Offline Arabic Handwriting Recognition Using BLSTMs Combination

Sana Khamekhem Jemni[1], Yousri Kessentini[1,2], Slim Kanoun[1] and Jean-Marc Ogier[3]
[1] University of Sfax, MIRACL Laboratory, Sfax, Tunisia
[2] Digital Research Center of Sfax, B.P. 275, Sakiet Ezzit, 3021 Sfax, Tunisia
[3] L3i Laboratory, University of La Rochelle, France
{sana.khamekhem, yousri.kessentini, slim.kanoun}@gmail.com
jean-marc.ogier@univ-lr.fr

Abstract—**We propose in this paper, an Arabic handwriting recognition system based on multiple BLSTM-CTC combination architectures. Given several feature sets, the low-level fusion consisted in projecting them into a unique feature space. Mid-level combination methods were performed using two techniques: the first one consists in averaging the a-posteriori probabilities of each individual BLSTM, and injecting them in the CTC decoding. The second is based on the training of a new BLSTM-CTC system using the sum of the a-posteriori probabilities generated by the individual systems. The high-level fusion is based on the combination of the individual decoding outputs. Lattice combination and ROVER strategies were evaluated in this context. The experiments conducted on the KHATT database showed that the high-level combination method significantly improves the recognition rate compared to the other fusion strategies.**

Keywords- BLSTM; CTC; Feature Fusion; Net Averaging; ROVER; Lattice Combination.

I. INTRODUCTION

Handwritten Arabic text recognition has been a popular research area since 1970 and is still an open issue in the pattern recognition field. The current systems performances are still deficient and more robust recognition systems are required. The recognition of offline Arabic handwritten script is challenging due to the variability between the different individual writings, the presence of touching letters and complex long-term context dependencies. Recognition systems have proven their success to recognize a limited vocabulary. However, this task remains difficult in the case of using a large vocabulary. Thus, exploring recognition in context free systems has gained the interest of several researchers. The use of handwriting recognition systems based on Recurrent Neural Networks (RNNs) has been widely studied during the past few years. Indeed, the Bi-Directional Long Short Term Memory neural network (BLSTM) is able to absorb variability and integrate contextual information. Therefore, a classifier combination could be an interesting issue to enhance the performance and lead to overcome the deficiency of one classifier [1].

In fact, various combination techniques have been explored in the literature. They can be classified into two categories: feature fusion, also known as low-level combination approach, and decision fusion, or high-level integration. The first category consists in combining the input feature representations into a single feature space, and consequently uses a classifier to model the combined observations in the unique input feature space. In [2], the authors propose a HMM based system for handwritten text line images. The conducted experiments prove the superior performance of the early integration method. The second category focuses on integrating the scores as delivered by the classifiers on various feature sets through a fixed combination rule. In [4], different combination levels were explored using a BLSTM-CTC in the case of isolated handwritten words recognition and showed that the low-level combination strategy outperforms the decision level combination. However, the reported classifiers combination works for handwritten text line recognition are scarce. The fusion of the outputs of multiple handwritten text line recognizers differs from the typical multiple classifier combination. The output of a text line recognizer is a sequence of words their number vary between various recognizers. In this context, Fiscus [5] proposed an algorithm known as Recognizer Output Voting Error Reduction (ROVER) for speech recognition, where a word sequence alignment is defined by a voting procedure coupled with optional word confidence scores. In the same way, to avoid the voting procedure, a lattice combination method, which derives the system weights from the Bayesian decision concept, is described in [8]. This method selects hypotheses for a single system with a minimum Bayes risk (MBR), then, combines the best fragments of these outputs to generate the best hypotheses. Although the popularity of multiple classifier combination in handwritten recognition has grown significantly, not much mid-level combination strategies for handwritten text line recognition are available in the literature [3].

In this context, the present study presents a comparative study of different combination levels of BLSTM-CTC recognition systems trained on different feature sets. A low-level fusion relying on feature concatenation was performed. Then, the fusion of the net outputs at a middle level was introduced. A later combination was explored by merging the outputs of each recognizer. ROVER and lattice combination were experimented for this aim. The experiments conducted on the Arabic KHATT dataset [16] have proven the advantage of cooperative systems over the individual one, specially using high-level combination.

The remaining of this paper was organized as follows. In Section II, the different components of the systems were introduced. Section III detailed the system combination

978-1-5386-3346-5/18 $31.00 © 2018 IEEE
DOI 10.1109/DAS.2018.54

levels. The results of this work were summarized in section IV before drawing our major conclusions in final section.

II. System overview

The handwritten text recognition (HTR) systems have multiple processing steps in order to convert a text image into its adequate transcription or Unicode. A segmentation stage is often required to separate words/characters of a text line image. A sliding window scans the line image from the right to the left in order to extract the features that are the input for the BLSTM networks. The width of the sliding window is set to three pixels with an overlap of two pixels. The extracted descriptors accompanied with the label sequences, which correspond to the sequence of characters in the ground-truth of the text line image, are sent to the BLSTM-CTC networks for training. Then, a CTC layer is involved in order to predict the test sequences. Thereby, the data is recognized at the line level. Figure 1 illustrates the pipeline of the proposed baseline system.

However, the segmentation free approach bypasses this step. A sliding window scans the line image from the right to the left in order to extract the features that are the input for the BLSTM networks. The extracted descriptors accompanied with the label sequences, which correspond to the sequence of characters in the ground-truth of the text line image, are sent to the BLSTM-CTC networks for training. Then, a CTC layer is involved in order to predict the test sequences. Thereby, the data is recognized at the line level. Figure 1 illustrates the pipeline of the proposed baseline system.

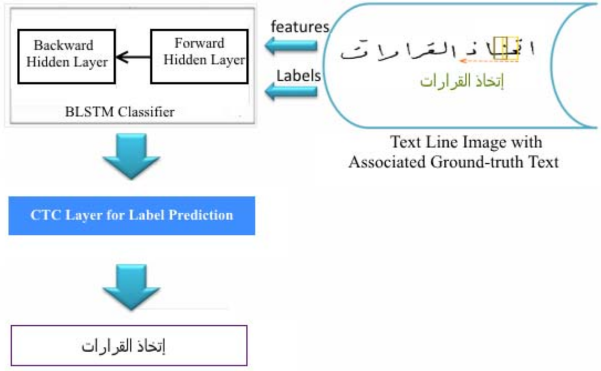

Figure 1. Pipeline of the proposed system.

A. Pre-processing and feature extraction

Handwriting in a free-style handwritten text may be slanted, skewed, or fluctuating. This is due to the style of the writer and his culture. Therefore, a preprocessing step is necessary in order to reduce the generated noise and eliminate any variability resource that occurred during the images scanning phase. The Sauvola algorithm [9] is applied to binarize the line image. A normalization step is performed to reduce the signal inclination to the horizontal. The skew angle is determined via the image contour and then it is corrected.

Arabic writing uses two basic lines: the upper baseline and the lower line. These baselines define three areas within a word: the central area, the upper area where the ascending ones can be found, and the lower area for the descendants. In order to extract baseline-dependent features, we used the method described in [24].

Two different feature extraction strategies were studied: Segment-based and Distribution-Concavity (DC) based features. Segment-based feature extraction is presented in [17]. A region of successive foreground pixels within the sliding window is referred to us as segment. An area, which corresponds to consecutive foreground pixels, within the sliding window, includes six centroid features $c_{1..6}$ plus six segment height features $h_{1..6}$, resulting in a 12-dimensional feature vector. There are n segments in a window. If n is equal to six, we set c_i to the y-coordinate of the centroid of the i-th segment, and h_i to the height of that segment. The number of segments within a window in Arabic usually does not surpass six. Figure 2 describes the segment based feature extraction process.

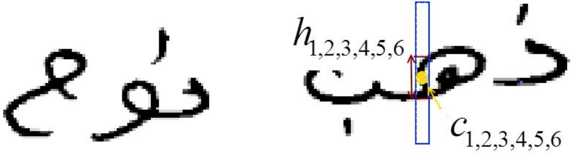

Figure 2. Segement-Based Feature Extraction.

The feature-based distribution consists of 11 features that describe the density of the foreground pixels within frames, which are extracted using the sliding window previously defined. The first feature F1 is the density of the foreground pixels within the frame. Feature F2 is the number of black/white transitions between two consecutive frames. Feature F3 is a derivative feature defined as the difference between the y-coordinate of the center of gravity of foreground pixels of two successive frames.

Features from F4 to F6 represent the densities of black (foreground) pixels for each vertical column of pixels in each frame (in our case, the width of the frame is 3 pixels). Feature F7 is the vertical distance between the lower baseline and the center of gravity of foreground pixels normalized by the height of the frame. Feature F8 (resp. F9) represents the density of the foreground pixels over (resp. under) the lower baseline.

Feature F10 is the number of transitions between two consecutive cells of different density levels above the lower baseline. Feature F11 characterizes the zone to which the gravity center of black pixels belongs, while taking account of the upper and lower baselines.

Concavity features are features that withdraw local concavity information and stroke directions within each frame. Each of the concavity features F12 to F17 describes the number of white pixels (background) that belong to one of six types of concavity configurations. These features are explored by using a 3x3 window (mask). Besides, the six additional and baseline dependent concavity features related to the core zone are added to these features.

B. BLSTM-CTC Training

The BLSTM [14], which is a variant of the recurrent neural network classifier, has been proven a success in sequence recognition. The data is recognized at the character level, thus overcoming the issues of the character segmentation problems. This architecture consists in coupling the bi-directional neural network and LSTM layers. Two hidden layers are used; one to process the input sequence forward while the other is for backward pass. Both layers outputs are combined in the next layer. To avoid the pre-segmented data requirement, a CTC (Connectionist Temporal Classification) output layer is designed, then, the probability of an output sequence label is predicted given an input sequence, as indicated in Figure 3.

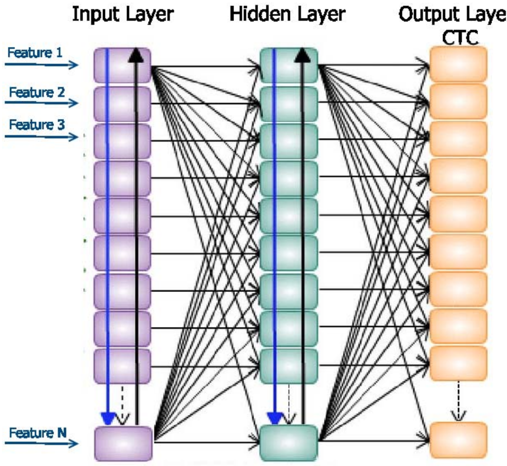

Figure 3. The BLSTM-CTC Classifier

In our experiments, the BLSTM-CTC were trained using the EESEN speech recognition toolkit [13]. The Arabic script features count 28 letters. Each letter has one to four forms. A clustering process is used to reduce the number of character classes by combining similar forms to the same class. We added punctuation symbols and numbers and ended up with a class forms set size of 108 for the KHATT corpus.

C. Decoding Stage

The CTC is a dedicated neural network layer devoted to transform BLSTM outputs into class posterior probabilities. Its aim is to train unsegmented data, such as handwriting or speech. Each output represents a character, the BLSTM signal is transformed into a sequence of characters. This layer has many characters outputs plus one additional output known as the "blank" output. Therefore, it avoids taking a decision in uncertain zones instead of continuously being trained to decide on a character in a low context region (e.g. uncertain).

To decode the CTC trained model, a weighted finite state transducer (WFST) [25] is used. Thus, we search the most probable output transcription y for a given input sequence X. During the decoding process, the lexicon drives the search within the sequences of characters that forms the lexicon elements represented by the lexicon FST (*L*). The language model give the recognition output of the most likely sequence of n-grams represented by the grammar FST (*G*).

D. Language Modeling (LM) and Lattice Rescoring

In handwritten text recognition, n-gram LMs are used to assign a probability to a sequence of words $W = \langle w_1, w_2, ..., w_N \rangle$.

$$P(W) = \prod_{i=1}^{N} P(w_i \mid h_1^{i-1}) \quad (1)$$

Where $h_1^{i-1} = \langle w_1, w_2, ..., w_i \rangle$ denotes the history context for word w_i .The probability distribution given any history context $h^{i-1}{}_1$.

Back-off n-gram LMs have been the principal form of statistical language models during the last few decades. In fact, they have a simple model structures, an efficient parameter estimation methods and discounting algorithms. A good generalization performance can be attained using back-off n-gram LMs when large amount of training data are available. The probability of the current word being predicted depends only on preceding $N-1$ words under a Markov assumption. This is given by:

$$P(w_i \mid h_1^{i-1}) = P(w_i \mid h_{i-N+1}^{i-1}) \quad (2)$$

In this work, n-gram (n=3) LMs are estimated on the training corpus of the KHATT database using the discounting method. SRILM toolkit [20] was used for this purpose.

III. COMBINATION STRATEGIES

Two BLSTM-RNNs were trained with different features. Diverse combination strategies were evaluated at various levels of the text recognizer.

A. Low-level Combination

A low-level combination was performed to measure the capacity of the BLSTM-CTC to treat height dimensional feature vectors. The achieved combination, by merging the features, required the learning of a unique BLSTM-CTC as shown in Figure 4. In our case, we concatenated features based on pixel distribution and concavity (DC) with the segment based descriptor (S).

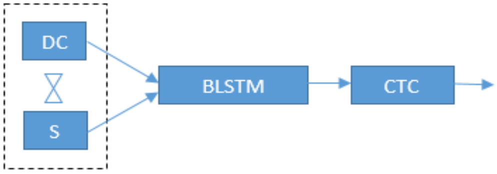

Figure 4. Low-Level Combination

B. Mid-Level Combination

The mid-level combination was applied before the CTC decoding step. The log a-posteriori probabilities of characters were extracted for both systems from the last BLSTM layer.

Two mid-level combination strategies were presented. The first method consists on the application of a linear weighted combination of the log a-posteriori extracted from the net outputs. Then, the resulting log- probabilities were injected in the CTC decoding layer. This combination pipeline, known as net averaging technique, is presented in Figure 5.

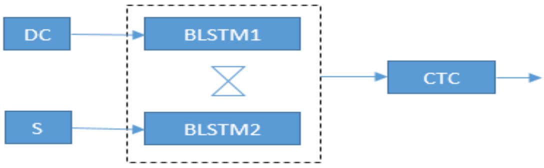

Figure 5. Mid-level combination (A)

The second combination method is inspired from the autoencoder concept [22]. A set of F BLSTM-CTC is trained on the F different feature representations in a first stage. Then, the CTCs are removed, hence removing the individual layers that transform each feature signal into a label sequence. The remaining BLSTMs output are then merged by training a new BLSTM architecture followed by a CTC layer as described in Figure 6.

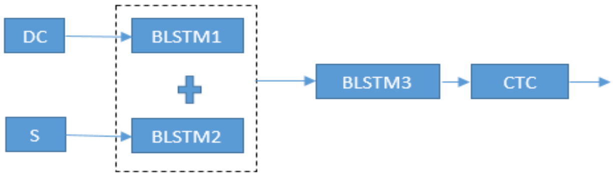

Figure 6. Mid-level combination (B)

C. High Level Combination

In the case of high-level combination, the generated outputs (lattices/hypothesis) of the two BLSTM-CTC recognition systems were combined using two techniques (see Figure 7). The first method is the lattice combination and it is based on MBR (Minimum Bayes Risk) combination [15]. In the lattice-based system combination task, the goal is to combine and decode lattices provided by several systems. MBR strategy consists in combining two algorithms; the standard quadratic-time algorithm used to calculate the Levenshtein distance, and the forward–backward algorithm used to compute the probabilities of crossing the lattice arcs. This method has proven its performance by reducing the WER of the proposed systems. The second combination strategy is ROVER hypothesis fusion. It is based on a sequence alignment coupled with maximum confidence scores, as presented in [5]. The combined systems order has an effect on the combination results. The most accurate system is placed in the first order of this process.

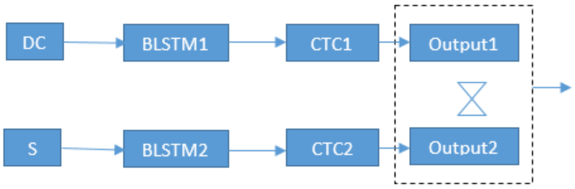

Figure 7. High-level combination

IV. Results and Analysis

A. Data Set Description

The performance of the proposed system combination levels was evaluated on the benchmarking dataset KHATT. KHATT [16] is a freely off-line handwritten text database consisting of 4000 paragraphs written by 1000 distinct writers. Text images are scanned at 300 dpi. In this work, the experiments were conducted on 9475 line images for training, 1901 for validation and 2007 for test. This database presents an OOV rate equal to 11.46%. Table I reports some statistics on the KHATT dataset used in the experiments.

TABLE I.
KHATT Statistics

Subset	Pages	Lines	Words	Characters
Train	690	9,475	129,826	605,537
Test	141	2,007	26,449	122757
Dev.	148	1,901	26,142	121,433

B. Experiments and Results

The main goal of these experiments was to evaluate and analyze the performance of the different combination level strategies. Performance was measured in terms of Word Error Rate (WER) and Character Error Rate (CER). To evaluate the accuracy of the presented systems, we used Levenshtein edit distance between the output text and the ground-truth. Edit distance is calculated by computing the number of edit operations (insertions, substitutions and deletions) that are needed to transform a source string into the target string.

$$\mathrm{WER} = \frac{\text{substitutions} + \text{insertions} \ + \text{suppressions .number}}{\text{Number of words in the reference}} * 100$$

Two main scenarios were taken into account while achieving the experiments:

1) scenario 1: In this task, the used lexicon consists of all the tokenized words extracted from the KHATT corpus. A 23K distinct words are used to direct the recognition process. The proposed systems accuracy is reported in table II.

TABLE II.
Individual System Results for Scenario 1 (in %)

	WER%	
System	**Test Set**	**Dev. Set**
Segment Based	31.96	34.27
+3-gram LM	17.36	19.89
DC	29.44	31.60
+3-gram LM	**15.87**	**17.42**

As shown in this table, the recognition results are improved when using the LM rescoring in the post-processing stage. The WER is respectively reduced by 14.60% and 13.57% for the segment-based and DC-based systems using the test set. As shown in the previous section, the features based on pixel distribution and concavity are more accurate than the segment-based features, with an improvement of 1.49% and 2.47% in terms of WER respectively on the test and dev. sets.

As our main objective was to study the system combination at different levels, we present in tables III, the obtained results of the different combination levels.

TABLE III.
Combination Results for Scenario 1 (in %)

Combination Strategy		WER		CER	
		Test	Dev.	Test	Dev.
Low-L.	F. Fusion	15.44	16.71	8.32	8.71
Mid- L.	Comb. (A)	15.84	17.61	8.93	9.61
	Comb. (B)	17.04	18.56	9.34	9.87
High- L.	Lat. Comb.	14.38	16.09	8.23	8.87
	ROVER	**13.52**	**15.19**	**7.85**	**8.35**

As shown in table III, the low-level combination strategy reduces the WER by respectively 0.43% and 0.73% on test and dev. sets. These results prove the features complementarity and the BLSTM-CTC capacity to treat height dimensional features. In order to combine several features using a BLSTM-CTC it is therefore preferable to directly combine the features.

For the mid-level combination, two strategies were evaluated. The first one consists in smoothing the log probabilities issued from the last BLSTM layer using a weighted log probability sum. A slight improvement of 0.03% in term of WER was achieved on the test set. In fact, the two systems may not generate the labels at the same timestep, therefore a good output label may be lost after the averaging stage. For the second strategy, which consists in using log-probabilities as features to train a new BLSTM-CTC system, the obtained results are not improved. As a result, the BLSTM-CTC can learn some information from the log probabilities, but not sufficient to ameliorate the overall system performance.

For the high level results, the lattice combination based on MBR decoding is interesting; it has respectively about 1.49% and 1.33% improvement in WER for both sets (test and dev.) when compared to the best individual system. However, experiments conducted using the ROVER combination technique show the best results with a WER reduction of 2.35% on the test set and 2.23% on the dev. set of the KHATT database.

2) Scenario 2: For this task, the used lexicon is restricted to words running in the training corpus. 18K distinct words were used in the vocabulary representing an OOV (Out Of Vocabulary) rate of 11.46%.

TABLE IV.
INDIVIDUAL SYSTEM RESULTS FOR SCENARIO 2 (IN %)

	WER%	
System	**Test Set**	**Dev. Set**
Segment Based	39.15	40.86
+3-gram LM	32.34	34.22
DC	37.80	38.94
+3-gram LM	**30.33**	**31.55**

The results of individual systems are presented in table IV. The DC features-based system outperforms the segment-based one and a gain of 2.01% in WER using a 3-gram LM as post-processing step is obtained.

TABLE V.
COMBINATION RESULTS FOR SCENARIO 2 (IN %)

Combination Strategy		**WER**		**CER**	
		Test	**Dev.**	**Test**	**Dev.**
Low-L.	F. Fusion	29.78	30.64	15.14	14.96
Mid- L.	Comb. (A)	30.36	31.50	16.15	15.88
	Comb. (B)	31.86	33.08	16.64	16.90
High- L.	Lat. Comb.	29.22	30.33	16.14	16.35
	ROVER	**29.13**	**30.32**	**16.27**	**16.51**

The obtained results presented in table V, confirm the superiority of high-level combination strategy compared to low-level and mid-level combination results for both presented scenarios. The obtained combination results support the hypothesis that the various systems make distinct mistakes, and they are able to complement each other. The difference between ROVER and lattice combinations is small on the two set of KHATT database. Nonetheless, the ROVER combination is clearly better than the lattice combination. These results which proves the complementarity of the systems due to the generation of various outputs.

TABLE VI.
RESULTS FOR SCENARIO 2 (IN %)

	WER%	
System	**Test Set**	**Dev. Set**
Our Best	**29.13**	**30.42**
BenZeghbia et al. [19]	33.00	31.50
Hamdani et al. [18]	33.60	34.10
Stahlberg et al. [17]	30.50	29.40
BenZeghbia [21]	34.30	33.70
BenZeghbia [23]	23.00	24.10

We compare in table VI, the obtained results to other works using KHATT dataset. As shown in tab VI, we succeeded in attaining a comparable performance with the achieved systems in the literature.

A closer analysis of the results revealed that most of the misclassification errors originated from the recognition of punctuation forms, diacritics misplacement in the text line image and false occlusions, as shown in Figure 9.

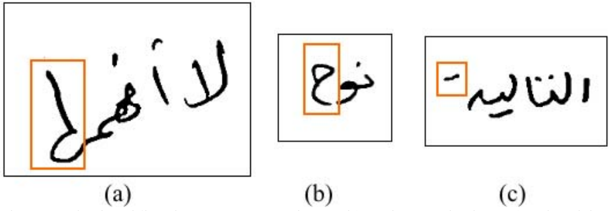

Fig. 9. Misclassification error samples: a-b) False occlusion, c) Diacritics misplacement.

V. CONCLUSION

We present in this paper a comparison of different combination strategies for the recognition of handwritten Arabic text based on BLSTM networks. Three combination levels were compared using KHATT database. The experimental results have shown that the high-level combination approach performs better than other levels for the two tasks. The accuracy of our system was improved, respectively for the test and dev. sets, by 2.35% and 2.23%, in term of WER, using a full vocabulary and by 1.2% and 1.23% in the presence of OOV words. Thanks to these results, our system yielded comparable results to those presented in the literature. Future work will explore the possibility of adding new features sets, especially training features using CNN. We can also explore the combination of combined levels.

APPENDIX

We define H as the height of the frame in an image, h be the variable height of a cell, w be the width of a frame, and n_c be the number of cells in a frame. The feature based distribution consists of 11 features that describe the density of the foreground pixels within frames. The first feature F1 is given by:

$$F1=\sum_{i=1}^{n_c} n(i) \tag{3}$$

where : $n(i)$ is the number of foreground pixels in the i-th cell (a cell inside a frame corresponds to one pixel).

Feature F2 is defined as:

$$F2=\sum_{i=2}^{n_c} \left|b(i)-b(i-1)\right| \tag{4}$$

where $b(i)$ is the density level of cell i , $b(i)$ is equal to one if the cell contains a least one foreground pixel and is equal to zero otherwise.

F7 is given by:

$$F7=\frac{g-L}{H} \tag{5}$$

where L is the position of the lower baseline and g is defined as follows:

$$g=\frac{\sum_{j=1}^{H} j.r(j)}{\sum_{j=1}^{H} r(j)} \tag{6}$$

where $r(j)$ is the number of foreground pixels in the j-th row of a frame.

Feature F8 and F9 are given by:

$$F8=\frac{\sum_{j=L+1}^{H} r(j)}{H.w} \tag{7}$$

$$F9=\frac{\sum_{j=1}^{L-1} r(j)}{H.w} \tag{8}$$

Feature F10 is given by:

$$F10=\sum_{i=k}^{n_c} \left|b(i)-b(i-1)\right| \tag{9}$$

where k is the cell that contains the lower baseline.

The concavity features are computed as follows: N_{lu} , (respectively, N_{ur} , N_{rd} , N_{dl} , N_v , and N_h) is the number of background pixels that have adjacent black pixels in the following directions: left and up (respectively, up-right, right-down, down-left, vertical, and horizontal). The six normalized concavity features are defined as:

$$F12=\frac{N_{lu}}{H} \tag{10}$$

$$F13=\frac{N_{ur}}{H} \tag{11}$$

$$F14=\frac{N_{rd}}{H} \tag{12}$$

$$F15=\frac{N_{dl}}{H} \tag{13}$$

$$F16=\frac{N_v}{H} \tag{14}$$

$$F17=\frac{N_h}{H} \tag{15}$$

References

[1] Velek O., Jäger S., Nakagawa M., "Accumulated-recognition-rate normalization for combining multiple on/off-line Japanese character classifiers tested on a large database", in Proceedings 4th Workshop on Multiple Classifier Systems, 2003.

[2] R. Bertolami, H. Bunke, "Early feature stream integration versus decision level combination in a multiple classifier system for text line recognition", in Proceedings of International Conference on Pattern Recognition, pp. 845–848, 2006.

[3] Mioulet L., Bideault G., Chatelain C., and Paquet T., "BLSTM CTC Combination Strategies for Off-line Handwriting Recognition", in International Conference on Pattern Recognition Applications and Methods, 2015.

[4] Mioulet L., Bideault G., Chatelain C., Paquet T. and Brunessaux S., "Exploring multiple feature combination strategies with a recurrent neural network architecture for off-line handwriting recognition", in Document Recognition and Retrieval, 2015.

[5] Fiscus, J., "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)", in IEEE Workshop on Automatic Speech Recognition and Understanding, 347–354, 1997.

[6] Bo-June Hsu, "Generalized linear interpolation of language models," in ASRU. IEEE Workshop, pp. 136–140, 2007.

[7] Hai-Son Le, Ilya Oparin, Alexandre Allauzen, J Gauvain, and Franc¸ois Yvon, "Structured output layer neural network language models for speech recognition", Audio, Speech, and Language Processing, IEEE Transactions on, vol. 21, no. 1, pp. 197–206, 2013.

[8] A. Sankar, "Bayesian model combination (baycom) for improved recognition", in ICASSP 2005.

[9] J. Sauvola, M. PietikaKinen, "Adaptive document image binarization", In Pattern Recognition, 2000.

[10] Jonathan J. Hull, "Documents skew detection: Survey and annotated bibliography", In Document Analysis Systems 1998.

[11] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoint", in International Journal of Computer Vision, 2004.

[12] N. Dalal, B. Triggs, " Histograms of Oriented Gradients for Human Detection", in CVPR 2005.

[13] Yajie Miao, Mohammad Gowayyed, Florian Metze, "EESEN: End-to-End Speech Recognition using Deep RNN Models and WFST-based Decoding", in ASRU 2015.

[14] Graves, A. and Gomez, F., "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks", in Proceedings of the 23rd International Conference on Machine Learning, 2006.

[15] Haihua Xu, Daniel Povey, Lidia Mangu, Jie Zhu, "Minimum Bayes Risk decoding and system combination based on a recursion for edit distance", in Computer Speech and Language, 2011.

[16] S. A. Mahmoud, I. Ahmad, M. Alshayeb, W. G. Al-Khatib, M. T. Parvez, G. A. Fink, V. Margner, and H. El Abed, " Khatt: Arabic offline handwritten text database", in ICFHR 2012.

[17] Felix Stahlberg and Stephan Vogel, "The QCRI Recognition System for Handwritten Arabic", in ICIAP 2015.

[18] M. Hamdani, A. El-Dosoky Mousa and H. Ney, "Open Vocabulary Arabic Handwriting Recognition Using Morphological Decomposition" , in ICDAR 2013.

[19] M. F. BenZeghiba, Jerome Louradour and Christopher Kermorvant, "Hybrid Word/Part-of-Arabic-Word Language Models For Arabic Text Document Recognition", in ICDAR 2015.

[20] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit", in International Conference on Spoken Language Processing, 2002.

[21] M. F. BenZeghiba, "Arabic Word Decomposition Techniques for Offline Arabic Text Transcription", in IEEE International Workshop on Arabic Script Analysis and Recognition (ASAR), 2017.

[22] Yoshua Bengio, "Learning Deep Architectures for AI", Foundations and Trends in Machine Learning, 2009.

[23] M. F. BenZeghiba, "A Comparative Study On Optical Modeling Units For Off-line Arabic Text Recognition", in the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), 2017.

[24] Stahlberg F., Vogel S., "Detecting dense foreground stripes in Arabic handwriting for accurate baseline positioning ", in: ICDAR. IEEE, 2015.

[25] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition", Computer Speech & Language, vol. 16, no. 1, pp. 69–88, 2002.