

COMPLETE KAZAKH HANDWRITTEN PAGE RECOGNITION USING START, FOLLOW AND READ METHOD

¹RUSLAN JANTAYEV, ²SHIRALI KADYROV, ^{1,3}YEDILKHAN AMIRGALIYEV

¹Computer Sciences Department, Suleyman Demirel University, Kazakhstan

²Mathematics and Natural Sciences Department, Suleyman Demirel University, Kazakhstan

³Al-Farabi Kazakh National University, Kazakhstan

E-mail: ¹ruslan.jantayev@sdu.edu.kz, ²shirali.kadyrov@sdu.edu.kz, ³yedilkhan.amirgaliyev@sdu.edu.kz

ABSTRACT

In this article we consider end-to-end full page Handwritten Text Recognition for offline Kazakh text images written in Cyrillic alphabet using Fully connected CNN and bidirectional LSTM. The model performs training of text segmentation and recognition jointly using a new Kazakh text images dataset, named Kazakh Handwritten Dataset (KHD). The novel method, which we introduce, uses three steps: Start, Follow and Read (SFR). The proposed model makes use of Region Proposal Network in order to find the starting coordinates of lines in the page. For the case when lines are not straight, we introduce a method that pursues text lines until the end of it and prepare it for the last recognition step. The SFR model works for Russian language as well since Russian alphabet is a subset of Kazakh alphabet. The experimental analysis shows that on average the model provides 0.11 Character Error Rate.

Keywords: *Computer Vision, HTR, CNN, Bidirectional LSTM, Kazakh Handwritten, Document Processing, Text Line Follower, Text line cutting.*

1. INTRODUCTION

Optical character reader (OCR) is a technology in the field of pattern recognition used to convert images of text into machine-encoded text which in particular enables search and edit options. To obtain a comprehensive OCR solution, it should be able to recognize text images regardless whether they are typed or handwritten. The offline-handwritten recognition is a very important domain of research as it appears in various kinds of images including lecture notes, notes written by graphic tablets, whiteboards, doctor's notes, and historical documents. It finds applications in reading postal addresses, car plates, bank checks, various forms, offline exam papers, and in digitization of historical documents etc. Moreover, a multilingual OCR solution would be highly appreciated [1].

Our goal in this article is two-fold; the first objective is to introduce our own dataset Kazakh Handwritten Dataset (KHD) of

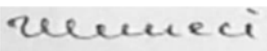
handwritten Kazakh texts in Cyrillic alphabet and the second objective is to propose the SFR model to recognize full-page handwritten Kazakh text. We note that the current Kazakh alphabet is in Cyrillic script and consists of 42 letters with 33 of them coming from Russian alphabet.

In general, it is a very time consuming and difficult task to establish a suitable dataset of handwritten text images. Moreover, most open access datasets are available in widely spoken languages which leaves a very tiny window to do research in handwritten recognition in less popular languages. In particular, there is almost no dataset of handwritten text images available in Kazakh [2] which makes it a very crucial task to develop new datasets. Therefore, our first objective to establish an open access dataset Kazakh Handwritten Dataset (KHD) of handwritten text images for end-to-end recognition tasks will help save time and budget for the researchers in the field. Such work will make a contribution to strengthening international research [3]. A very recent work [4] on Kazakh

handwritten recognition brought to our attention where a labeling approach to collect a dataset is proposed and for the collected dataset the authors reach accuracy of 85.63 %.

Automated handwritten text recognition (HTR) is very crucial as it saves a lot of time effort. While the state-of-the-art HTR showed promising progress in some languages in recognizing handwritten certain text images [3], it still remains a challenging task due to variations in handwriting styles and differences in language specific features [5]. Especially in Cyrillic alphabet handwritten (cursive) text can be very confusing. For example, the handwritten text in Table 1 from our dataset KHD may correspond to any of the typed text in the second column. It means “his mother” which corresponds to the first typed text in the second column and remaining words are gibberish.

Table 1. Confusing nature of Cyrillic handwritten text.

Kazakh handwritten word	Possible transcriptions
	шешесі, шииесі, шлиесі, гшшесі, гешіесі, геешесі, гииіесі, иеемесі, иіиіесі, ...

Thus, the state-of-the-art model in one language may not perform as expected in another language.

To sum up, our first contribution is to generate a new open access dataset KHD made available upon request. The second contribution is to propose modified version of SFR [8] adopted to the new dataset with moderate modifications. For details we refer to section 3.

2. RELATED WORK

In this section we review the recent work, progress on HTR, and the state-of-the-art algorithms in the field.

Handwritten text recognition is one of the active areas of research in computer vision. Artificial Neural Networks, Kernel Methods such as Support Vectors Machines and Principal Component Analysis, Statistical Techniques such

as Logistic regression, Hidden Markov Models (HMM), and k-Nearest Neighbors, Template matching methods, and finally Structural pattern recognition methods including Chain code histogram and grammar-based approaches are some of the classical HTR approaches used in the past three decades [6], [7], [3]. With the elevation of the large database availability, the deep learning-based approaches with special attention to Convolutional Neural Networks (CNN) are becoming increasingly popular among the current researchers and most recent CNN based techniques can be considered as the state-of-the-art approaches [3]. In artificial intelligence, neural networks are sophisticated machine learning technologies inspired by well-connected human neural networks that generate electrical activities through neurons in our nervous system. In simple mathematical terms, an Artificial Neural Networks model is a highly nonlinear function from the space of input (independent) variable to the space of output (dependent) variables. Nonlinearity comes from activation functions such as ReLU, tanh, and sigmoid. With simple linear activations, the ANN model would become a simple finite dimensional linear transformation, in other words, it would simply be a matrix multiplication and hidden layers would contribute nothing to the training process. With a sufficient number of hidden layers with adequate number of nodes, a neural network can approximate almost any nonlinear (and linear) continuous function. In ANN, an n dimensional input variable is regarded as an $n \times 1$ vector. For example, if we want to train our ANN model for a human face recognition problem, then the system first reads the two-dimensional faces into two-dimensional matrix of numbers with possible thirds dimension on color images, but then it flattens the input into a column vector and hence losing the topological structure and geometric structure of the original input. On the other hand, Convolutional Neural Networks have the power of keeping the input image dimensions as they are and hence preserving the topological structure. As the handwritten text is a two dimensional image, CNN-based architecture is definitely worth trying.

A comprehensive HTR system has two main parts, namely, text segmentation and transcription. While more recent approaches consider the training of their models as a combined form of the two parts [8]–[10] the

traditional approach was to consider them as two separate tasks [11]–[14].

The text segmentation means to pre-process the text image through line annotation of the text position or subdividing the text into meaningful units such as letters or words. Here we briefly review some related work on text line detection. One traditional approach was to sum the pixel values of the rectangular image along the rows, thus obtaining a vertical vector [15]. The local minima of the vector would give the positions where the text lines appear. Clearly, standing assumptions here are that the text lines are straight without any curving and that any two text lines have a gap. To accommodate for the curved text lines, seam carving techniques were proposed [16]. The idea is to compute the value of information energy for each pixel where a pixel containing part of a text has high energy. Then, the text line is determined by continuous elimination of pixels with large energy. More recent improvements of seam carving are based on CNN and Bi-directional Long-Short Term Memory (BLSTM) [11]. In general terms, Long-Short Term Memory architectures and in particular BLSTM models are widely used in natural language processing research field. LSTM is a part of recurrent neural network architecture effective in the study of ordered sequences and time series such as words, sentences. The connections among the nodes of recurrent neural networks are in the form of directed graphs within time series which gives them the ability to remember the input variables during the learning process. LSTM technology can process the whole sequence, say a word, hence preserving the dependencies between the data points. It usually consists of one cell which memorizes the values over certain time intervals and three types of gates, namely, an input gate, an output gate, and finally a forget gate that controls the information flow.

Recently, Progressive Scale Expansion Network, a kernel-based model, was proposed which first considers pixel-level segmentation to locate a text instance and then a progressive scale expansion algorithm is used to determine the adjacent text instances [17].

Once the text line is found the next step is text transcription through character image recognition. In various situations the Hidden Markov Models (HMM) approach proved

effective in HTR [13], [18]. Hybrid Recurrent Neural Network (RNN) models with CNN started becoming effective character recognizers as a result of development of Connectionist Temporal Classification (CTC) loss function [19]. Introduced in 2006 [20], CTC is a loss function which is nowadays used in LSTM as an alternative to HMM for sequence-to-sequence learning problems. CTC is related to the conditional probability of correctly identifying the labeled sequence and is given by

$$CTC(\mathbf{1}, \mathbf{x}) = -\ln \sum_{\pi \in B^{-1}(\mathbf{1})} p(\pi|\mathbf{x}),$$

where $\mathbf{1}$ is the label corresponding to the sequence \mathbf{x} and B is transformation that maps the given sequence into its label, for details see e.g. [21].

A very recent work on handwritten Russian and Kazakh text recognition uses novel Attention-Gated-CNN architecture with bidirectional gated recurrent unit and CTC loss [2]. For more information on recent advances in HTR we refer to [22].

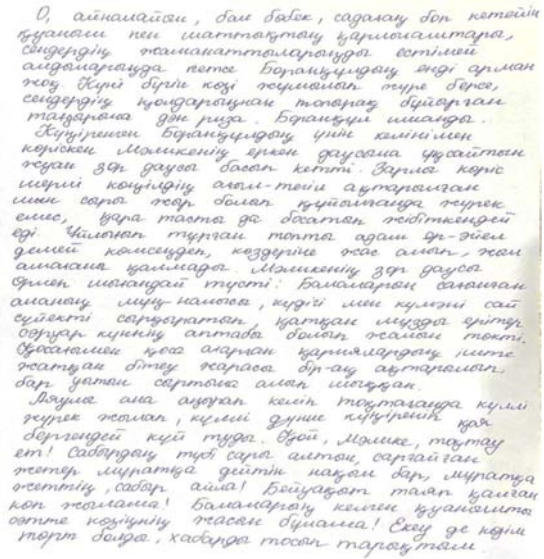
Recently, the end-to-end full page HTR models that train text segmentation and transcription jointly showed promising performance [8]. Wigington et al [8] proposed Region Proposal Network (RPN) to determine the start of a line, a recurrent network approach to segment polygonal regions for text lines, and finally application of CNN-LSTM to identify the characters for historical documents in English. Another end-to-end HTR method was proposed in [23] where the algorithm finds the start of the line, and then the network recognizes the characters without explicit determination of the end of the text line. In [24] the end-to-end model was integrated with feature extractor ResNet-50 improving the prediction speed. We finally note that a study [9] suggests that joining segmentation with recognition is more likely to improve the model performances.

3. EXPERIMENTAL SETUP AND PROPOSED MODEL

In this section we first introduce the details of our new dataset of handwritten text images and data pre-processing and segmentation procedures. We then provide some details of the proposed algorithm.

3.1. Dataset and Pre-processing

Dataset consists of A4 format 400 pages of handwritten text images, collected manually with participation of over 40 university students. Each page of handwritten text image has a corresponding original text file available. Both image file and text file were processed and validated manually to keep common format among all participants. All images were rescaled to have a fixed width of 512px. According to our calculations, one handwritten text page contains on average 35 lines. Each line contains around 5 words i.e. around 60 letters Fig. 1. Overall, we have 14,000 text lines, 70,000 Kazakh words and 420,000 letters. In order to best train our model, we used 90% (360 pages) of our data for training and 10% (40 pages) for validation purposes which is a k-fold validation method. For the purpose of cross validation we consider two test data Test 1 and Test 2.



О, айналайын, бал бөбек, садаған боп кетейін
 қуаныш пен шаттықтың қарлығаштары,
 сендердің жаманаттыларыңды естімей
 алдыларыңда кетсе Боранқұлдың енді арман
 жоқ. Күні бүгін көзі жұмылып жүре берсе,
 сендердің қолдарыңнан топырақ бұйырған
 тағдырына дән риза, Боранқұл иманды.
 Күніренген Боранқұлдың үнін келінімен
 көріскен Мәлікенің еркен даусына ұқсайтын
 жуан зор даусы басып кетті. Зарлы көріс
 шерлі көңілдің ағыл-тегіл ақтарылған
 шын сыры жыр болып құйылғанда жүрек
 емес, қара тасты да босатып жібіткендей
 еді. Ұйлығып тұрған топты адам ер-әйел
 демей кемсеңдеп, көздеріне жас алып, жыл
 амағаны қалмады. Мәлікенің зор даусы
 өрлеп шығандай түсті: Балаларын сағынған
 ананың мұң-наласы, күдігі мен күмәні сай
 сүйекті сырқыратып, қатқан мұзды ерітер
 сәруар күннің аптабы болып жалын төкті.
 Қосағымен қоса ағарған қариялардың іште
 жатқан бітеу жарасы бір-ақ ақтарылып,
 бар уытын сыртына алып шыққан.
 Аяулы ана аңырап келіп тоқтағанда күллі
 жүрек жылап, күллі дүние күніреніп қоя
 бергендей күй туды. Қой, Мәлике, тоқтау
 ет! Сабырдың түбі сары алтын, сарғайған
 жетер мұратқа дейтін нақыл бар, мұратқа
 жеттің, сабыр айла! Бейуақыт таяп қалған
 көп жылама! Балаларың келген қуанышты
 сәтте көзіңнің жасын бұлама! Екеу де көзім
 төрт болды, хабарыңды тосып тарықтым.

О, айналайын, бал бөбек, садаған боп кетейін
 қуаныш пен шаттықтың қарлығаштары,
 сендердің жаманаттыларыңды естімей
 алдыларыңда кетсе Боранқұлдың енді арман
 жоқ. Күні бүгін көзі жұмылып жүре берсе,
 сендердің қолдарыңнан топырақ бұйырған
 тағдырына дән риза, Боранқұл иманды.
 Күніренген Боранқұлдың үнін келінімен
 көріскен Мәлікенің еркен даусына ұқсайтын
 жуан зор даусы басып кетті. Зарлы көріс
 шерлі көңілдің ағыл-тегіл ақтарылған
 шын сыры жыр болып құйылғанда жүрек
 емес, қара тасты да босатып жібіткендей
 еді. Ұйлығып тұрған топты адам ер-әйел
 демей кемсеңдеп, көздеріне жас алып, жыл
 амағаны қалмады. Мәлікенің зор даусы
 өрлеп шығандай түсті: Балаларын сағынған
 ананың мұң-наласы, күдігі мен күмәні сай
 сүйекті сырқыратып, қатқан мұзды ерітер
 сәруар күннің аптабы болып жалын төкті.
 Қосағымен қоса ағарған қариялардың іште
 жатқан бітеу жарасы бір-ақ ақтарылып,
 бар уытын сыртына алып шыққан.
 Аяулы ана аңырап келіп тоқтағанда күллі
 жүрек жылап, күллі дүние күніреніп қоя
 бергендей күй туды. Қой, Мәлике, тоқтау
 ет! Сабырдың түбі сары алтын, сарғайған
 жетер мұратқа дейтін нақыл бар, мұратқа
 жеттің, сабыр айла! Бейуақыт таяп қалған
 көп жылама! Балаларың келген қуанышты
 сәтте көзіңнің жасын бұлама! Екеу де көзім
 төрт болды, хабарыңды тосып тарықтым.

Figure 1: Kazakh handwritten text page with corresponding text file (label) from KHD dataset.

Vectorized lines were made over rescaled images to provide base information about lines. Segments of lines were cut out over vectorized paths. Horizontal and vertical histogram were calculated for each segment. The most left boundary with higher derivative on horizontal histogram was selected as starting x coordinate and the most higher derivatives from top and bottom were selected to get scale and y coordinate. Tangent line over vectorized path was used to calculate radians of start of line.

3.2. Proposed algorithm

Our proposed model is similar to [8] with main distinction in the line follower algorithm. For the sake of completeness, we provide details here. The end-to-end HTR algorithm has three main components: Start-of-Line finder, Line Follower, and finally Handwritten Text Recognizer.

Start-of-Line finder. Since we consider the full pages of texts, the first mission is to find the starting points of each text line. The Start-of-Line (SOL) follower is an RPN with a truncated Very Deep Convolutional Network VGG-19 architecture [23], [25]. We chose to work with VGG-19 due to its higher capacity as opposed to VGG-11 used in [8], see Fig. 2.

The truncated VGG-19 produces five dimensional vectors as an output, namely, the coordinates of the starting point of a line, the direction angle of the text line, the scale, and a probability of occurrence. The image patch that contains the SOL is determined according to whether it is one or zero. Once the correct image patch is found, the coordinates are set to be the center of this rectangular patch. The scale gives the size of the text and the direction of text slope.

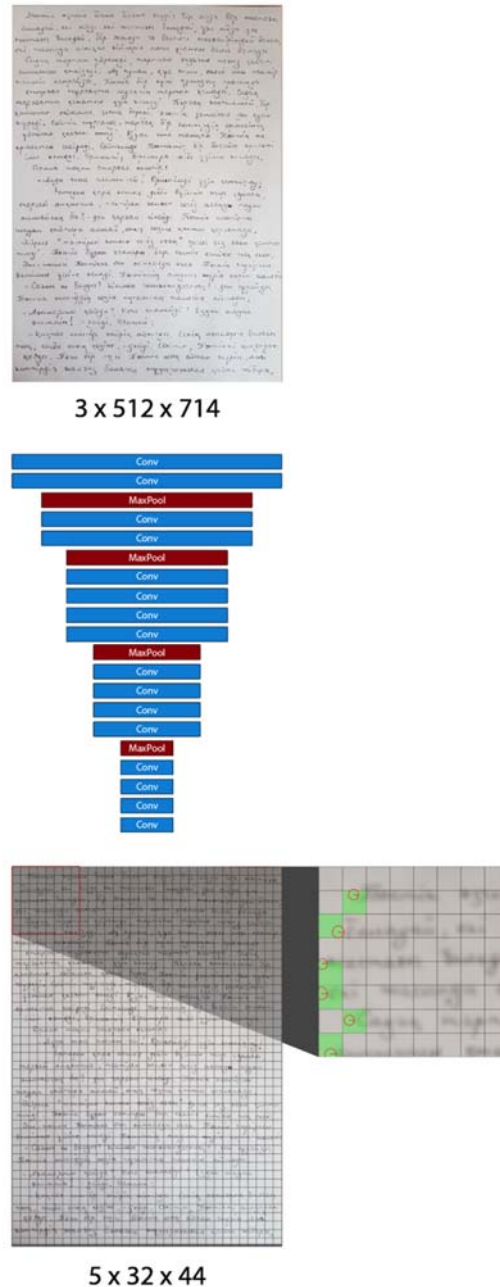


Figure 2 : RPN based SOL

Below is the Architecture of SOL:

1. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
2. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
3. Max pool: Kernel = 2x2, Stride = 2, Padding = 0

4. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
5. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
6. Max pool: Kernel = 2x2, Stride = 2, Padding = 0
7. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
8. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
9. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
10. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
11. Max pool: Kernel = 2x2, Stride = 2, Padding = 0
12. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
13. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
14. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
15. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
16. Max pool: Kernel = 2x2, Stride = 2, Padding = 0
17. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
18. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
19. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1
20. Convolution layer: Kernel = 3x3, Stride = 1, Padding = 1

Line Follower. Once the starting point of each line is determined by SOL, the next problem is to follow the line where the text appears. To this end, a piece of image is cut at the start of a line. This cut image is resized to have a fixed size of height 25 px and width 512 px. Customized network model predicts next position given the cut image. On the output it gives the following values: directions, distance, angle respectively, see Fig. 3.

For this purpose, we use a different methodology than [8]. More precisely, we implement the CNN model is implemented with its topology consisting of seven layers: 3x3

kernels and 64, 128, 256, 256, 512, and 512 feature maps. After 4th and 5th layers, Batch Normalization (BN) is implemented and a 2x2 Max Pooling (MP) filter is applied after 1st, 2nd, 4th, and 6th layers. Finally, a fully connected layer is implemented to obtain the next coordinates and the direction as outputs. The model predicts positions recursively and stops when the end of the line is reached. The model was trained in an unordinary way. During the training process windows are rotated 180 degrees randomly in order to follow in reversed directions. It gives the possibility to detect the end of lines. We have figured out that the model gets lost at the end of a line. This tricky case was used to detect the end of line. So, the line follower moves a certain number of steps forward and tries to come into the initial position backward. Followed by forward and backward paths are compared. If the difference between two paths is significant, then the text line is lost.

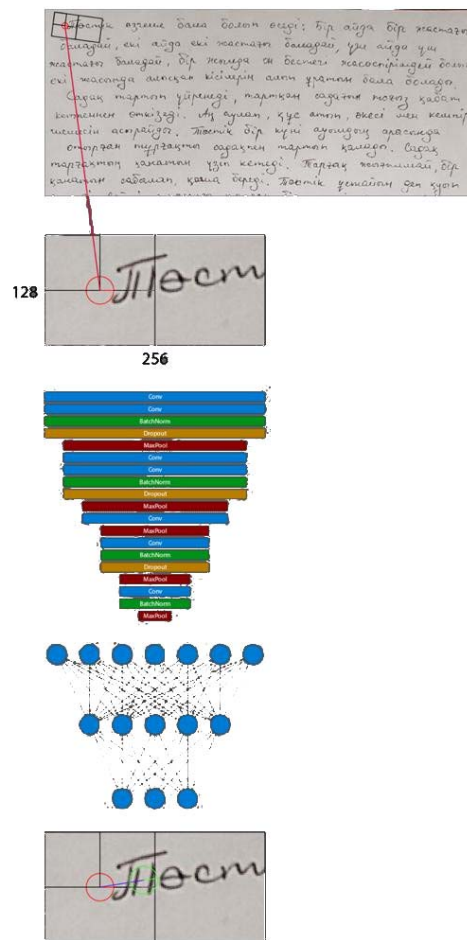


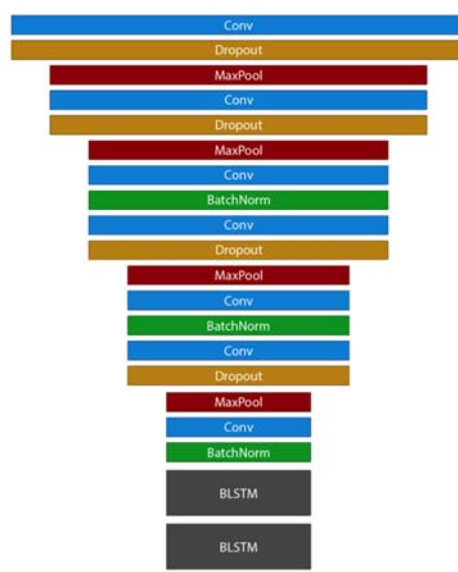
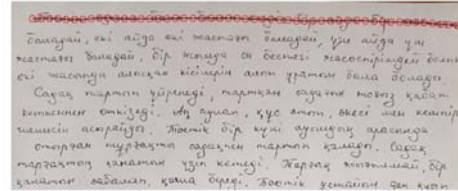
Figure 3 : Line Follower

Below is the Architecture of LF:

1. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 128x256x64
2. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 128x256x128
3. Batch Normalization
4. Dropout
5. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 64x128x128
6. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 64x128x256
7. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 64x128x256
8. Batch Normalization
9. Dropout
10. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 32x64x256
11. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 32x64x256
12. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 16x32x256
13. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 8x16x128
14. Batch Normalization
15. Dropout
16. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 4x8x128
17. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 2x4x128
18. Batch Normalization
19. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 1x2x128
20. Linear, output: 256
21. Linear, output: 128
22. Linear, output: 3 (r,d,nr)

Handwritten Text Recognizer. Given the SOL and the entire line where text lies, the final duty is to recognize handwritten characters. To this end, a deep learning architecture, CNN-LSTM HWR, similar to[26] is considered as shown in Fig. 4. More specifically, the model topology consists of six convolutional layers 64-128-256-256-512-512. The BN filters are used in 4th and 5th layers, after layer 1 and 2 a max pooling filter with stride 2 is applied, and lastly 4th and 5th layers followed the max pooling filters with vertical stride of 2 and a horizontal stride of 1. The list of 1024-dimensional feature vectors are formed and are

fed to a 2-layer BLSTM network with 512 nodes and 0.5 probability of dropout of each node.



1. [0, 0, 0, 1, 0, 0, ..., 0]
2. [0, 0, 0, 0, 0, 0, ..., 0]
3. [0, 1, 0, 0, 0, 0, ..., 0]
4. [0, 0, 0, 0, 0, 1, ..., 0]
5. [1, 0, 0, 0, 0, 0, ..., 0]
- ...
64. [0, 0, 0, 0, 0, 0, ..., 1]

Figure 4: Text Recognizer

Below is the Architecture of LF:

1. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 32x512x64
2. Dropout
3. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 16x256x64

4. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 16x256x128
5. Dropout
6. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 8x128x128
7. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 8x128x256
8. Batch Normalization
9. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 8x128x256
10. Dropout
11. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 4x64x256
12. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 4x64x512
13. Batch Normalization
14. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 4x64x512
15. Dropout
16. Max Pool: Kernel =2x2, Stride = 2, Padding = 0, Output = 4x64x512
17. Convolutional Layer: Kernel =3x3, Stride = 1, Padding = 1, Output = 1x64x512
18. Batch Normalization
19. Bidirectional LSTM, output: 64x512
20. Bidirectional LSTM, output: 64x512

Note that the [2] made an experiment on text-lines of Kazakh/Russian language and received CER 0.64 accuracy test1 and CER 0.45 accuracy. However, our work is done for whole page text recognition which is much advanced work. We can easily state that our work is a continuation of [2] with new KHD dataset and with novel SFR model. Our SFR model simultaneously reads whole lines at time, i.e our SFR model reads 33 times faster than [2]. Our novel SFR model shows CER 0.11 on test_1 and CER 0.13 on test_2 as shown in Table x.

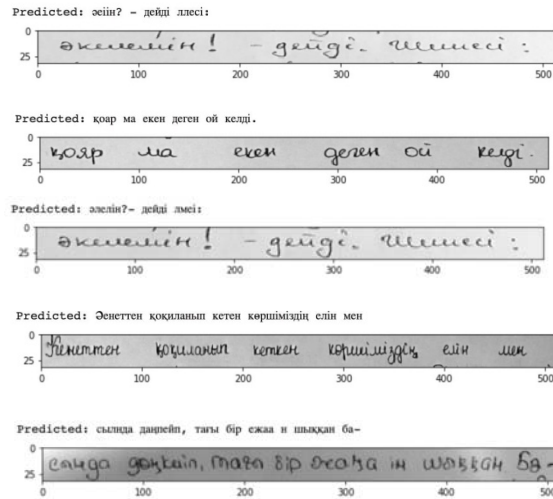


Figure 5: Predicted: Some output results of our experiment for Kazakh/Russian language

4. RESULTS OF THE EXPERIMENT

In this section we state the findings of our experiment. To report the performance of our model we use the Character Error Rate (CER) as a metric, which gives the percentage of error made during character recognition also known as Levenshtein distance [27]. More specifically, for a text with total (N) characters, CER is given by

$$CER = \frac{S + I + D}{N}$$

where (S) is the minimal number of character substitutions, (I) the minimal number of character inclusions, and (D) the minimal number of character deletions required to transform the original text into the model predicted text. Clearly, the lesser the CER, the better the model performance.

Fig. 5 provides some examples of model predicted text and the handwritten text.

Table 2: CER result on test_1 and test_2

Test	Dataset	CER
Test 1	KHD	0.11
Test 2	KHD	0.13

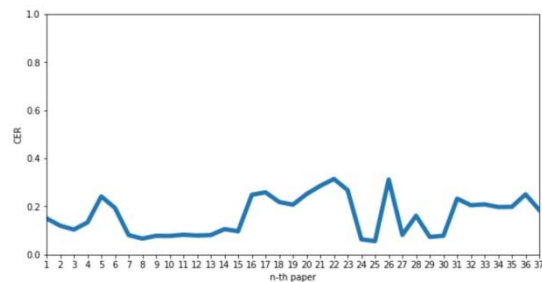


Figure 6: Graph of CER for each test page

We also report one example Fig 7 to show the performance of both the start of line finder and line follower parts of our model. The red dots show predicted lines.



Figure 7: The result of line follower for one page

It is readily seen that the algorithm is doing a great job to determine the lines for this particular page sample.

4. DISCUSSION AND CONCLUSION

In general, an individual has unique writing skills. Every person, under the influence of various factors, develops skills in drawing alphabets in writing. This manifests itself in the manner of writing - in the manner of execution of a letter, which in its totality of features of the outline becomes unique.

In this work, we considered full page handwritten Kazakh text recognition. To achieve this objective, we developed and used the state-of-the-art CNN based architecture. The model closely follows [8]. To train the model and test its accuracy we used a newly developed dataset of handwritten text images. The findings show that the proposed algorithm has a CER of 0.11 on Test_1 and 0.13 on Test_2.

We note that in the very recent work [2], dual Kazakh and Russian handwritten text recognition problem was studied using architecture similar to our model to certain extent. The model was reported to show a promising progress with CER of 0.045 and 0.064 for the newly developed database of the authors. However, we note that they do not consider the line follower as it is given in their text images improving the accuracy while our dataset does not provide such information. This hints that improving the line follower part of our proposed algorithm may improve the performance.

While the reported CER can be considered as a good result given the moderate size of the dataset, clearly it is worse than the human level accuracy. This limitation is due to the size of our KHD due to computational cost. The general convention in the deep learning community is that the larger the dataset the better model performs. So, it is more likely that the performance of the algorithm improves with the increase of the handwritten text images dataset. It is also interesting to see how it performs for other minority Cyrillic and non-Cyrillic languages.

We note that in certain related articles, both character error rates and word error rates were reported as a metric to evaluate the proposed model performance. Word error rate is the word analogue of character error rate reviewed in the previous section, which roughly tries to find the percentage of correctly identified words. It is no surprise that usually word error rate is worse than character error rate. Since our proposed algorithm tries to predict individual characters instead of words, we found it appropriate to only report character error rates for our tests. One possible future direction is to develop a novel algorithm that considers word-based predictions as opposed to character-based predictions. To train the network for character recognition problem, one needs a dataset containing many versions of characters/letters. For example, Kazakh alphabet contains 42 letters and hence as far as the data contains many example for each letter, the model should work fine. However, in any particular language, the number of words is way higher than the number of letter in the alphabet. Hence, to train the network for word recognition, one expects a very large amount of data available which is already a challenge especially for less studied minority languages. Thus, developing new methods to improve word recognition with

fewer amounts of data would be much appreciated.

Another limitation of the current work is lack of comparison. This is because there is almost no research done in this direction in Kazakh full text recognition. Moreover, since we used our own dataset, there was no chance to compare other works done in the same dataset as the performance results in such experiments are highly dataset sensitive.

In conclusion, the proposed algorithm achieves promising results. As pointed out in the introduction, see Table 1, the Cyrillic scripts have confusing nature in terms of letter recognitions. So, simply because one state-of-the-art algorithm is performing well in one particular language does not guarantee the same performance in another one. Hence, the language specifics should be taken in consideration when building a text recognition model.

REFERENCES

- [1] A. Ul-Hasan and T. M. Breuel, "Can we build language-independent OCR using LSTM networks?," 2013, doi: 10.1145/2505377.2505394.
- [2] A. Abdallah, M. Hamada, and D. Nurseitov, "Attention-based Fully Gated CNN-BGRU for Russian Handwritten Text," *arXiv Prepr. arXiv2008.05373*, 2020.
- [3] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): a comprehensive systematic literature review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020.
- [4] M. Amirgaliyev, B. Yeleussinov, A. Taizo, "Kazakh handwritten recognition," *J. Theor. Appl. Inf. Technol.*, vol. 98, no. 14, pp. 2744–2754, 2020.
- [5] M. Husnain *et al.*, "Recognition of urdu handwritten characters using convolutional neural network," *Appl. Sci.*, vol. 9, no. 13, p. 2758, 2019.
- [6] A. Vinciarelli, "A survey on off-line Cursive Word Recognition," *Pattern Recognit.*, 2002, doi: 10.1016/S0031-3203(01)00129-7.
- [7] R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal, "Automatic processing of handwritten bank cheque images: A survey," *Int. J. Doc. Anal. Recognit.*, 2012, doi: 10.1007/s10032-011-0170-8.
- [8] C. Wigington, C. Tensmeyer, B. Davis, W. Barrett, B. Price, and S. Cohen, "Start, follow, read: End-to-end full-page handwriting recognition," 2018, doi: 10.1007/978-3-030-01231-1_23.
- [9] M. Carbonell, J. Mas, M. Villegas, A. Fornes, and J. Lladós, "End-to-End Handwritten Text Detection and Transcription in Full Pages," 2019, doi: 10.1109/icdarw.2019.40077.
- [10] M. Yousef and T. E. Bishop, "OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by learning to unfold," 2020, doi: 10.1109/CVPR42600.2020.01472.
- [11] J. A. Sanchez, V. Romero, A. H. Toselli, M. Villegas, and E. Vidal, "ICDAR2017 Competition on Handwritten Text Recognition on the READ Dataset," 2017, doi: 10.1109/ICDAR.2017.226.
- [12] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," 2016, doi: 10.1007/978-3-319-46484-8_4.
- [13] T. Plötz and G. A. Fink, "Markov models for offline handwriting recognition: A survey," *Int. J. Doc. Anal. Recognit.*, 2009, doi: 10.1007/s10032-009-0098-4.
- [14] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," 2009.
- [15] A. Antonacopoulos and D. Karatzas, "Document Image Analysis for World War II Personal Records," 2004, doi: 10.1109/DIAL.2004.1263263.
- [16] C. A. Boiangiu, R. Ioanitescu, and M. C. Tanase, "Handwritten documents text line segmentation based on information energy," *Int. J. Comput. Commun. Control*, 2014, doi: 10.15837/ijccc.2014.1.160.
- [17] W. Wang *et al.*, "Shape robust text detection with progressive scale expansion network," 2019, doi: 10.1109/CVPR.2019.00956.
- [18] B. Mor, S. Garhwal, and A. Kumar, "A Systematic Review of Hidden Markov Models and Their Applications," *Arch. Comput. METHODS Eng.*, 2020.
- [19] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009, doi: 10.1109/TPAMI.2008.137.
- [20] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal

- classification: Labelling unsegmented sequence data with recurrent neural networks,” 2006, doi: 10.1145/1143844.1143891.
- [21] H. Li and W. Wang, “Reinterpreting CTC training as iterative fitting,” *Pattern Recognit.*, vol. 105, p. 107392, 2020.
- [22] F. Lombardi and S. Marinai, “Deep Learning for Historical Document Analysis and Recognition—A Survey,” *J. Imaging*, vol. 6, no. 10, p. 110, 2020.
- [23] B. Moysset, C. Kermorvant, and C. Wolf, “Full-Page Text Recognition: Learning Where to Start and When to Stop,” 2017, doi: 10.1109/ICDAR.2017.147.
- [24] W. Sui, Q. Zhang, J. Yang, and W. Chu, “A novel integrated framework for learning both text detection and recognition,” in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 2233–2238.
- [25] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.
- [26] C. Wigington, S. Stewart, B. Davis, B. Barrett, B. Price, and S. Cohen, “Data Augmentation for Recognition of Handwritten Words and Lines Using a CNN-LSTM Network,” 2017, doi: 10.1109/ICDAR.2017.110.
- [27] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, 1966, vol. 10, no. 8, pp. 707–710.