IEICE TRANS. INF. & SYST., VOL.E97–D, NO.7 JULY 2014

LETTER

Scene Text Character Recognition Using Spatiality Embedded Dictionary

Song GAO[†], ***Student Member***, **Chunheng WANG**[†a)], ***Member***, **Baihua XIAO**[†], **Cunzhao SHI**[†], **Wen ZHOU**[†], ***and*** **Zhong ZHANG**[†], ***Nonmembers***

SUMMARY This paper tries to model spatial layout beyond the traditional spatial pyramid (SP) in the coding/pooling scheme for scene text character recognition. Specifically, we propose a novel method to build a dictionary called spatiality embedded dictionary (SED) in which each codeword represents a particular character stroke and is associated with a local response region. The promising results outperform other state-of-the-art algorithms.

key words: *scene text character recognition, coding and pooling, spatial pyramid, spatiality embedded dictionary*

1. Introduction

A robust scene-text-extraction system can be used in lots of areas such as image retrieval, intelligent transportation, robot vision and so on. To obtain text information from scene images, two stages are usually included: text detection and text recognition. In the past years, many efficient systems have been proposed by researchers to detect scene texts while scene text recognition has not been fully studied. In this paper, we focus on the scene text recognition stage.

Most scene text recognition techniques could be divided into two categories: Optical Character Recognition (OCR) based methods and object recognition based methods. OCR based methods [1] rely on the off-the-shelf OCR technique, which has been highly developed in the past decades, and focus on scene text binarization. However, traditional OCR techniques are designed for scanned documents which are usually easy to binarize. Scene text binarization is difficult due to complex backgrounds, different lighting conditions and heavy occlusions. Thus, object recognition based methods [2], [3] skip the binarization stage and each kind of scene character is regarded as a special object. Even though those methods [2], [3] usually combine scene text character recognition with some language models and report results of whole-word recognition, we argue that single scene text character recognition always plays a significant role. Thus, we focus on single character recognition in this paper. In particular, we adopt an object recognition based method for its simplicity and robustness, and a popular coding/pooling scheme is introduced for scene character recognition.

Manuscript received January 15, 2014.
Manuscript revised March 12, 2014.
[†]The authors are with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China.
a) E-mail: chunheng.wang@ia.ac.cn
DOI: 10.1587/transinf.E97.D.1942

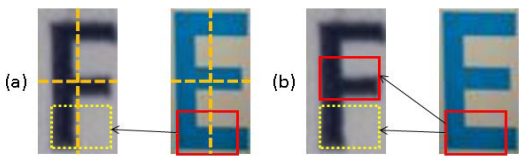

Fig. 1 Motivation: (a) the discriminative stroke to tell 'E' from 'F' is separated into less discriminative parts when using SP; (b) this discriminative stroke appears in another position of 'F', which may bring about recognition confusion if spatial information is ignored. Similar conditions may appear between 'L' and 'E', 'X' and 'Y', 'V' and 'W'.

The coding/pooling pipeline has been quite successful for object recognition in recent years. Various coding methods have been proposed, including nearest neighbor vector quantization, soft assignment [4], localized soft coding [5] and sparse coding [6]. As for pooling, average pooling and max pooling are usually used. When incorporating spatial information into coding/pooling scheme, spatial pyramid (SP) [7] has been the predominant approaches, which usually partitions one image into a set of regions beforehand and then describes them independently before concatenating code vectors. As for scene text character recognition, sizes of most character images are usually very small so image regions partitioned by SP may not be able to provide more information for character classification. Besides, rough regions division in SP may lose the power of discriminative strokes which are also separated as shown in Fig. 1 (a). Dropping SP and ignoring spatial information as in [8] can bring about character classification confusion. That's because one part of a character may appear in another location of another character as in Fig. 1 (b). So when using coding/pooling scheme for scene character recognition, it's necessary to find a way to incorporate spatial information beyond SP.

In this paper, we propose to build a new type of dictionary called spatiality embedded dictionary (SED) to include more precise spatial information than SP for scene character recognition. In SED, each codeword represents a particular character stroke and is associated with a local response region. Based on SED, localized soft coding can be performed more fast and effectively. We try to give out theoretical analysis to explain the superiority of SED over SP. The proposed mechanism has achieved 82.0% on ICDAR2003 scene text character recognition dataset and 67.1% on CHARS74K dataset which outperform other state-of-the-art methods.

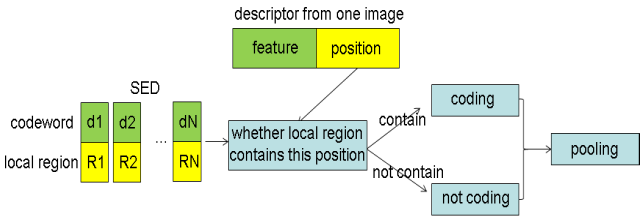

Fig. 2 How SED incorporates spatial information. Whether to code one entry for one descriptor depends on if response region of corresponding codeword covers the descriptor's position.

This paper is organized as follows. An overview of SED is given in Sect. 2. Details of the proposed method is presented in Sect. 3. Then Sect. 4 explains why SED is superior over SP theoretically. Afterwards, experiment results are given in Sect. 5. Finally, conclusions are drawn in Sect. 6.

2. Overview of SED

Instead of using K-means to cluster all descriptors regardless of their positions as before, SED performs codeword collection considering descriptors' positions. In SED, spatial information is incorporated into dictionary directly by reserving a local response region for every codeword. Thus, SED can include more precise spatial information than SP which partitions images into local regions and codes them sequentially. Based on SED, coding can be performed locally spatially depending upon the spatial relationship between codeword and descriptor as shown in Fig. 2. That will alleviate computation time and retain discrimination at the same time. The procedure of how SED incorporates spatial information is given in Fig. 2.

3. Proposed Method

3.1 Building Spatiality Embedded Dictionary

The procedure is as follows:

a) all character training images are normalized to the same size $height = H, width = W$ and partitioned into $n_h * n_w$ blocks (orange dotted lines in Fig. 3);

b) for every training image, HOG [9] with size of n_{hog} dimensions are extracted within every block and connected sequentially to form a 1-D feature vector with size of $n_h * n_w * n_{hog}$ dimensions as the overall representations;

c) assuming class c_i (i $\in \{1, 2, 3, \ldots, N_c\}$, N_c is the number of categories) has n_{train,c_i} training images, K-means clustering is performed based on the overall representations to get $n_{train,c_i}/k_{c_i}$ centers (k_{c_i} is the parameter to control the number of clustering centers). We find in experiments K-means clustering based on Euclidean distance can cluster character images with similar fonts. So clustering based on overall representations enables us to consider characters of different fonts;

d) for every clustering center of class c_i, reshape the overall representation 1-D vector to 3-D matrices with three dimension sizes of n_h, n_w, n_{hog};

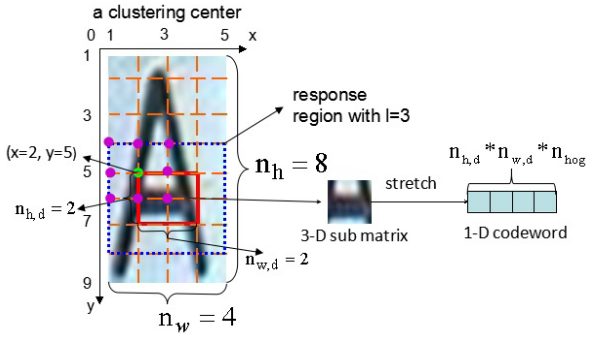

Fig. 3 Illustration of SED building: Partitioned blocks (orange dotted lines), codeword (red rectangle), sampling location (green point) and response region (blue rectangle) are shown. Extracted sub 3-D matrix is stretched to generate one 1-D codeword. We give out a visible image in this figure but clustering centers are virtual 3-D matrices actually.

e) extract sub 3-D matrices with three dimension sizes of $n_{h,d}$, $n_{w,d}$, n_{hog} from the clustering centers densely. These sub 3-D matrices are then stretched to 1-D vectors with sizes of $n_{h,d} * n_{w,d} * n_{hog}$ dimensions as shown in Fig. 3. These 1-D vectors are regarded as our codewords d_j. Extraction interval is set to be $n_{w,d}/2$ for horizontal and $n_{h,d}/2$ for vertical. The number of collected codewords from one clustering center depends on the choice of $(n_{h,d}, n_{w,d})$;

f) for every codeword d_j, we record a response region, which will enable us to code locally spatially afterwards. To illustrate the definition of response region clearly, X-Y coordinate is introduced and coordinates of top left corner (green point in Fig. 3) is regarded as coordinates of one codeword.

Then response region R_j for this codeword can be represented by the points (magenta points in Fig. 3) around the codeword's top left corner. Every point in R_j is represented as its coordinate (x, y). It should be noted that the codeword sampling position should also be included in set R_j. Actually, a response region (blue rectangle with dotted line in Fig. 3) can be regarded as the area covered by patches nearby the codewords. These patches have codeword's size and use points from R_j as their top left corners. We assume $l * l$ points are contained in R_j (l is the length of response points square as illustrated in Fig. 3.);

g) combine all codewords and their corresponding response regions into the spatiality embedded dictionary $D = \{(d_1, R_1), (d_2, R_2), (d_3, R_3), \ldots, (d_{N_D}, R_{N_D})\}$, in which N_D is the dictionary size. It should be noted that every codeword d_j corresponds to a response region R_j.

If we have totally N_{train} in the training set, we can get N_{one} codewords from one clustering center, and k_{c_i} is set to be k uniformly for all c_i, then the number of codewords in the final spatiality embedded dictionary is $N_D = N_{train} * N_{one} * (1/k)$. The procedure of generating one codeword is given in Fig. 3.

3.2 Coding and Pooling

Based on SED, coding can be performed locally spatially according to reserved codewords' response regions, which alleviates computation burden and retains discrimination power at the same time.

To code one descriptor ϕ sampled from position (x_ϕ, y_ϕ) ($x_\phi \in [1, n_w - n_{w,d} + 1]$,$y_\phi \in [1, n_h - n_{h,d} + 1]$) of one image, only entries, whose corresponding codewords' response regions R_j contain point (x_ϕ, y_ϕ), should be coded while the other entries are set to be zeros directly. It should be noted that descriptor ϕ has the same size as codeword d_j, namely $n_{h,d} * n_{w,d} * n_{hog}$ dimensions. The difference is that codeword d_j belongs to the dictionary while descriptor ϕ refers to the extracted feature. Assuming the accountable codewords set is $D_\phi = \{d_{\phi,1}, d_{\phi,2}, d_{\phi,3}, \ldots, d_{\phi,n_\phi}\} \subset D$ for descriptor ϕ, localized soft assignment [5] is selected for its efficiency and modified as below:

$$u_j = \begin{cases} \dfrac{\exp(-\beta \left\|\phi - d_j\right\|^2)}{\sum_{a=1}^{K} \exp(-\beta \left\|\phi - d_a\right\|^2)}, d_j \in D_\phi \text{ and } d_a \in NN_{(K)}(\phi) \\ 0, otherwise \end{cases} \tag{1}$$

$NN_{(K)}(\phi)$ is the K nearest neighbors of $D_\phi$in Euclidean space for descriptor ϕ. K is set to be 5 and β is set to be 0.1 in the experiments.

After coding, max pooling is performed on the whole image to obtain code vector U. Max pooling is chosen rather than average pooling because of the implied physical meaning of coding. For every descriptor, each entry u_j represents the possibility of the codeword d_j appearing in the descriptor's position (also around codeword d_j original collecting position). Besides, every codeword d_j actually corresponds to a particular character stroke located in a special location. Prior knowledge is that one stroke structure always appears only once around one position of a character image so it's reasonable to consider only the most likely appearing location.

3.3 Classifier Training and Testing

Scene text character recognition is a multi-class classification problem. For simplicity and efficiency, the resulting coding vectors $U = (u_1, u_2, u_3, \ldots, u_{N_D})$ are directly feed into linear SVM [10] for training. Regularization parameter is set to the best by cross-validation on the training set.

In the testing stage, code vectors are calculated as stated in Sect. 3.2 and labels are assigned according to the highest scores obtained from the N_c linear SVMs.

4. Theoretical Analysis

Inspired by [11], we try to give out mathematical derivation to explain the superiority of SED over SP. As soft coding is derived from hard coding, it's reasonable for us to take 1-of-k codes obtained by hard assignment as an example. Consider an image region containing P coded locations, we extract the *ith* coding entry to form a P-dimension vector $U_{iP} = (u_{i1}, u_{i2}, u_{i3}, \ldots, u_{iP})$. Then max-pooling is formalized as: $f_m = max_p U_{iP}$.

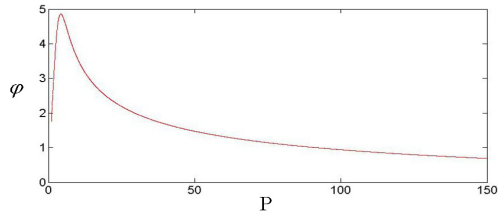

Fig. 4 Theoretical analysis with $\alpha_1 = 1.10^{-2}, \alpha_2 = 5.10^{-3}$. ϕ represents linear separability and P is the number of coded locations.

Assume i.i.d. Bernoulli variables for the *ith* coding entry with probability of α_1 and α_2 for class c_1 and c_2 respectively. As stated in [11], that assumption results in distributions f_{m1} and f_{m2} with mean $\mu_{m1} = 1 - (1 - \alpha_1)^P$ and $\mu_{m2} = 1 - (1 - \alpha_2)^P$, variance $\sigma_{m1}^2 = (1 - (1 - \alpha_1)^P)(1 - \alpha_1)^P$ and $\sigma_2^2 = (1 - (1 - \alpha_2)^P)(1 - \alpha_2)^P$. Ideally, far distance between μ_{m1} and μ_{m2} and large values of σ_{m1} and σ_{m2} bring about less overlap between the distributions of f_{m1} and f_{m2}. Less overlap means better linear separability. So the linear separability between class c_1 and c_2 can be approximated with parameter P as follows:

$$\varphi = \frac{\left|(1-\alpha_1)^P - (1-\alpha_2)^P\right|}{\sqrt{(1-(1-\alpha_1)^P)(1-\alpha_1)^P} + \sqrt{(1-(1-\alpha_2)^P)(1-\alpha_2)^P}} \tag{2}$$

As different characters contain very different parts, α_1 may be high for class C_1 while α_2 can be low for class C_2. An example of φ is given in Fig. 4 (assume $\alpha_2 \ll \alpha_1$). From Fig. 4, we can see that there is a long range in which smaller P results in better performance. That means corresponding each codeword with a local region as in SED outperforms SP which uses larger and codeword irrelevant regions.

5. Experiment

5.1 Dataset and Settings

We employ two public scene text character datasets: ICDAR2003 [12] and CHARS74K [13]. Both of these two datasets contain 62 character classes, namely digits 0-9, upper English letters A-Z and lower English letters a-z. ICDAR2003 dataset contains 6185 training patches and 5430 testing patches cropped from different scene images while CHARS74K has totally 12503 images not split into training and testing dataset. When performing CHARS74K-15 evaluation, we split training and testing set as in [8].

All of the image patches are normalized to $W = 32, H = 64$ and partitioned into blocks with $n_w = 8, n_h = 16$. HOG [9] features are extracted within every block with bin number 9, cell size of 2 pixels and normalization block size of 2*2 cells.

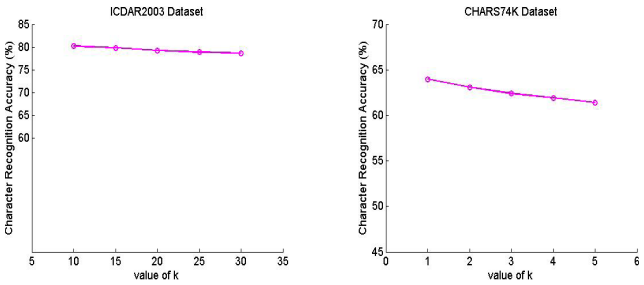

Fig. 5 Illustration of how k affect recognition accuracy. $(n_{h,d}, n_{w,d})$ is set to be (7,7) and l is set to be 3 for both datasets. Left figure is for ICDAR2003 dataset and right figure is for CHARS74K.

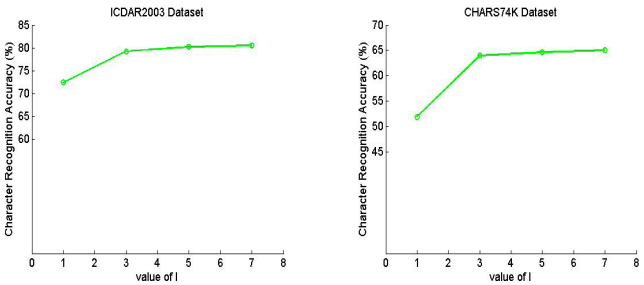

Fig. 6 Illustration of how l affect recognition accuracy. $(n_{h,d}, n_{w,d})$ is set to be (7,7). k is set to be 20 and 1 for ICDAR2003 and CHARS74K datasets respectively. Left figure is for ICDAR2003 dataset and right figure is for CHARS74K.

5.2 Discussions of Parameter k and l

Larger k can generate bigger dictionary which may be beneficial for classification, but at the same time it may result in heavier computation burden. According to our experiments, when k rises up to a peak point for both datasets, the classification accuracy seems to be still. Illustration of how k affect recognition accuracy is shown in Fig. 5. To balance classification performance and computation time, k is set to be 20 for ICDAR2003 dataset and 1 for CHARS74K dataset respectively. It should be noted that k is set to be 1 for CHARS74K dataset, which means no clustering is applied. That's because when performing CHARS75K-15 evaluation, training samples are very limited (only 15 images for each character category). Besides, CHARS74K dataset has large font variations and is more difficult than ICDAR2003 dataset. Thus, in order to achieve good classification performance, k is set to be 1 and a big dictionary is built.

For parameter l, it is ideal to set different l for different codewords as position range of strokes may be various intuitively. However, it's difficult and labor-intensive to identify different l for different codewords. Illustration of how l affect recognition accuracy is given in Fig. 6. In the experiments, l is set to be 3 for both datasets directly and empirically.

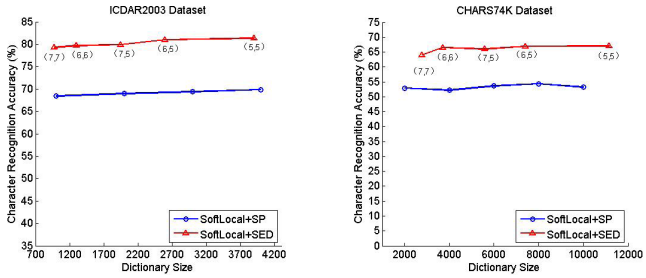

Fig. 7 Spatiality embedded dictionary (SED) vs spatial pyramid (SP): left figure is for ICDAR2003 dataset and right figure is for CHARS74K. For SED, corresponding codewords size $(n_{h,d}, n_{w,d})$ = (7,7), (6,6), (7,5), (6,5) and (5,5) are labeled in the figures. For SP, identical size $(n_{h,d}, n_{w,d})$ = (2, 2) is used. Localized soft coding and max pooling are chosen.

Table 1 Character recognition results on ICDAR2003 and CHARS74K dataset (%).

Algorithm	ICDAR2003	CHARS74K-15
HOG+NN [2]	51.5	58
SYNTH+FERNS [2]	52	47
NATIVE+FERNS [2]	64	54
MSER [14]	67	-
Global HOG [8]	76	62
Geometrical blur+SVM [13]	-	53
Multiple Kernel Learning [13]	-	55
HOG Columns [15]	-	66.5
Our method (SED)	**82.0**	**67.1**

5.3 Comparison with Spatial Pyramid

When building spatiality embedded dictionary, codewords of different sizes $(n_{h,d}, n_{w,d})$ = (7,7), (6,6), (7,5), (6,5) and (5,5) are chosen to build various sizes of dictionaries both for ICDAR2003 and CHARS74K dataset. As for spatial pyramid, codewords with identical size of $(n_{h,d}, n_{w,d})$ = (2,2) are collected using K-means clustering and popular 3-level is used. Limited to the partitioned patches by SP (when one image is partitioned into 4*4 regions), we are not allowed to choose bigger codewords for SP.

Results are shown in Fig. 7. We can see that SED outperforms SP for both datasets no matter what dictionary size is chosen. Based on SED, coding can be performed locally spatially, which alleviates computation burden greatly compared to SP. Besides, pooling based on SED is performed on the whole image region while SP performs pooling in local regions before concatenating code vectors. So code vectors of SED for one image is smaller than SP, which can reduce running time during testing. Actually, when recognizing samples from ICDAR2003 testing dataset, SED only takes about 0.15 seconds on average to classify a character image on PC with Intel (R) Core (TM) i5-3210M CPU 2.50 GHZ when $(n_{h,d}, n_{w,d})$ is set to be (7,7) (N_D is equal to 795). SP takes around 2 seconds with the same dictionary size.

For both datasets, if codewords of different sizes are combined to consider strokes of different sizes, results of SED are compared with the latest published algorithms as shown in Table 1. For fair comparison on ICDAR2003

Fig. 8 Correctly recognized scene characters from ICDAR2003 testing set. It can be seen that our method is able to classify characters with various fonts, different illumination conditions and complex backgrounds.

dataset, we only list the methods which only use samples from its own training set to train classifiers. It can be seen that, our method outperforms state-of-the-art algorithms. Some correctly recognized scene characters are given out in Fig. 8, from which we can see that the proposed method is able to classify characters with various fonts, different illumination conditions and complex backgrounds. As stated in step (c) of Sect. 3.1, for one character category, we extract codewords from different clustering centers (based on overall representations) which correspond to different fonts. Thus, our SED can recognize characters of different fonts as shown in Fig. 8.

Compared to [8] which uses HOG and non-linear SVMs, we use more simpler linear SVMs and obtain inspiring 6 percent improvement on ICDAR2003 testing dataset, which demonstrates the representation power of SED. However, performance of our system on CHARS74K dataset is still not satisfying perhaps because of the large fonts variations.

6. Conclusion

In this paper, we propose a new type of dictionary called spatiality embedded dictionary to model more precise spatial information than SP for scene character recognition. Based on SED, coding can be performed more quickly. Scene character recognition results of the proposed method outperform state-of-the-art algorithms.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No.61172103, No.60933010, State 863 Project under Grant No. 2012AA041312.

References

[1] X. Chen and A.L. Yuille, "Detecting and reading text in natural scenes," CVPR, pp.366–373, 2004.
[2] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," ICCV, pp.1457–1464, 2011.
[3] A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," CVPR, pp.2687–2694, 2012.
[4] J.C. van Gemert, C.J. Veenman, A.W. Smeulders, and J.M. Geusebroek, "Visual word ambiguity," IEEE Trans. Pattern Anal. Mach. Intell., vol.32, no.7, pp.1271–1283, 2010.
[5] L. Liu, L. Wang, and X. Liu, "In defense of soft-assignment coding," ICCV, pp.2486–2493, 2011.
[6] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," CVPR, pp.1794–1801, 2009.
[7] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," CVPR, pp.2169–2178, 2006.
[8] C. Yi, X. Yang, and Y. Tian, "Feature representation for scene text character recognition: A camparative study," ICDAR, pp.907–911, 2013.
[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," CVPR, pp.886–893, 2005.
[10] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, "Liblinear: A library for large linear classification," J. Machine Learning Research, vol.9, pp.1871–1874, 2008.
[11] Y.L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," ICML, pp.111–118, 2010.
[12] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," ICDAR, pp.682–687, 2003.
[13] T.E.D. Campos, B.R. Babu, and M. Varma, "Character recognition in natural images," Computer Vision Theory and Applications, pp.273–280, 2009.
[14] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," ACCV, pp.770–783, 2010.
[15] A.J. Newell and L.D. Griffin, "Multiscale histogram of oriented gradient descriptors for robust character recognition," ICDAR, pp.1085–1089, 2011.