN classifier compared to the SVM classifier using the different kinds of color space for the three datasets. Since the SIFT model is created only once, it obtains similar results in each graph. The performance of SIFT using the 1NN classifier is better than the color features (CAH) using the same classifier at approximately 90% for the Chars74K data set. However, the performance of SIFT by the SVM classifier presents lower accuracy than 1NN at approximately 86% for similar data sets. Surprisingly, when considering a TSIB data set, CAH beats SIFT by two color spaces that are RGB and YUV (around 88%). Meanwhile, the performance of CAH using 1NN and SVM is still lower on average for FG/BG classification on the ICDAR2015 data set than for the other two data sets.

Table 7 and Table 8 show the evaluation via precision, recall, and f-measure for the classification results of the CAH and SIFT features using 1NN classifier and SVM classifier, respectively, on the three benchmark datasets. The performance of the CAH feature is comparable to state-of-the-art features such as SIFT. Although SIFT yields better results than the CAH feature for both 1NN and SVM classifiers, we also observe that the f-measure on the CAH feature of TSIB (a colorful data set)

ACCEPTED MANUSCRIPT

in Table 7 and Table 8 gives better results compared to SIFT. This gives an indication for the application domain of the CAH feature for classifying text and non-text blocks in images.

Experiment 4: According to the results of SIFT and CAH using 1NN in Experiment 3, we apply the benefit of the CAH and SIFT classifiers to be the adjoint SIFT+CAH feature to evaluate text/non-text classification by counting the number of correct matching feature descriptors. Figure 16 describes the classification process. A testing image is extracted feature descriptors by the CAH features and SIFT features. The CAH and SIFT features are classified as text (FG) or non-text (BG), respectively, by the CAH and Sift classifiers using 1NN. Then all classified features are counted as FG/BG. If the total number of FG is larger than BG, the image will be classified as text and vice versa. So, from Fig. 16, both the CAH and SIFT features of the testing image are classified as text (FG) because CAH classifier presents total number of correct matching for FG is 148 and 117 for BG, which the total number of FG is larger than BG. As well as SIFT classifier shows total number of correct matching for FG is 247 and 18 for BG, which the total number of FG is larger than BG. The result of adjoin feature for FG and BG is obtained from the total number of FG (148) and BG (117) by CAH classifier plus the total number of FG (247) and BG (18) by SIFT classifier (FG: 148 + 247 = 395, BG: 117 + 18 = 135), which the total number of FG is larger than BG. Therefore, the classification result for adjoin feature is text (FG).

The proposed technique, therefore, utilizes the advantage of counting matching descriptors for the adjoined features. The feature is made up from the summation of the total count of matching CAH and SIFT feature descriptors for both FG and BG. Finally, the class with the higher number of matching descriptors is considered to be the result of the classification, e.g., a given test image is classified as FG.

The results in Fig. 17 show that the proposed concept gives good classification accuracy (green graph) for the Chars74K and TSIB data sets. The accuracy with Chars74K presents a dramatic increase at 94% for other color spaces except $O_1O_2O_3$, HSV, and $C_1C_2C_3$. Furthermore, applying the adjoint feature to the TSIB dataset provides a better result for RGB and YUV at 92% of accuracy. Unfortunately, the SIFT feature outperforms the other features on ICDAR2015. However, we eventually find that when using a 1NN classifier for the adjoint SIFT+CAH feature, particularly color RGB or YUV, in order to classify text and non-text blocks for Asian scripts, Chars74K and TSIB is practical. Table 9 presents the comparison of adjoint features on the benchmark datasets by precision, recall, and f-measure. It shows that the proposed technique, i.e., the additional use of color, improves the performance, most clearly for TSIB.

Figure 18 shows the results of the experiments accurately classifying FG and BG of all three datasets. Although our proposed technique achieved a good accuracy for Asian script, it also has some observations with visual characteristics of images that can improve performance. In Figure. 19, the image is classified incorrectly. For instance, there are some low contrast images on FG where the background and foreground color look similar. Some text blocks are struck by reflection, through which both the CAH and SIFT classifiers cannot produce accurate results. In addition, the word image embedded in various color and cluttered backgrounds cannot provide a good result. Its feature descriptors are dominated by background features, and as a result, it has a chance to match BG keypoints in the codebook. Regarding a number of keypoints between FG and BG, if there are not many differences, it does not provide adequate information. Hence, the method could misclassify FG and BG. There are some BG blocks which have explicit color distinction after being extracted by SIFT. There are many keypoint descriptors. Therefore, the possibility of matching keypoints between the testing image and text keypoints in the codebook is increased. In the case of the background block with a logo, the background is filled in letters, and it is not a ground truth of text. When the block is an extracted feature descriptor, it could be matched to the descriptor of the FG codebook. So, this block may be classified as a foreground.

7.1. Comparison with State-of-the-art

To investigate the efficiency of the proposed method, we hence evaluated the performance of text and non-text classification compared to the other methods (different datasets and data sizes). The performance of Wu's method, tested with 84 test images, contained 367 text regions (minimum of three characters per region) of the ICDAR2003. Further, Alves and Hashimotos method also tested text and non-text classification with the same dataset as Wu's method. While, Sriman and et al. tested the experiments on the ICDAR2015 dataset, which 1,095 random FG zones and 1,035 random BG zones were used to evaluate the classification performance. For our proposed method used the ICDAR2015 dataset, which contains 1,943-word images and 1,943 random background images were used in the experiment. The experimental results in Table 10 show that the performance of the proposed method performs competitively against the results of existing methods in term of precision and recall.

7.2. Computational Complexity

This section presents the explanation about the computational complexity of the proposed algorithm through their execution time in order to determine the efficiency of the method. The advantage of this approach is the real estimation of the algorithm running time; however, it depends on the sample input instances, selection of functions for software development, as

ACCEPTED MANUSCRIPT

Table 7: The criteria performances of CAH and SIFT features using 1NN classifier for Chars74K, TSIB, and ICDAR2015 datasets.

Color space	Chars74K			TSIB			ICDAR2015		
	p	r	f	p	r	f	p	r	f
$O_1O_2O_3$	0.51	0.90	0.65	0.90	0.67	0.75	0.77	0.68	0.70
HSV	0.78	0.83	0.78	0.93	0.68	0.77	0.95	0.63	0.75
$C_1C_2C_3$	0.72	0.89	0.78	**0.94**	0.62	0.73	0.77	0.71	0.73
$L_1L_2L_3$	0.74	0.92	0.81	0.70	0.78	0.74	0.76	0.69	0.70
$I_1I_2I_3$	0.87	**0.94**	0.89	0.91	0.75	0.81	0.91	0.67	0.77
HSI	0.87	0.92	0.89	0.91	0.74	0.80	**0.95**	0.65	0.76
YIQ	0.87	0.93	0.89	0.90	0.83	0.85	0.91	0.69	0.77
RGB	0.83	0.92	0.87	0.87	0.89	0.88	0.94	0.65	0.76
YUV	0.85	0.92	0.88	0.87	**0.89**	**0.88**	0.92	0.70	0.78
$SIFT$	**0.93**	0.90	**0.91**	0.89	0.81	0.85	0.94	**0.80**	**0.86**

Table 8: The criteria performances of CAH and SIFT feature using SVM classifier for Chars74K, TSIB, and ICDAR2015 datasets.

Color space	Chars74K			TSIB			ICDAR2015		
	p	r	f	p	r	f	p	r	f
$O_1O_2O_3$	0.63	**0.96**	0.73	0.84	0.75	0.77	0.95	0.59	0.72
HSV	0.79	0.78	0.77	0.87	0.74	0.78	0.82	0.63	0.71
$C_1C_2C_3$	0.63	0.57	0.58	0.90	0.67	0.75	0.97	0.60	0.74
$L_1L_2L_3$	0.77	0.76	0.77	0.84	0.67	0.71	0.80	0.70	0.71
$I_1I_2I_3$	0.83	0.82	0.81	0.93	0.65	0.74	0.88	0.66	0.75
HSI	0.80	0.79	0.79	0.93	0.63	0.75	**0.98**	0.61	0.74
YIQ	0.86	0.80	0.82	**0.95**	0.64	0.76	0.95	0.61	0.74
RGB	0.81	0.73	0.76	0.93	0.68	**0.78**	0.93	0.64	0.75
YUV	0.86	0.80	0.82	0.94	0.68	0.77	0.95	0.60	0.73
$SIFT$	**0.92**	0.84	**0.87**	0.57	**0.91**	0.70	0.77	**0.84**	**0.78**

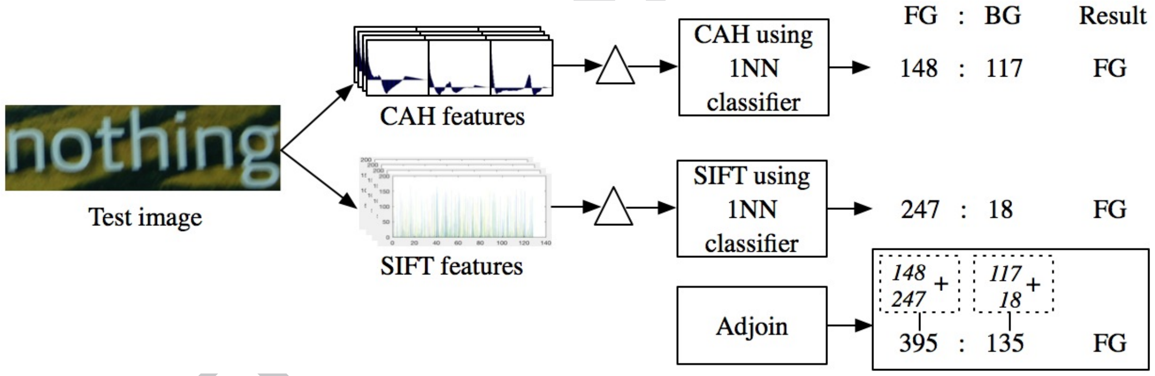


Fig. 16: An example of FG and BG classification using 1NN classifier with both the CAH and SIFT features of the testing image are classified as text (FG) because CAH classifier presents the total number of correct matching for FG is 148 and 117 for BG, which the total number of FG is larger than BG. As well as SIFT classifier shows the total number of correct matching for FG is 247 and 18 for BG, which the total number of FG is larger than BG. The result of the adjoining feature for FG and BG is obtained from the total number of FG (148) and BG (117) by CAH classifier plus the total number of FG (247) and BG (18) by SIFT classifier (FG: 148 + 247 = 395, BG: 117 + 18 = 135), which the total number of FG is larger than BG. Therefore, the classification result for adjoin feature is text (FG).

Table 9: The criteria performances of adjoint feature for Chars74K, TSIB, and ICDAR2015 datasets.

Adjoint features	Chars74K			TSIB			ICDAR2015		
	p	r	f	p	r	f	p	r	f
$O_1O_2O_3 + SIFT$	0.86	0.95	0.90	0.94	0.72	0.80	0.90	0.76	0.81
$HSV + SIFT$	0.90	0.90	0.88	0.95	0.73	0.81	0.97	0.67	0.79
$C_1C_2C_3 + SIFT$	0.88	0.96	0.91	**0.96**	0.66	0.77	0.92	**0.79**	**0.85**
$L_1L_2L_3 + SIFT$	0.92	0.95	0.93	0.86	0.87	0.86	0.93	0.76	0.83
$I_1I_2I_3 + SIFT$	0.92	0.96	0.93	0.94	0.80	0.85	0.96	0.73	0.82
$HSI + SIFT$	0.93	0.96	0.94	0.94	0.79	0.85	0.98	0.69	0.81
$YIQ + SIFT$	0.94	0.95	0.94	0.93	0.86	0.89	0.95	0.75	0.83
$RGB + SIFT$	0.92	0.96	0.94	0.91	0.93	0.92	**0.98**	0.70	0.81
$YUV + SIFT$	**0.95**	**0.94**	**0.94**	0.92	**0.93**	**0.92**	0.96	0.74	0.83

well as the employed hardware. There are four principal parts of the proposed algorithm: localization of point of interest, fea-

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

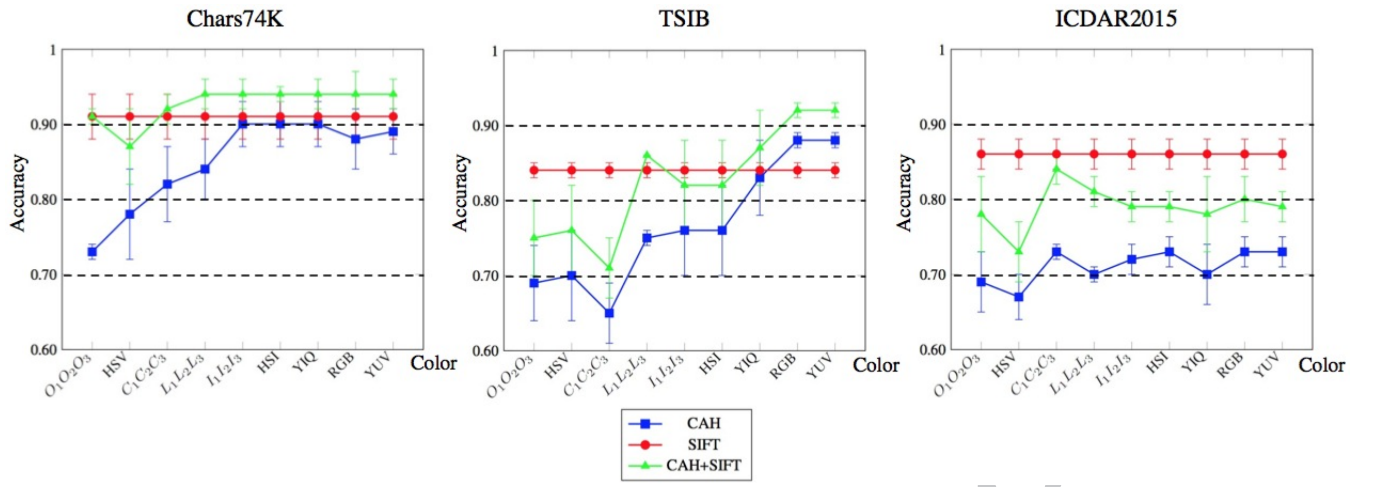


Fig. 17: The accuracy of FG and BG classification using a 1NN classifier for the adjoint SIFT+CAH feature, compared for the three datasets. The results of the adjoint method confirm the usefulness of color (*RGB* or *YUV*) for Asian scripts Chars74K and TSIB.

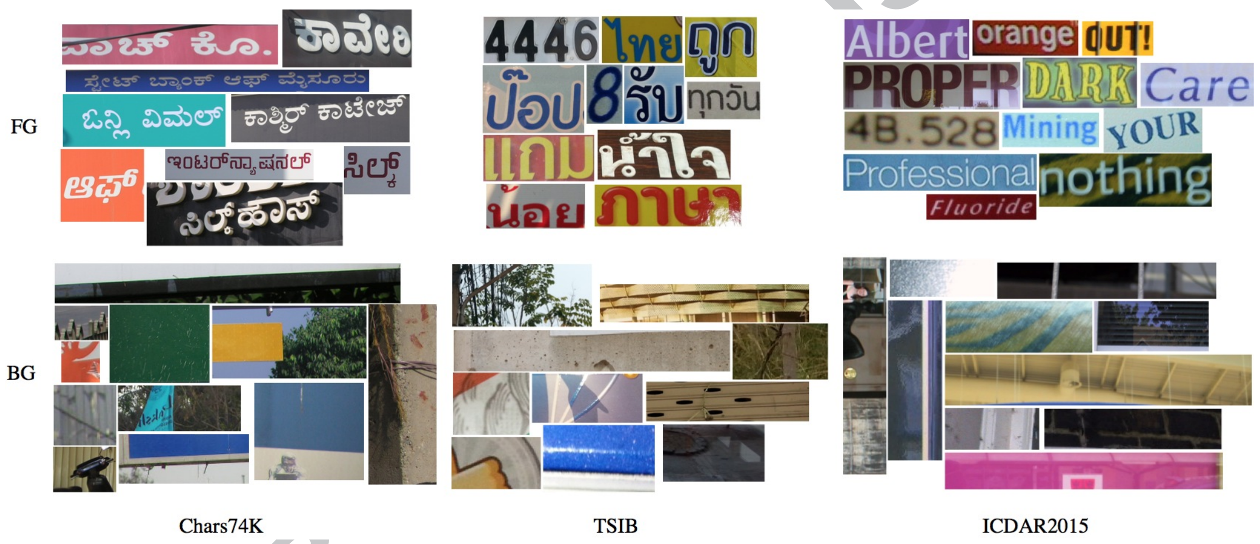


Fig. 18: Example of correct foreground and background classification on three datasets.

Table 10: Text and non-text classification results for different methods.

Methods	Precision	Recall
Wus method [38]	78.87%	76.29%
Alvess and Hashimoto's method [1]	97.00%	88.00%
Sriman and et.al's method [29]	94.89%	84.17%
Proposed method (adjoint feature)	**98.00%**	**70.00%**

ture extraction, codebook histogram modeling and SVM classifier, and classification.

7.2.1. *Localization of Point of Interest by Harris-Laplace detector (HL)*

The interest points extracted by the HL detector are not only invariant to scale, rotation and translation, but also robust to illumination changes and limited viewpoint changes. To compute the complexity of HL detector is following this step.

- Create the scale space of an image f by a Gaussian function $P(\mathbf{x}, \sigma) = g(\mathbf{x}, \sigma) * f(\mathbf{x})$ where $\mathbf{x} = (x, y)$, which the complexity is O(nmp).

- For computing the subsequent detection of interest points at the scale, let $P_x(\mathbf{x}, \sigma_D)$, $P_y(\mathbf{x}, \sigma_D)$ is the derivatives at position $\mathbf{x}$ computed using the Gaussian filter of local scale σ_D. Then these intermediate results are composed to obtain three images $P_x^2(\mathbf{x}, \sigma_D)$, $P_y^2(\mathbf{x}, \sigma_D)$, $P_xP_y(\mathbf{x}, \sigma_D)$. These images are filtered by a Gaussian window size $\sigma_I(g(\sigma_I))$. Then, the single scale Harris corner indicator at position $\mathbf{x}$ in the image f is computed with $P(\mathbf{x}) = det(\mu(\mathbf{x}, \sigma_I, \sigma_D)) - \alpha trace^2(\mu(\mathbf{x}, \sigma_I, \sigma_D))$. At each scale, calculate local maxima for a corner, which gives runtime O(n^2). So, the complexity of this stage is O(nmp) + O(n^3) + O(n^2) $\approx$ O(n^3).

- Selecting the image blob characteristic scales, and the normalized LoG kernels with image scale space are defined by $|LoG(\mathbf{x}, \sigma_n)| = \sigma_n^2 \left|P_{xx}(\mathbf{x}, \sigma_n) + P_{yy}(\mathbf{x}, \sigma_n)\right|$. So, the runtime is O($n$) + O($nmp$) $\approx$ O(n).

ACCEPTED MANUSCRIPT

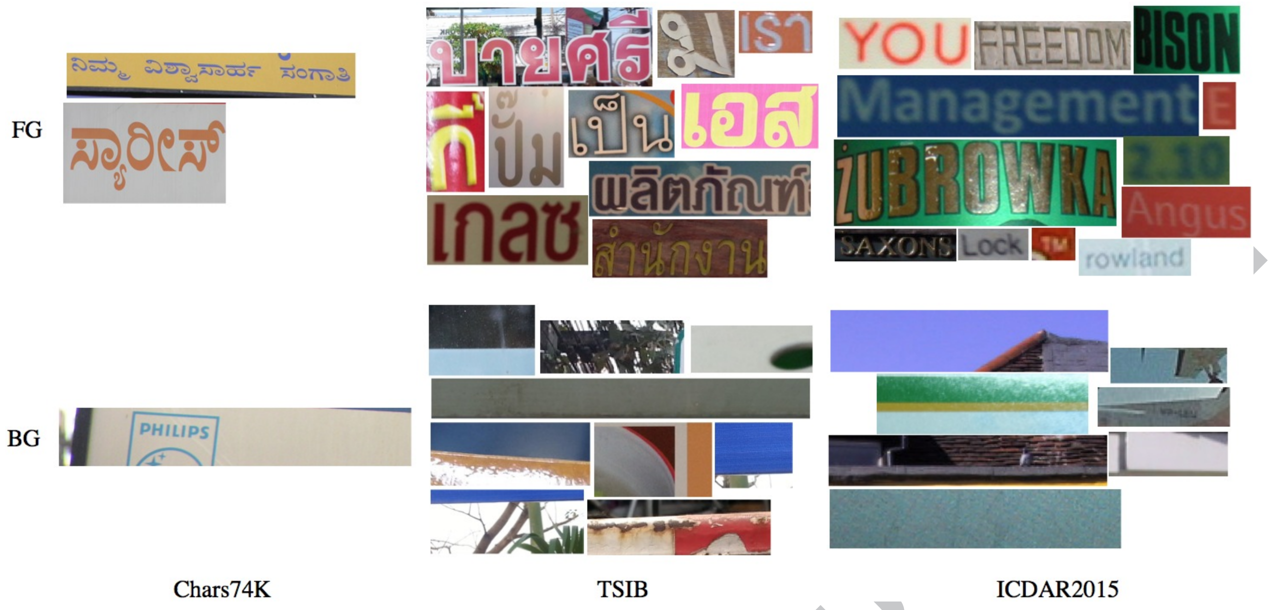


Fig. 19: Example of missing foreground and background classification on three datasets.

- Salient positions in the scale space are detected by computing a multi-scale stack of the Harris corner indicator $\{H(\mathbf{x}, \sigma_k, \gamma\sigma_k)\}_{k=1}^{n}$, γ is a scalar (O(n)). For each scale σ_k, the local maxima of the Harris indicator $H(\mathbf{x}, \sigma_k, \gamma\sigma_k)$ are computed by $\{(\hat{\mathbf{x}}, \sigma)\} = argmaxlocal_{\mathbf{x}} H(\mathbf{x}, \sigma_k, \gamma\sigma)$. Then, select points at which the normalized LoG is maximal across scales and the maximum is above a threshold ($\hat{\sigma} = argmax_{\sigma} |LoG(\hat{\mathbf{x}}, \gamma\sigma)|$). The spatial location $\hat{\mathbf{x}}$ which do not hit a scale maximum, is given up. So, the runtime is O(n^2).

7.2.2. *Feature Extraction*

1. SIFT feature: There is the research to propose analyzing SIFT computational complexity [8], which presents into five stages: space scale extrema detection, keypoint localization, exact localization, keypoints descriptors and orientation, and matching. However, the matching stage does not employed in this paper. The first stage is to create a scale space by the original image is generated in several octaves. Each octave's image size is half of the previous one, and the individual images are progressively blurred out using the Gaussian blur operator. In computational terms of the Gaussian blur by applying a series of single-dimensional Gaussian matrices in the horizontal direction, then repeating the process in the vertical direction is $O(m_{kernel} m_{image} n_{image})$, where m is width and n is height. The computation complexity of this stage is $T(m, n) = O(mn - (mn/min(m, n)))$, where m and n represent the width and height respectively in one image.

The second stage is keypoint localization by using the blurred images from the previous stage, to be computed the extrema (min and max) local values for each difference of Gaussians $D(x, y, \sigma)$ over scale and space at each octave. Therefore, each pixel in an image is compared with their eight neighbors into the current image as well as compared with nine pixels in the above and below scales. This process is done for all octaves iteratively in order to obtain potential keypoint localization. The complexity of this stage is $T(m, n) = O(mn - (mn/min(m, n)))$.

The third stage is exact localization after the potential keypoints are found in the previous step, which some of them were located on edge, or they have low contrast. So, these keypoints need to be defined by the Taylor expansion [22] to get a more accurate location of extrema, and if the intensity at this extrema is less than a threshold value, the keypoint is eliminated. The complexity of this procedure is $O(k)$, where k is the number of extrema found in the previous stage.

The fourth stage is keypoint orientation and descriptors. Currently, the scale of detected keypoints is invariance. The rotation of the keypoints should be assigned to have invariance as well by computing the prominent orientations $\theta(x, y)$ and magnitudes $m(x, y)$. The $m(x, y)$ and $\theta(x, y)$ is calculated for all pixels around the keypoint of filtered image L with big O notation for $\theta(x, y)$ is $O(1)$, and $m(x, y)$ is $O(n^{\frac{1}{4}})$. Then a histogram is created to have 36 bins covering 360 degrees of orientation for the pixels in the region. It is weighted by gradient magnitude and gaussian-weighted circular window with σ equal to 1.5 times the scale of keypoint.The highest peak in the histograms correspond to dominant directions of the local gradient and any peaks above 80% of the highest peak are converted into a new keypoint, which has the same location and scale as the original, but different directions. After the keypoint orientation definition, the keypoint descriptor is created. The descriptors are computed using the gradient magnitude and orientation around the keypoint. To do this, there are 16 windows, and each window is broken into 4×4 subregion. For each subregion computes gradient magnitudes and orientations to have eight directions, which are put into 8 bin histogram. Then the 16×8 = 128 dimensional descriptors of this keypoint is completed. The time complexity in big O notation of this stage is $O(k)$, where k is the number of keypoints.

ACCEPTED MANUSCRIPT

2. CAH feature: This feature is the combination between an autocorrelation function and a color space. The input image (I) of creating the CAH feature is the cropped area image of a POI with the patched size received from Experiment 1. For each input image $I = 1toK$, loop through all POI of an image, is generated to three color level histogram of a color space (O(n) + O(3)). Next, the three color histograms are applied by autocorrelation function and they are adjoined to be the CAH feature (O(4n)). So, the complexity time of the CAH feature is O(n).

7.2.3. Codebook Histogram Modeling and SVM Classifier

1. Codebook Histogram Modeling: In the clustering stage, we utilize the function `vl_kmeans` in the VlFeat implementation, is the k-means method of Lloyds algorithm [21], to cluster CAH features and SIFT descriptors. Many parameters are used in the `vl_kmeans` function, i.e., initialization methods, number of time to restart k-means, and optimization algorithms. An approximated algorithm using approximate nearest neighbors (ANN) is one of a parameter in the optimization algorithms and suitable for huge problems, is set to the function. The ANN algorithm supports float or double data and uses Squared Euclidean distance (Eq. (20)) with time complexity in big O notation is O(n^2), to accelerate the sample-to-center comparisons. The following option for ANN algorithm is a k-d tree forest algorithm, which is used for containing the k-d tree indexing the matrix data and queries the index. The time complexity for the k-d tree in the worst case is O(n) (for all space, search, insert, and delete algorithm). The initialization method for determining the cluster centers or the means of the corresponding data points is k-means++ [2], a popular method to avoid poor clustering found by the standard k-means algorithm. The k-means++ selects k centers arbitrary at random from the data points to initialize the cluster centers. The algorithm obtains O(log k) competitive that is guaranteed to find the optimal k-means solution.

$$\textstyle\sum_{i=1}^{d}(x_i - y_i)^2 \tag{20}$$

2. SVM Classifier: The time complexity of creating SVM classifier is described as follow.

- For $I = 1$ to J, loop through all FG train images (O(n)) and BG train images (O(n)). Each image is extracted feature (O(n^2)). So, this get O(n) + O(n) + O(n^2), which means (O(n^2)).
- The codebook and train image keypoint descriptors are normalized to the range [0,1], which requires O(n).
- The image keypoint descriptors match to the codebook by computing distance (city block) between them to get a smallest pairwise distance and the matrix I contains the indices of the observations in the codebook (ncb) corresponding to the distances in D, using pdist2 function in MATLAB ($[D, I] = pdist2(ncb, des,' cityblock',' Smallest', 1)$), where des is the descriptor matrix of train image (O(n)).
- Create FG train image histogram and BG train image histogram, which has a dimension equal to the codebook size, from the smallest distances derived from the previous step. For each dimension of the matrix I (O(n^2)), add 1 to the matched index in the FG histogram (O(1)) and repeat this step to the BG histogram. So, the complexity is O(2n^2) + O(2) ≈ O(n^2).
- The FG histogram and BG histogram from the previous step are used to make SVMModel using fitcsvm function, and to make ScoreSVMModel using fitSVMPosterior function. The runtime is O(1).

7.2.4. Classification

1. Nearest Neighbor (1NN): The time complexity of the 1NN algorithm with the distance function named pdist2 in MATLAB is described as follow.

- For $I = 1$ to J, loop through all FG and BG testing image observations (O(n)). Each image is extracted feature (O(n^2)). So, this get O(n) + O(n^2), which means (O(n^2)).
- The codebook and image keypoint descriptors are normalized to the range [0,1], which requires (O(n)). Determine the position of FG descriptors and BG descriptors in the codebook by dividing the codebook size into two parts using the expression, $sepPointCB = size(ncb, 1)/2$, where ncb refers to the codebook metric, $sepPointCB$ refers to the half position number of the codebook (O(n)).
- The image keypoint descriptors match to the codebook by computing distance (city block) between them to get a smallest pairwise distance and the matrix I contains the indices of the observations in the codebook corresponding to the distances in D, using pdist2 function in MATLAB ($[D, I] = pdist2(ncb, des,' cityblock',' Smallest', 1)$), where des is the descriptor matrix of test image. Then, count the total number of the indices I to decide the descriptor is text or non-text, if $firstHalf = sum(I <= sepPointCB)$ the descriptors are located in the first half of the codebook that means these descriptors are text. If $secondHalf = sum(I > sepPointCB)$ the descriptors are located in the second half of the codebook that means these descriptors are non-text. So, this step gets runtime O(n) + O(1) ≈ O(n).
- To make a decision, if $firstHalf > secondHalf$ that means this image is text and vice versa, which the runtime O(1).

2. Support Vector Machine (SVM): The time complexity of the SVM classification is described as follow.

- For $I = 1$ to J, loop through all test image observations (O(n)). Each image is extracted feature (O(n^2)). So, this get O(n) + O(n^2), which means (O(n^2)).
- The codebook and image keypoint descriptors are normalized to the range [0,1], which requires O(n).

ACCEPTED MANUSCRIPT

- For each observation in the image descriptors, pdist2 function in MATLAB ($[D, I] = pdist2(ncb, des, ' cityblock', ' smallest', 2)$) finds the two smallest city block distances by computing and comparing the distance between observations in the codebook (ncb) and observation in the image descriptors (des). The matrix I contains the indices of the observations in the codebook corresponding to the distances in D (O(n)).

- Create test image histogram, which has a dimension equal to the codebook size, from a distance ratio comparison between two smallest distances derived from the previous step. For each dimension of the matrix I (O(n^2)), the first distance compares to the second distance times match ratio in order to analyze the keypoints that have been matched to point 1 and point 2 are different or not. If it is different distinctly, the keypoint is matched and add 1 to the matched index in the histogram (O(1)). But, if there are very similar, it is ambiguous that means the keypoint does not match. So, the complexity is O(n^2) + O(1).

- The histogram from the previous step is predicted by SVM classifier, which returns class labels and a matrix of scores (score) indicating the likelihood that a label comes from a particular class. The runtime is O(1).

7.3. Computation Times

The computation times of the proposed scheme for classification is elaborated in Table 11, 12 and 13, by depending on the optimal parameters received from Experimminent 1. The experiments are tested on the ICDAR2015 with total 3,498 train images (half each for text/non-text) for creating a model, and total 388 test images (half each for text/non-text). The minimum and maximum text sizes are 15×12 pixels and 1,735×833 pixels, respectively. For non-text, the minimum and maximum sizes are 6×4 pixels and 1,726×824 pixels, respectively.

Table 11: The execution time (in minutes) of the proposed scheme for SIFT classification using the 1NN classifier for the ICDAR2015 dataset.

Computation times	fold 1 ()	fold 2 (min)	fold 3 (min)	fold 4 (min)	fold 5 (min)
Model creation	**310.06**	**369.10**	**341.03**	**371.49**	**295.31**
FG extraction	214.06	206.20	186.04	173.42	156.12
BG extraction	131.44	129.54	108.30	130.09	105.23
FG clustering	13.53	33.12	31.21	33.12	24.55
BG clustering	5.36	34.46	15.08	34.46	9.01
Classification	**12.00**	**17.48**	**9.02**	**15.23**	**45.12**
one image(avg)	0.031	0.045	0.023	0.039	0.116
Total process	**322.06**	**386.58**	**351.08**	**388.21**	**340.43**

The results of the computation times (in minutes) of SIFT feature classification based on the same measurements, using the 1NN classifier (Table 11) compared to the SVM classifier (Table 12), which computed in 5-folds can be described that classification by the 1NN classifier took as less processing time as the SVM classifier. Table 13 depicts the execution times (in minutes) of the proposed scheme for CAH+SIFT feature (the

Table 12: The execution time (in minutes) of the proposed scheme for SIFT classification using the SVM classifier for the ICDAR2015 dataset.

Computation times	fold 1 (min)	fold 2 (min)	fold 3 (min)	fold 4 (min)	fold 5 (min)
Model creation	**373.39**	**365.05**	**384.03**	**300.59**	**312.48**
FG extraction	210.10	202.19	197.37	155.34	164.36
BG extraction	129.15	128.58	138.43	118.15	118.18
FG clustering	13.12	14.07	25.03	14.20	15.20
BG clustering	21.02	19.41	22.40	12.50	14.34
Classification	**15.19**	**20.58**	**10.54**	**56.41**	**48.37**
one image(avg)	0.039	0.053	0.027	0.145	0.125
Total process	**388.58**	**386.00**	**394.57**	**357.00**	**361.25**

adjoint feature in different color space) classification using the 1NN classifier for ICDAR2015 dataset in 5-fold. The adjoint feature $HSV + SIFT$ provided the shortest execution times in total process at 609.42 minutes. It can be noted that the computation times could be affected by the size and texture of an image, i.e., feature extraction of text gives much execution times rather than non-text. Consequently, computing the iterations of the sample-to-center comparisons in k-means clustering also have the ability to increase or decrease the processing speed.

8. Conclusion

In this paper, an algorithm for a robust text and non-text block classification is proposed. It is based on the optimal hybrid combination of two features, CAH and SIFT. An elaborate selection experiment was performed in order to find the optimal classification models. The CAH and SIFT features were extracted around points of interests (POIs) detected by the Harris-Laplace (HL) detector. The results shows that the HL provides more useful POIs than the traditional difference-of-Gaussians method and has therefore been selected to be used here. All the POIs or keypoints on the image are collected for feature extraction process. We performed experiments with multiple sets of criteria to find the optimal parameters for generating the CAH and SIFT features. Extracted features were used as initial parameters for generating the classification models. Among the possible sets of parameter settings, we selected the optimal set based on the obtained accuracy, using statistical testing. Finally, we also retained the features that represent color information in an intensity-invariant manner, by using the autocorrelation functions (ACF) on color histograms of color intensity and achieved a satisfactory FG/BG classification of around 90% accuracy. The accuracy of FG/BG classification with a 1NN classifier using the various types of color spaces tested on three datasets shows that color schemes RGB and YUV perform well on the Chars74K and TSIB datasets, while the performance on the ICDAR2015 benchmark data set is slightly lower.

In addition, in general, feature extraction using the SIFT algorithm provides a good result in scale and rotation invariant. It improves the accuracy of the models when it is combined with the CAH features. In the classification process, 1NN and SVM are used. The experiments show that 1NN gave a better result than SVM with both the CAH and SIFT features. In addition, the computation times of classification using 1NN compared to SVM, which tested on SIFT feature and based on the

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Table 13: The execution time (in minutes) of the proposed scheme for CAH+SIFT feature (the adjoint feature in different color space) classification using the 1NN classifier for the ICDAR2015 dataset in 5-fold.

Computation times (minutes)	HSV + $SIFT$	YUV + SIFT	HSI + $SIFT$	YIQ + $SIFT$	$I_1I_2I_3$ + $SIFT$	RGB + $SIFT$	$L_1L_2L_3$ + $SIFT$	$O_1O_2O_3$ + $SIFT$	$C_1C_2C_3$ + $SIFT$
Model creation	**553.34**	**561.53**	**573.08**	**573.42**	**575.58**	**576.36**	**583.00**	**596.15**	**600.55**
FG extraction	309.25	313.10	323.13	318.00	323.24	323.01	328.34	335.20	338.42
BG extraction	179.40	183.06	187.15	191.11	186.27	186.24	188.48	195.22	192.06
FG clustering	45.43	47.15	44.37	46.18	47.03	47.45	46.39	46.23	50.03
BG clustering	18.46	18.22	18.03	18.13	19.04	19.26	18.59	19.10	20.04
Classification	**56.08**	**53.17**	**56.29**	**56.42**	**56.34**	**56.23**	**56.29**	**58.00**	**56.03**
One image(avg)	0.145	0.137	0.145	0.145	0.145	0.145	0.145	0.149	0.144
Total process	**609.42**	**615.10**	**629.37**	**630.24**	**632.32**	**632.59**	**639.29**	**654.15**	**656.58**

same measurements appear that classification by 1NN is faster than SVM. Regarding text in different languages, using different color spaces on Asian text gives more efficient information than in Western script, and this improves the performance of the classification models. Therefore, the parameters of the models can result in different classification performance; for example, a small C value of the SVM makes a large margin while a big C make a smaller margin or the appropriate number of support vector also improves the accuracy of the models based on the experimental results shown in this paper. Finally, it should be noted that the approach is highly convenient for additional, later classification by other methods, such as convolutional neural networks: once interesting text regions are detected and pre-classified by a computationally efficient method as proposed here, further refinement can be realized by more demanding procedures that now only need to process a small fraction of a potentially large image, e.g., '8k' type image of 7,680×4,320 pixels, a format that will be increasingly prevalent in applications.

9. Acknowledgments

We sincerely thank the anonymous reviewers for their highly constructive feedback to improve this work. This research was supported by Netherlands Fellowship Programmes (NFP) Grant no. CF8777/2013. This support is gratefully acknowledged.

References

[1] Alves, W. A. L. and Hashimoto, R. F. [2010], Classification of regions extracted from scene images by morphological filters in text or non-text using decision tree, *in* 'WSCG', pp. 165–172.

[2] Arthur, D. and Vassilvitskii, S. [2007], K-means++: The advantages of careful seeding, *in* 'Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms', SODA '07, Society for Industrial and Applied Mathematics, pp. 1027–1035.
URL: *http://dl.acm.org/citation.cfm?id=1283383.1283494*

[3] Azad, P., Asfour, T. and Dillmann, R. [2009], Combining harris interest points and the sift descriptor for fast scale-invariant object recognition, *in* 'Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems', IROS'09, IEEE Press, Piscataway, NJ, USA, pp. 4275–4280.
URL: *http://dl.acm.org/citation.cfm?id=1732643.1732747*

[4] Beaudoin, N. and Beauchemin, S. S. [2002], An accurate discrete fourier transform for image processing, *in* 'Object recognition supported by user interaction for service robots', Vol. 3, pp. 935–939.

[5] Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D. J. and Ng, A. Y. [2011], Text detection and character recognition in scene images with unsupervised feature learning, *in* 'Proceedings of the 2011 International Conference on Document Analysis and Recognition', ICDAR '11, IEEE Computer Society, Washington, DC, USA, pp. 440–445.
URL: *http://dx.doi.org/10.1109/ICDAR.2011.95*

[6] Cortes, C. and Vapnik, V. [1995], 'Support-vector networks', *Machine Learning* **20**(3), 273–297.
URL: *http://dx.doi.org/10.1007/BF00994018*

[7] Dalal, N. and Triggs, B. [2005], Histograms of oriented gradients for human detection, *in* 'Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01', CVPR '05, IEEE Computer Society, Washington, DC, USA, pp. 886–893.
URL: *http://dx.doi.org/10.1109/CVPR.2005.177*

[8] Drews, P., de Bem, R. and de Melo, A. [2011], Analyzing and exploring feature detectors in images, *in* '9th IEEE International Conference on Industrial Informatics', pp. 305–310.

[9] EmÃdio de Campos, T., Rakesh Babu, B. and Varma, M. [2009], Character recognition in natural images, *in* 'VISAPP 2009 - Proceedings of the 4th International Conference on Computer Vision Theory and Applications', Vol. 2, pp. 273–280.

[10] Farag, A. A. [2014], *Biomedical Image Analysis: Statistical and Variational Methods*, Cambridge University Press.

[11] Gevers, T. and Smeulders, A. W. [1999], 'Color-based object recognition', *Pattern Recognition* **32**(3), 453 – 464.
URL: *http://www.sciencedirect.com/science/article/pii/S0031320398000363*

[12] Gong Lejun, Feng Jiacheng, Y. R. [2016], *Corner Detection-Based Image Feature Extraction and Description with Application to Target Tracking*, Springer India, New Delhi, pp. 1069–1076.

[13] Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S. and Shet, V. [2013], 'Multi-digit number recognition from street view imagery using deep convolutional neural networks', *CoRR* **abs/1312.6082**.
URL: *http://arxiv.org/abs/1312.6082*

[14] He, T., Huang, W., Qiao, Y. and Yao, J. [2016], 'Text-attentional convolutional neural network for scene text detection', *Trans. Img. Proc.* **25**(6), 2529–2541.
URL: *http://dx.doi.org/10.1109/TIP.2016.2547588*

[15] Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V. R., Lu, S., Shafait, F., Uchida, S. and Valveny, E. [2015], Icdar 2015 competition on robust reading, *in* '2015 13th International Conference on Document Analysis and Recognition (ICDAR)', pp. 1156–1160.

[16] Khan, F. S., Muhammad Anwer, R., van de Weijer, J., Bagdanov, A. D., Lopez, A. M. and Felsberg, M. [2013], 'Coloring action recognition in still images', *International Journal of Computer Vision* **105**(3), 205–221.
URL: *http://dx.doi.org/10.1007/s11263-013-0633-0*

[17] Koenderink, J. J. [2010], *Color for the Sciences*, The MIT Press.

[18] Kwok, N. M., Ha, Q. P. and Fang, G. [2009], Effect of color space on color image segmentation, *in* '2009 2nd International Congress on Image and Signal Processing', pp. 1–5.

[19] Li, Y., Jia, W., Shen, C. and van den Hengel, A. [2014], 'Characterness: An indicator of text in the wild', *IEEE Transactions on Image Processing* **23**(4), 1666 – 1677.

[20] Lindeberg, T. [1998], 'Feature detection with automatic scale selection', *International Journal of Computer Vision* **30**(2), 79–116.
URL: *http://dx.doi.org/10.1023/A:1008045108935*

[21] Lloyd, S. [2006], 'Least squares quantization in pcm', *IEEE Trans. Inf.*

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Theor. **28**(2), 129–137.
URL: *http://dx.doi.org/10.1109/TIT.1982.1056489*

[22] Lowe, D. G. [2004], 'Distinctive image features from scale-invariant keypoints', *International Journal of Computer Vision* **60**(2), 91–110.
URL: *http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94*

[23] Lucchese, L. and Mitra, S. K. [2001], 'Colour segmentation based on separate anisotropic diffusion of chromatic and achromatic channels', *IEE Proceedings - Vision, Image and Signal Processing* **148**(3), 141–150.

[24] Mikolajczyk, K. and Schmid, C. [2004], 'Scale & affine invariant interest point detectors', *International Journal of Computer Vision* **60**(1), 63–86.
URL: *http://dx.doi.org/10.1023/B:VISI.0000027790.02288.f2*

[25] Minetto, R., Thome, N., Cord, M., Leite, N. J. and Stolfi, J. [2014], 'Snoopertext: A text detection system for automatic indexing of urban scenes', *Computer Vision and Image Understanding* **122**(Supplement C), 92 – 104.
URL: *http://www.sciencedirect.com/science/article/pii/S1077314213001914*

[26] R., B. J., Daniel, M. M. and Singer, A. C. [2002], *Computer Explorations in Signals and Systems Using MATLAB*, 2nd edition edn, Upper Saddle River and NJ: Prentice Hall.

[27] Salton, G. and McGill, M. J. [1986], *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., New York, NY, USA.

[28] Sharma, G., Jurie, F. and Schmid, C. [2012], Discriminative spatial saliency for image classification, *in* '2012 IEEE Conference on Computer Vision and Pattern Recognition', pp. 3506–3513.

[29] Sriman, B. and Schomaker, L. [2015*a*], Explicit foreground and background modeling in the classification of text blocks in scene images, *in* '2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)', pp. 755–759.

[30] Sriman, B. and Schomaker, L. [2015*b*], *Object Attention Patches for Text Detection and Recognition in Scene Images using SIFT*, Vol. 1, SciTePress, pp. 304–311.

[31] Tetsu, M. and Takio, K. [2010], *Image Classification Using Probability Higher-Order Local Auto-Correlations*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 384–394.
URL: *http://dx.doi.org/10.1007/978-3-642-12297-2_37*

[32] Torp, H., Kristoffersen, K. and Angelsen, B. A. J. [1994], 'Autocorrelation techniques in color flow imaging: signal model and statistical properties of the autocorrelation estimates', *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* **41**(5), 604–612.

[33] Troost, J. M. and De Weert, C. M. M. [1991], 'Naming versus matching in color constancy', *Perception & Psychophysics* **50**(6), 591–602.
URL: *http://dx.doi.org/10.3758/BF03207545*

[34] von Goethe, J. W. [1982], *Theory of Colours*, Cambridge, MA: MIT Press, Cambridge.

[35] Wang, W., Luo, D. and Li, W. [2009], 'Algorithm for automatic image registration on harris-laplace features', *Journal of Applied Remote Sensing* **3**(1), 033554–033554–13.
URL: *http://dx.doi.org/10.1117/1.3256135*

[36] Wang Zhang, T. S. [2016], 'Reference beam pattern design for frequency invariant beamforming based on fast fourier transform', *Sensors (Basel, Switzerland)* **16**(10).

[37] Weixing, W., Ting, C., Sheng, L. and Enmei, T. [2015], 'Remote sensing image automatic registration on multi-scale harris-laplacian', *Journal of the Indian Society of Remote Sensing* **43**(3), 501–511.
URL: *http://dx.doi.org/10.1007/s12524-014-0432-2*

[38] Wu, J.-C., Hsieh, J.-W. and Chen, Y.-S. [2008], 'Morphology-based text line extraction', *Mach. Vision Appl.* **19**(3), 195–207.
URL: *http://dx.doi.org/10.1007/s00138-007-0092-0*

[39] Wu, R. J. F. Q. L. X. G. [2006], Detecting and segmenting text from natural scenes with 2-stage classification, *in* 'Sixth International Conference on Intelligent Systems Design and Applications', Vol. 2, IEEE, pp. 819–824.

[40] Xu, J., Shivakumara, P., Lu, T., Phan, T. Q. and Tan, C. L. [2014], Graphics and scene text classification in video, *in* '2014 22nd International Conference on Pattern Recognition', pp. 4714–4719.

[41] Y.Ohta, T. and T.Sakai [1980], 'Color information for region segmentation', *ComputerGraphics and Image Processing* **13**(3), 222–241.

[42] Zanibbi, S. Z. R. [2016], A text detection system for natural scenes with convolutional feature learning and cascaded classification, *in* 'IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 625–632.

[43] Zhang, J., Chen, Q., Bai, X., Sun, Q., Sun, H. and Xia, D. [2009], An advanced harris-laplace feature detector with high repeatability, *in* 'Image and Signal Processing, 2009. CISP '09. 2nd International Congress on', pp. 1–5.

[44] Zhu, A., Wang, G. and Dong, Y. [2015], 'Detecting natural scenes text via auto image partition, two-stage grouping and two-layer classification', *Pattern Recognition Letters* **67**(Part 2), 153 – 162. Granular Mining and Knowledge Discovery.
URL: *http://www.sciencedirect.com/science/article/pii/S0167865515001713*

ACCEPTED MANUSCRIPT