Collaboration: The workload is divided almost equally while each task has someone in charge.
Yu Cao 05369675
Tao Wang 05365513
Mar 10, 2011, 1 late day

CS224N Project: Automatic Author Name Transliteration via OCR and NLP

1 Introduction

When a non-English speaker is browsing through the catalog of English books, most English words in the title or selected reviews can be translated and understood by the reader, since they carry semantic meaning in modern English. However it will be more difficult for the reader to know the author(s), since many names are not in the dictionary and the reader does not know how to pronounce them. It will be very helpful if the names in English can be transliterated to the reader's language so that one knows how to pronounce the names of the author(s).

Given an image of the cover page of an English book, our project aims to perform the following tasks:

1) Optical character recognition (OCR)
2) Distinguish the name(s) of the author(s) from other texts
3) Transliterate the English names into Chinese characters

Firstly, natural scene text recognition is a largely unsolved problem and is receiving growing attention nowadays. It can be split into two stages: automatic text detection and text recognition. While text detection is slightly easier, the state-of-the-art of text recognition is lingering around 60-70% in terms of accuracy. We hope to improve the Latin character recognition accuracy with the knowledge of both image processing and an English language model. Our focus is on the OCR part, assuming text blocks in the natural scene have already been detected. Our first step in OCR is to build a multi-class character classifier based on histogram of oriented gradients (HOG) and a support vector machine (SVM) with linear kernel. The overall character-level accuracy of the classifier is 74.4%. The probabilistic prediction distribution of the input character image is then fed into a Bayesian Network (BN) for further refinement based on a character-level bigram language model. This increases the result to 75.3%.

In our specific application with an image of the book cover, we try to distinguish authors from titles, publishers, reviews, quotes etc. Then we will transliterate the English names to Chinese characters to facilitate the pronunciation and/or recognition by Chinese readers. Author name recognition can be considered as a special case of named entity recognition (NER), where there are only two types of labels, "PERSON" and "NONPERSON". We built a maximum entropy Markov model, together with some feature engineering to choose a good set of features for "PERSON" recognition. Our model achieved a F1 score of 77.5% (Precision 76.9% and Recall 78.1%).

Since English is phonographic, it is relatively easier to transliterate from English to similar phonographic languages such as French. Readers who can speak French may also find it easier to pronounce English names. One the other hand, transliterating to Chinese makes it more indirect since Chinese is ideogram. And once it can be achieved, it is also more helpful to readers who only speak Chinese. We consider the

transliteration task as a special case of machine translation. In this case, one or more English letters, instead of words, need to be mapped to one Chinese character. For an evaluation set of 120 English names, the number of acceptable transliterated Chinese names is about 100 by human inspection, which is around 83%.

2 Prior Work

The problem of scanned documents OCR has been well studied, and near human accuracies can be obtained via accurate image binarization [3]. On the contrary, OCR in natural scene images is a relatively new area. The challenge lays in the blur, skew, complicated background clutter and adverse lighting conditions present in natural scene images, which make it difficult to be binarized and segmented. Several existing works on natural scene images used feature-based learning for character recognition. In [2], experiments with different features show that Geometric Blur [1] outperforms other features such as Scale Invariant Feature Transform [9] and Spin Image [7]. Apart from that, HOG has been proven effective in generic object (such as pedestrians) detection in natural images [4] [10]. The performance of HOG-based features in OCR-related tasks has been explored in a few works [13] [16], but mainly on spotting given words in images. Our project extends the above works to perform character classification and generic word recognition in natural scene images.

To our best knowledge, there are in general two approaches to the English – Chinese transliteration problem. One way is to start from an English word, convert it to its phonemic representation, then translate the English phoneme sequence into an equivalent Chinese Pinyin sequence, and finally map the Pinyin symbols to Chinese (Han) characters [14] [15]. This approach usually involves the use of English phoneme – Chinese Pinyin mapping table and Pinyin – Han character mapping table. Another approach is to extract transliterated pairs from parallel corpora, such as in [8]. This approach has much better performance. However, it is based on the fact that the corresponding transliterated Chinese characters always exist in the parallel text. Once they are located, it is impossible to get the characters or phrase wrong. It is not exportable for use in open-ended transliteration.

3 Our Approach

3.1 OCR

To solve the optical character recognition problem, a multi-class linear SVM classifier is built based on HOG features. The classifier outputs a score to each possible label (i.e. 52 capital and small letters) given the input image patch, which is a character embedded in its natural scene background. The scores for each image patch is normalized into a proper probability distribution over the 52 classes, and subsequently combined with a character-level bigram language model using BN to choose the best prediction output.

The character classifier model is trained and tested on the ICDAR 2011 dataset, which comprises of labeled patches with single characters. We chose the ICDAR datasets because they are taken in natural scenes, and many of them are images of book covers. Moreover, the ICDAR dataset provides information on the position of each character, so that we can focus on the task of character recognition without worrying about the detection stage. The draw back of the ICDAR datasets is that it contains relatively few samples, and certain character class such as lower case ‘z’ does not have any instance in the training samples. To address this issue, we generate more data by distorting the ICDAR training samples.

We extract OCR features by running an 8×8 window densely over the input image, computing HOG features in each window by collecting votes into 9 orientation bins, and concatenating them into a higher dimensional vector. It is a standard approach to normalize the HOG feature vectors from each 8×8 window before concatenating them [10]. However, by doing this in our case, we will lose contrast information across different windows, which is essential for the classifier to distinguish between characters. Therefore, we normalize the entire feature vector after concatenation. The feature vectors obtained from all the training images, together with their true labels, are used to train L1-regularized linear SVM classifier with 5-fold cross validation [5].

The character-level bigram English model is obtained from the EUROPARL dataset with 1,140,473 lower-cased sentences. Since there are 52 classes, we need to compute frequencies for three types of bigrams, i.e. capital letter followed by small letter, capital letter followed by capital letter and small letter followed by small letter. The case of capital letter followed by small letter is counted by considering the first letter of each word as capital. We chose to capitalize the first letter of each word instead of sentence case style because it is unlikely for text on book covers to follow sentence case style. The bigram model is then smoothed using Laplace-smoothing by adding one to each count. Laplace-smoothing does not introduce problems as for word-level bigram models because there are only a total of 52 classes.

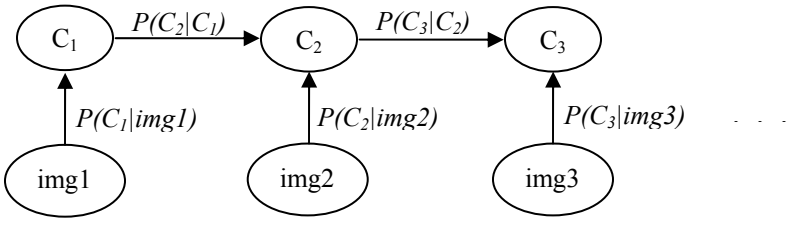
We improve the character recognition accuracy further by taking character transition probabilities into account. For example, we can better distinguish between "h" and "b" if we know that the previous letter is "t", because "th" is much more likely to appear than "tb" in English. We consider it as a BN as shown. The transition probability $P(C_i|C_{i-1})$ is obtained from the language model, where C is the actual character. The image probability $P(C_i|img_i)$ is obtained from the HOG classifier, where *img* is the input image patch. The probability of the word can be computed as follows:

$$P(C_1, C_2, C_3) \propto P(C_1 \mid img_1) P(C_2 \mid C_1) P(C_2 \mid img_2) P(C_3 \mid C_2) P(C_3 \mid img_3)$$

We need to compute the character sequence with highest likelihood, that is:

$$C = \arg\max_{C_1, C_2, \ldots C_l} P(C_1 \mid img_1) \prod_{i=2}^{l} P(C_i \mid C_{i-1}) P(C_i \mid img_i)$$

However, this would require searching a 52^l space, which is exponential in the length of the character sequence and thus intractable. As an alternative, we used a greedy algorithm which predicts the most likely character $\arg\max_{C_i} P(C_i \mid C_{i-1})^{\lambda} P(C_i \mid img_i)$ one at a time iteratively, where λ is the language model weight between 0 and 1. This approach does not guarantee to find the sequence with the highest probability, but it is very efficient and does give good results in many cases.



3.2 NER

After recognizing the text, we will identify the text indicating authors. This task can be considered as a binary NER task with essentially two types of labels, "PERSON" and "NONPERSON". A maximum entropy Markov model is used to accomplish the task.

The training data for the maximum entropy classifier is the Message Understanding Conference (MUC) 7 corpora (http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T02), which is a newswire text with entity annotations. We kept the "PERSON" and "ORGANIZATION" labels while replacing all the other labels as "OTHER". "ORGANIZATION" is kept because it usually contains names, e.g. "Kennedy Space Center". A similar situation in book cover pages will be publishers such as "HarperCollins Publishers". We need the classifier to be able to distinguish the names of authors from the names of publishers.

After some trials, the set of features includes "CURRENT WORD", "PREVIOUS LABEL", "MIDDLE INITIAL", "IN DICTIONARY", "IN NAME DATABASE", and "NEXT WORD". "PREVIOUS LABEL" and "NEXT WORD" allows the classifier to come up with decisions in the context. "MIDDLE INITIAL", "IN DICTIONARY" and "IN NAME DATABASE" are all Boolean features. "MIDDLE INITIAL" evaluates whether the current word resembles the format of a middle initial. "IN DICTIONARY" assesses whether the current word is in a dictionary of 127,195 common English words. "IN NAME DATABASE" checks whether the current word is among the common English names, based on the statistics by U.S. Census Bureau (http://www.census.gov/genealogy/www/data/1990surnames/names_files.html). Features such as whether a word begins with a capital letter or the entire word is capitalized are not included because of the special situation in book cover design, where most words do not follow the case style in sentences. A Viterbi decoder is then used to find the best possible sequence of labels, instead of greedily finding the best label at each point.

3.3 Transliteration

Most transliteration models make use of phonetic mapping rules. They tend to first convert the English name into a phoneme sequence, and then map it to Chinese Pinyin symbols, and finally choose Chinese characters corresponding to the Pinyin symbols. Our approach uses character-level translation models to directly map one or more successive English letters into a Chinese character. This may be considered analogous to phrase-based translation.

The training data are English – Chinese name pairs obtained from online tools such as http://www.chinese-tools.com/names/list.html. We filtered out some European names with accents. Some modifications on the transliterated Chinese names are also made since the data from these online sources are not completely correct. The final data set contains 4,256 pairs of transliterated names. We have also prepared a separate set of 846 pairs for tuning of the parameters, and a set of 120 English names obtained by NER for testing.

The transliteration model is mainly built on the software package Moses, a statistical machine translation system that allows automatic training of translation models for any language pair. Firstly, a trigram Chinese language model is built with the SRI language modeling toolkit. Then an alignment model is trained with the parallel text using the GIZA++ toolkit. Finally a decoder Moses is tuned to provide the translation model. The main difference from the usual translation model is that tokens are now individual English letters instead of words. Since the transliterated Chinese names are usually shorter than the original English

names, one or more English letters need to be mapped to a Chinese character. According to the difference in pronunciations when letters are combined in difference ways, they should be mapped to different characters. For example, the letter sequence "ri" in "Price" should be mapped to "赖", while the same combination in "Princeton" should be mapped to "林". This is analogous to words being interpreted differently in different contexts.

In this application, the transliteration problem can be defined as follows:

$\hat{c} = \arg\max_{c} P(e \mid c) P(c)$, where c is the Chinese character sequence, e is the English letter sequence.

The $P(c)$ distribution depends on the Chinese language model. $P(e \mid c)$ can be obtained from the alignment model by $P(e \mid c) = \sum_{a} P(e, a \mid c)$, where a is the alignment from e to c.

The common alignment models are IBM models 1-5 and HMM model. We have experimented with different combinations of these models. There are four parameters to be tuned in the translation model, distortion (reordering) weight, language model weight, phrase translation weight, and word penalty. These parameters can be tuned with the separate development data set, or with hand-set values. In our specific application, the characters of the transliterated Chinese name always follow the same order as the letters in the original English name. Therefore it is reasonable to set the distortion weight to be 1, which implies little distortion. In addition, the transliterated Chinese name is usually shorter in terms of characters. So a word penalty of 0 – 3 is expected to restrict the length of the transliterated name.

4 Results

4.1 OCR

Our classifier has 52 character classes, including upper and lower case English letters. We achieved character-level recognition accuracy of 72% by applying overlapping 8×8 HOG windows at a step of 4 pixels across the given image patch. We experimented with a smaller step of 2 and achieved 74.4% accuracy, which outperforms experiments using other features in [2]. This result also suggests that running HOG windows densely can improve classification results.

When combining the HOG classifier output with the language model via BN, the recognition accuracy improves slightly. We experimented with a range of language model weights in the final probability expression. The optimal range seems to be in the range 0.05 - 0.1. With language model weights set to 0.1, the recognition accuracy increases to 75.3%. Apart from the character-level accuracy, we also evaluated the overall accuracy at word-level, which is the percentage of words with all their characters recognized correctly. The word-level accuracy is 47% using the HOG classifier alone, and improved to 48.83% after combining with the BN. The results confirmed that incorporating a character-level language model enhances both character-level and word-level recognition accuracies.

4.2 NER

Since the NER task can be considered as binary classification, the evaluation metric used is F1 score, or Precision and Recall. Overall accuracy is not used to evaluate the performance as the proportion of words labeled as "PERSON" is small. We have obtained results via three methods:

1) Our maximum entropy Markov model with feature engineering

2) Stanford Named Entity Recognizer with the built-in model
3) Stanford Named Entity Recognizer with self-training

Method 1 and 3 are both trained on the MUC7 data set. Method 2 has a built-in model, which is trained on a mixture of CoNLL, MUC6, MUC7 and ACE named entity corpora, and is thus fairly robust across domains. Both method 2 and 3 makes use of the Stanford Named Entity Recognizer, which is a Conditional Random Field (CRF) classifier, coupled with well-engineered feature extractors. The performance is evaluated on a test set of text extracted from 50 book cover pages, ranging from cook books to novels. The results are summarized in Table 1 below.

Table 1 Summary of results for NER

Method	Precision	Recall	F1
maximum entropy Markov model	0.7686	0.7815	0.775
Stanford NER	0.9059	0.6471	0.7549
Stanford NER with self-training	0.8125	0.2185	0.3444

We can observe that method 1 and 2 have similar F1 scores, but method 1 has a more balanced performance on precision and recall. The Stanford NER models have better precision in general, but poorer recall performance. Our model with feature engineering is specifically targeted at "PERSON" recognition and achieves better overall results for the application purpose. However we do notice some undesirable results. Firstly, some names which carry semantic meaning are less likely to be recognized. For example, "Mark" can be a common first name, but it is also a commonly used word in sentences, either as a verb or noun. Secondly, depending on the label of the previous word, the name of the publisher can still be mislabeled as "PERSON". The model may also mislabel words such as "Britain" and "Africa", which do have transliterations but are not likely to be names of authors.

4.3 Transliteration

The evaluation of the transliteration results in Chinese is more difficult due to two reasons. Firstly, many Chinese characters can have the same pronunciation. It is perfectly acceptable to use "利" instead of "历" since both characters are pronounced as "Lì". Secondly, transliteration attempts to write down what a word sounds like, which can be subjective. For instance, the name "Victoria" can be transliterated as "维多利亚" (Wéi duō lì yà) or "维克托利亚" (Wéi kè tuō lì yà). It all depends on how specific one would like to map each phoneme. Therefore, we decided to evaluate whether a transliterated name is acceptable with human judgment, instead of matching it with a golden rule. The judgment could be slightly subjective; however the rule is that the transliterated Chinese name should sound almost the same as the English name. If one important vowel phoneme is transliterated in the wrong way, the whole name is marked as unacceptable, even if only one character corresponding to the vowel is wrong. One such example would be transliterating "Price" as "普利斯" (Pǔ lì sī) instead of "普赖斯" (Pǔ lài sī).

Since the evaluation is done manually, we did not perform it on a large test set. We prepared a total of 120 words labeled as "PERSON" from the previous NER task. Firstly, we compared the results with and without asserting the distortion weight to be 1. An acceptance score of 91 /120 is obtained without imposing the heuristic that reordering is unlikely. The score improves to 102 /120 when we set the distortion weight to be 1, which restricts the possibility of distortion to be almost zero. We have also

experimented with different combinations of the alignment models. The acceptance score does not show significant variation; it is always within 100 ± 2 /120. It is difficult to choose the best combination since each one may perform better on certain names and worse on other names. The combination "m1=5, mh=5, m3=3" maps "Quant" to "奎安特" (Kuí ān tè), which is better than "夸地" (Kuā dì) from "m1=5, m3=3, m4=3". However it maps "Lidwell" to "利德厄尔" (Lì dé è ěr), which is not as good as "利德威尔" (Lì dé wēi ěr).

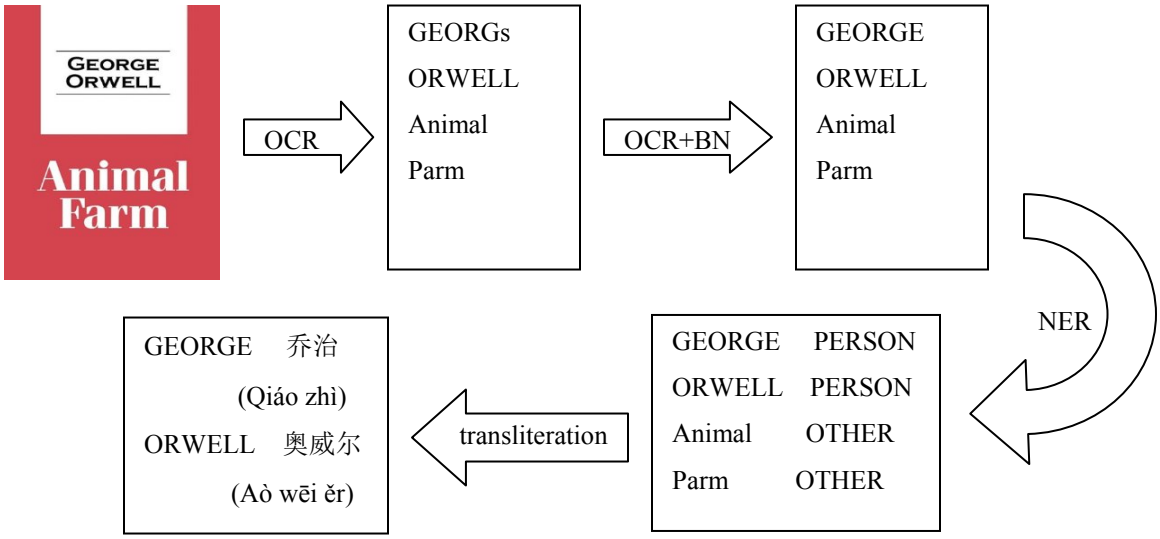
Table 2 Comparison of different combinations of alignment models

	M1=5, m3=3, m4=3	m1=5, mh=5, m3=3	m1=5, m3=5	m1=5, mh=3, m3=3, m4=3
Score /120	102	101	100	98

*mh: HMM model; m#: IBM model#; m1=5: run IBM model 1 for 5 iterations

4.4 Overall Integration

Here is one example of the actual application.



5 Discussion

In this project, we have implemented an automatic system for author name transliteration from English to Chinese. Assume the text blocks in a book cover have been detected, we perform optical character recognition, followed by named entity recognition to find out the authors, and then the names are transliterated into Chinese equivalents. The OCR classifier achieves an accuracy of 75.3% when combined with a character-level bigram language model. The NER yields a F1 score 0.775. Evaluation of transliteration model is done manually with an acceptance score of 100/120.

We explored the performance of HOG feature in OCR applications, and we improved both character-level and word-level accuracies by taking the character transition model into account using a BN. Currently we use a greedy approach to find a highly likely character sequence. Another possibly better approach would be using the Viterbi Algorithm to find the most likely character sequence. For future works, we may switch

to a conditional random field (CRF) to represent the dependencies between adjacent characters, which can be trained on a forward-backward algorithm to find the optimum character sequence efficiently.

For the NER task in this specific application, our model only makes use of the text on the book cover. Taking into consideration of a typical book cover design, we suggest incorporating other relevant information such as the relative position of the successive text blocks, or introducing color or font detection to differentiate among titles, authors, publishers etc. Another effective but more expensive way would be to construct a corpus with text that closely resembles the characteristics of the text appearing on book covers. The training corpus we used is from newswire source. Although it does contain sentences, titles and names, it is different from the text on book covers in terms of length or variety of sentences, expression patterns etc. For example, author names printed on book covers are often preceded with "by" or "author". Such highly indicative features are less likely to appear in newswire text, and therefore, less likely to be learnt by our NER model.

For the transliteration model, our approach greatly simplifies the procedures used by existing methods. However, due to the small size of the training corpus, not all English letter sequences have a fair presence. Thus it is difficult for the model to figure out some special pronunciations when the letters are grouped in a special way. For example, the model can never learn to transliterate "Washington" correctly by muting the letter 'g' if such special cases are not included in the corpus.

6 References

[1] Berg, A. C., Berg, T. L., and Malik, J. 2005. Shape Matching and Object Recognition Using Low Distortion Correspondences. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1* (CVPR '05), Vol. 1. IEEE Computer Society, Washington, DC, USA, 26-33.

[2] Campos, T. E., Babu, B. R., and Varma, M. 2009. Character Recognition in Natural Images. In *VISAPP*, 05-08 February 2009, Lisbon, Portugal.

[3] Casey, R. G., and Lecolinet, E. 1996. A Survey of Methods and Strategies in Character Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 7 (July 1996), 690-706.

[4] 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 - Volume 1* (CVPR '05), Vol. 1. IEEE Computer Society, Washington, DC, USA, 886-893.

[5] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. and Lin, C.-J. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9 (2008), 1871-1874.

[6] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., j Bojar, O., Constantin, A., and Herbst, E. 2007. Moses: Open Source Toolkit for Statistical Machine Translation, Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic.

[7] Lazebnik, S., Schmid, C., and Ponce, J. 2005. A Sparse Texture Representation Using Local Affine Regions. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 8 (August 2005), 1265-1278.

[8] Lee, C., Chang, J. S., and Jang, J. R. 2006. Extraction of transliteration pairs from parallel corpora using a statistical transliteration model. *Inf. Sci.* 176, 1 (January 2006), 67-90.

[9] Lowe, D. G. 1999. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision - Volume 2* (ICCV '99), Vol. 2. IEEE Computer Society, Washington, DC, USA, 1150-.

[10] Ludwig, O., Delgado, D., Goncalves, V. and Nunes, U. 2009. Trainable Classifier-Fusion Schemes: An Application to Pedestrian Detection. In *Proceedings of the 12th International IEEE Conference On Intelligent Transportation Systems* (ITSC '09), Vol. 1. St. Louis, 432-437.

[11] Och, F. J., Ney, H. 2003. A Systematic Comparison of Various Statistical Alignment Models, *Computational Linguistics*, 29, 1 (March 2003), 19-51.

[12] Stolcke, A. 2002. SRILM -- An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. on Spoken Language Processing - Volume 2*, Denver, 901-904.

[13] Terasawa, K., and Tanaka, Y. 2009. Slit Style HOG Feature for Document Image Word Spotting. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition* (ICDAR '09). IEEE Computer Society, Washington, DC, USA, 116-120.

[14] Virga, P., and Khudanpur, S. 2003. Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL 2003 workshop on Multilingual and mixed-language named entity recognition - Volume 15* (MultiNER '03), Vol. 15. Association for Computational Linguistics, Stroudsburg, PA, USA, 57-64.

[15] Wan, S., and Verspoor, C. M. 1998. Automatic English-Chinese name transliteration for development of multilingual resources. In *Proceedings of the 17th international conference on Computational linguistics - Volume 2* (COLING '98), Vol. 2. Association for Computational Linguistics, Stroudsburg, PA, USA, 1352-1356.

[16] Wang, K., and Belongie, S. 2010. Word spotting in the wild. In *Proceedings of the 11th European conference on Computer vision: Part I* (ECCV'10), Kostas Daniilidis, Petros Maragos, and Nikos Paragios (Eds.). Springer-Verlag, Berlin, Heidelberg, 591-604.