Handwritten Paragraph Recognition Using Spatial Information on Russian Notebooks Dataset

Samah Mohammed
ITMO University
St. Petersburg, Russia
samah369@mail.ru

Nikolay Teslya
SPC RAS
St. Petersburg, Russia
teslya@iias.spb.su

Abstract— **Handwritten paragraph recognition is a vital aspect of handwritten document analysis, enhancing accuracy and usability across various applications. However, recognizing paragraphs in handwritten documents is challenging due to layout variations and irregularities. Spatial information, encompassing spatial relationships between text elements, is essential for accurate paragraph segmentation and document comprehension. Recent works in handwritten Russian recognition have primarily focused on character and line-level recognition. This study is the first attempt on paragraph-level recognition for Russian handwriting, utilizing the Vertical Attention Network (VAN) with a hybrid attention method. Key contributions include the preparation of a unique Russian dataset at the paragraph level, containing around 2600 images with PAGE XML-encoded ground truth. The VAN model was fine-tuned for whole paragraph recognition, and comprehensive experiments were conducted, comparing its performance against alternative non-layout-aware approaches. This work advances layout-aware recognition in handwritten Russian documents, addressing an unexplored area in the field.**

I. INTRODUCTION

Layout-aware handwritten paragraph recognition is a critical component in the field of handwritten document analysis. By leveraging the spatial layout and structure of paragraphs, this technology enhances the accuracy, coherence, and usability of handwritten text recognition systems, making them more effective in capturing the content and context of handwritten documents. The applications of layout-aware recognition span various domains, including document digitization [1], information retrieval [2], document understanding [3], historical preservation, and accessibility [4], making it an indispensable tool in modern document analysis and recognition tasks.

Nevertheless, recognizing paragraphs in handwritten documents poses several challenges due to variations in layout, line spacing, and textual arrangements. Challenges such as overlapping text, cursive writing, ambiguous indentation, writer-specific variations, and irregular line spacing, when the lines are closer together or farther apart which can lead to difficulties in distinguishing between lines within a paragraph. These challenges make the task more complex compared to recognizing machine-printed text.

Spatial information plays a crucial role in understanding the layout and structure of handwritten paragraphs. It refers to the spatial relationships between text elements, such as lines, words, and characters, within a paragraph and across the entire document.it provides important cues for accurately segmenting paragraphs within a handwritten document. By analyzing the vertical and horizontal distances between lines and characters. Also, it is essential for comprehending the overall layout of a handwritten document.

Recent works to recognize the handwritten Russian data, have applied recognition at character level. Later, recognition was applied at line level [5], [6]. To our knowledge, this is the first attempt to recognize the Russian writings at paragraph level. For this purpose, we conduct experiments on the Vertical Attention Network [7]; the free-segmentation end-to-end model that processes the handwriting paragraphs using hybrid attention method.

In brief, we made the following contributions:

1) We prepare the first Russian dataset at paragraph-level; the dataset contains about 2600 images of school notebooks with handwritten notes in Russian where the ground truth is encoded in the PAGE XML format that provide a detailed explanation of layout regions.

2) We fine-tune the Vertical Attention Network (VAN) to recognize whole paragraphs on Russian notebooks dataset.

3) We conduct experiments and present a comparative analysis of the performance of the VAN model on Russian notebooks dataset against alternative approaches that doesn't rely on layout-aware recognition.

The rest of the paper is organized as follows. In section II we the related work on handwritten text recognition. In section III we explain the dataset preparation methodology, the methodology for conducting experiments on the prepared dataset is presented in section IV. We present our experiments and provide discussion in section V. Finally, the conclusion is presented.

II. RELATED WORK

First handwritten recognition models focused on recognizing individual text elements, such as characters or words [8], without considering the spatial relationships between them. It processes raw images or sequences of text elements independently, often using an explicit word/line segmentation. These models are simpler and computationally efficient, making them suitable for tasks like character recognition [9] or word spotting [10]. However, they may struggle with handling complex document layouts and variations in line spacing, leading to potential inaccuracies in recognizing paragraphs within handwritten documents. Recently, the focus has been directed on leveraging spatial information for handwritten paragraph segmentation and recognition.

Projection-based methods can be classified as the earliest segmentation techniques that utilize vertical or horizontal projections to analyze the spatial distribution of text lines in a document. For paragraph extraction, vertical projection profiles [9] are often used. The method involves computing the vertical projection histogram, which represents the density of black pixels along each column of the document image. Another segmentation technique is Region-based segmentation methods [10]; it divides the document image into regions based on layout properties and spatial information. The process involves analyzing the line spacing, line alignment, and character density in different regions of the document. After dividing the document into regions, the spatial properties of each region are analyzed to determine paragraph boundaries. Regions with closely spaced lines are grouped into the same paragraph, while regions with significant vertical gaps are considered as separate paragraphs. These methods are valuable tools in layout analysis and paragraph extraction from handwritten documents. The choice of method depends on the complexity of the document layout and the specific requirements of the recognition task.

Deep learning approaches are now emerging as powerful alternatives, capable of leveraging spatial information for more accurate and comprehensive paragraph recognition. Convolutional Neural Networks (CNNs) are powerful for image feature extraction.

Dhsegment [11] which follows the deep residual network ResNet-50, is proposed for page extraction, baseline detection, document layout analysis and ornament detection on historical documents. A Multimodal-FCN(MFCN) introduced in [12] for document semantic structure, it uses both textual and visual information. In [13], a CNN+MDLSTM is used to predict start-of-line references and an MDLSTM is used for text lines recognition with a dedicated end-of-line token. This is particularly useful in the context of multi-column texts. In [14], a VGG-11-based CNN is used as start-of-line predictor. Then, a recurrent process predicts the next position based on the current one until the end of the line, generating a normalized line. CNN+BLSTM is finally used as OCR on lines.

To incorporate spatial information, attention mechanisms can be integrated into CNNs [15][16]; these attention mechanisms allow the model to focus on relevant spatial regions within the document image. By attending to specific areas related to line spacing, line alignment, and paragraph boundaries, the model can make more informed decisions during paragraph segmentation.

Transformers have demonstrated state-of-the-art performance in various NLP and computer vision tasks. LayoutLMv2 [17] is a pre-trained language model that incorporates both textual and spatial information. It uses a combination of mask language modeling and object detection tasks during pre-training to understand the layout and structure of documents, including paragraphs.

For instance, the first End-to-End Document Image Segmentation Transformer (DocSegTr) introduced in [18], this model shows high performance with overlapped layout objects but doesn't manifest many improvements for smaller regions. The self-supervised pre-training for Document image Transformer proposed in [19], the model is pre-trained with large scale unlabeled document images where each document image is divided into nonoverlapping patches before passing it into a stack of transformers.

By employing attention mechanisms, deep learning models can effectively capture spatial dependencies and encode layout information. This enables them to better understand the structure and organization of handwritten paragraphs, leading to more accurate and coherent layout-aware paragraph recognition and improved performance in various document analysis and understanding tasks. VAN [7] is a specialized deep learning model designed to capture vertical relationships between text lines. It leverages vertical attention mechanisms to identify paragraph boundaries based on vertical cues.

III .DATASET

Since there aren't available datasets to recognize the Russian handwriting at paragraph level and all the available datasets exists at word level as in Russian handwritten notebooks dataset [20] or line level as in "Digital Peter" dataset [5], we decided to prepare the first Russian handwritten dataset at paragraph level.

A.Dataset Description

For dataset preparing the Russian handwritten notebooks' dataset was chosen as the initial point. The data set contains handwritten texts from student notebooks in modern Russian, which makes it easier to check the quality of recognition and pages markup. The dataset contains 1557 images for training, 150 images for validation and 150 images for testing, the available annotation in COCO format.

The provided ground truth for Russian handwritten notebooks' dataset has the following dictionaries in JSON format:

1) Categories: a list of dictionaries with a category's information (category names and indexes). The category names include the next fields: (pupil text, pupil comment, teacher comment, paper, text line.)

2) Images: a list of dictionaries with a description of images, each dictionary must contain the following fields:

a. file_name: name of the image file.
b. id: for image id.
c. Height and width of the image.

3) Annotations: a list of dictionaries with markup information. Each dictionary stores a description for one polygon from the dataset, and must contain the following fields:

a. image_id: the index of the image on which the polygon is located.
b. category_id: the polygon's category index.
c. attributes: text annotation for the word, where each word in each text line is surrounded by a polygon and to define the text lines, we will work on concatenating the polygons of all the words in the same text line.

B. Dataset Preparation

The dataset was prepared automatically using script on Python. We read the ground truth information from Russian Notebooks dataset at word level, analyzes, and process it, and generate the ground truth of the paragraph level dataset. We can divide the process of preparing the handwritten Russian notebooks at paragraph level into 3 main steps:

1) *Defining text lines.* As mentioned before, the baseline points for every line in the dataset and the polygon surrounding each word are defined. The baseline points can be defined as fixed points of each line that can be used for comparison purposes with other lines. Each baseline point is necessarily contained within a word's polygon. So, we define the words of the line as the set of words that their polygon contains one of the baseline points of that line. By sorting the words in each line along coordinate x (horizontally), the text of the line is generated by concatenating the annotation text of each word. The coordinates of the line's polygon are generated by merging the polygons of the line's words in the following way:

a. For each two consecutive polygons, we find the most two right points of the left polygon and the most two left points of the right polygon.
b. The line that connects each two points of them is removed.
c. From the points defined in step a, we connect the upper point and lower point of the left polygon to the upper point and the lower point of the right polygon respectively as shown in Fig. 1.

Fig. 1. An example of Russian text line with associated baseline points.

2) *Splitting to pages.* The polygon coordinates of the page are defined. Line belongs to the page if one of its baseline points is contained in the polygon of the page. In some cases, a line from the left page interferes with the right page, so, we don't check whether the line belongs to the right page if it belongs to the left page.

3) *Splitting page to paragraphs.* In general, there are rules for splitting text into paragraphs e.g., a paragraph begins with an indent in the first sentence, but since the dataset is simple exercises written by primary school students who usually don't adhere to such rules for splitting text into paragraphs, we decided to make a simple definition: two lines in the same page belongs to the same paragraph if they share a common horizontal coordinate and the vertical distance between them is less than 5% of the image height as shown in Fig. 2 where each green polygon indicates a text line and each red polygon indicates a paragraph. The coordinates of the paragraph are the coordinates of the convex hull polygon that contains all the coordinates of the paragraph's lines. Convex hull polygon for a set of points is the smallest convex polygon that contains all the points. The convex hull polygon is used instead of rectangle to reduce the overlapping between paragraphs' coordinates.

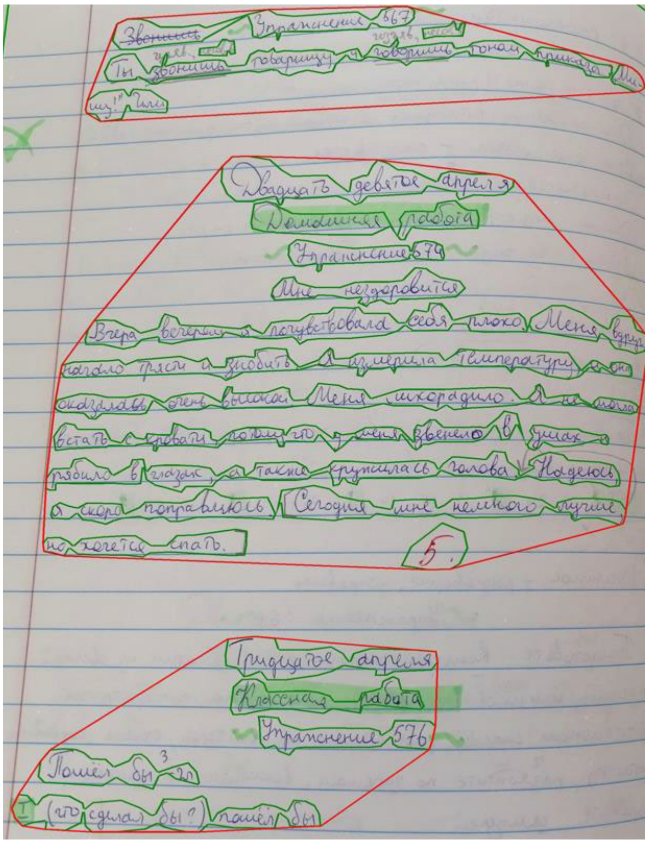

Fig. 2. An example of Russian notebook with associated paragraphs and baseline points for each text line.

IV Experimental methodology

In this section, we explain the experimental setup, including the selection of deep learning model and the evaluation metrics used to assess the performance of the handwritten paragraph recognition model.

A Model Architecture

We opted for the VAN model [7] as it is the first end-to-end encoder-decoder segmentation-free architecture using hybrid attention. The model is designed for joint recognition of text and layout in handwritten documents, with a focus on capturing the spatial relationships between text lines within a paragraph.

We didn't make any changes on VAN architecture. As shown in Fig 3 the VAN takes an input handwritten document image, denoted as $X \in R^{H\times W\times C}$ with H, W, C are the height, width, and number of channels, respectively (C = 3 for an RGB image). This image is pre-processed to extract visual features $f_{2D} \in R^{H_f\times W_f\times C_f}$, where $H_f = \frac{H}{32}$, $W_f = \frac{W}{8}$, and $C_f = 256$ using Fully Convolutional Network (FCN) as it can handle images from different sizes.

The core element of the Vertical Attention Network (VAN) is its vertical attention mechanism, which aims to iteratively generate text line representations l_t in a designated reading sequence, such as from the top to the bottom of the document.

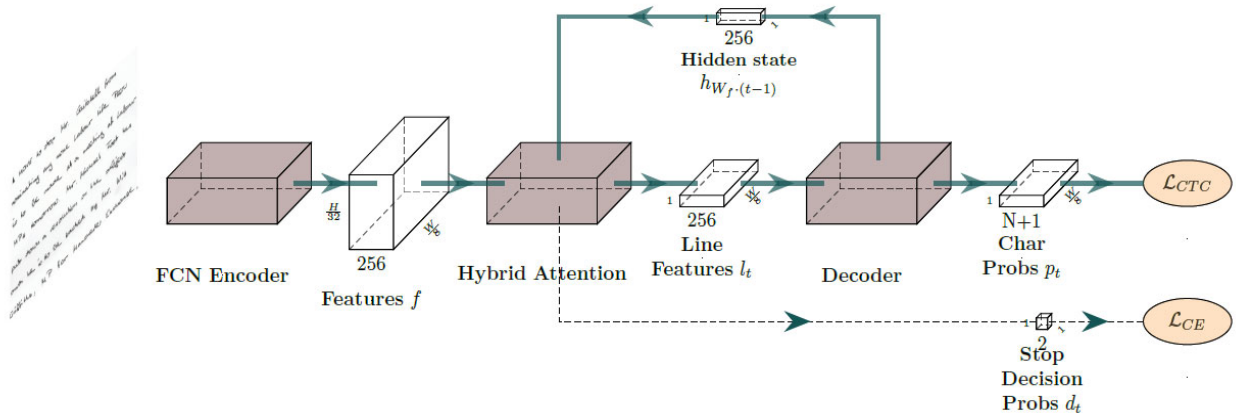

Fig. 3. Vertical Attention Network architecture[7].

Furthermore, the mechanism determines where to stop the generation of a new line representation, indicating the end of a paragraph. This process involves employing a hybrid attention approach, where attention weights are calculated based on both the textual content and the positional information of the output layers. Within the VAN model, the calculation of attention weights combines three elements:

- $f \in R^{H_f \times W_f \times C_f}$: the features in which we want to select specific line features.
- $a_{t-1} \in R^{H_f}$: the previous attention weights. To determine which line to choose, it is necessary to ascertain the lines that have already undergone processing. This is the location-based part.
- $h_{W_{f}(t-1)} \in R^{C_h}$: the decoder hidden state after processing the previous text line (Each line features comprise frames). It encompasses details about the recognized content so far. This is the content-based part.

Since f, a_{t-1} and $h_{W_f(t-1)}$ cannot be combined directly due to dimension mismatching, the horizontal axis of the features must be collapsed.

A learned-stop strategy is introduced within the VAN model to identify the paragraph's end. This method involves training the model to decide when to stop the recognition of a new text line after processing the entire paragraph.

At every iteration, the model calculates a decision probability d_t to assess whether probability p_t should be considered or disregarded. Through this approach, the model undergoes a series of iterations $L + 1$ to acquire the ability to predict the end of the process. This probability p_t is derived from two components: the multi-scale information s_t, which imparts visual information and what remains to be decoded, s_t is computed for each row as shown in (1).

$$s_{t,i} = \tanh\,(W_f . f_i^{'} + W_j . j_{t,i} + W_h . h_{W_f(t-1)}), \qquad (1)$$

where W_f, W_j, W_h are weights of densely connected layers, f' the remaining vertical representation after collapsing the horizontal axis, and $h_{W_f(t-1)}$ the decoder hidden state. The second essential component for calculating the probability p_t is the hidden state of the decoder $h_{W_f(t-1)}$, which potentially holds information regarding the recognized content. For instance, if the last identified character is a dot, there's a higher likelihood of it indicating the paragraph's end.

To match the dimensions, the vertical dimension of $s_{t,i}$ is collapsed and then combined with $h_{W_f(t-1)}$ through a concatenation the channel axis, leading to b_t.

Finally, the dimension is reduced to 2 to compute the probabilities $d_t \in R^2$ through a densely connected layer of weights $W_d \in R^{(C_f + C_h) \times 2}$ as shown in (2):

$$d_t = W_d . b_t \qquad (2)$$

This approach leads to the addition of a cross-entropy loss to the CTC loss, applied to the decision probabilities d_t. The corresponding ground truth δ_t is one-hot encoded, deduced from the line breaks. The cross-entropy loss is defined as follows (Eq. 3):

$$\mathcal{L}_{CE}(d_t, \delta_t) = -\sum_{i=1}^{2} \delta_{t_i} \log d_{\,t_i} \qquad (3)$$

The final loss is then as shown in (4):

$$\mathcal{L}_{ls} = -\sum_{k=1}^{L} \mathcal{L}_{CTC}(p_k, y_k) + \lambda \sum_{k=1}^{L+1} \mathcal{L}_{CE}(d_k, \delta_k), \qquad (4)$$

where λ is set to 1.

B. Metrics

The performance of text recognition is evaluated using the Character Error Rate (CER) as shown in (5), CER is the most common metric to evaluate the text recognition approaches [21]. It is the sum of levenshtein distance (d_{lev}) among the ground truth y^{text} and the predictions $\hat{y}^{text}$ normalized by the total length of the ground truth $y_{len_i}^{text}$.

$$CER = \frac{\sum_{i=1}^{K} d_{lev}(y^{text}, \hat{y}^{text})}{\sum_{i=1}^{K} y_{len_i}^{text}}, \qquad (5)$$

where d_{lev} is the minimum number of single-character (or word) edits (i.e., insertions, deletions, or substitutions) required to change one word (or sentence) into another word.

Word Error Rate (WER) is also used to evaluate text recognition and it is computed in the same way but at word level where we also consider punctuation characters as words.

V. EXPERIMENTS

The results presented in this section are given of the VAN on Russian notebooks dataset at paragraph level, with pretraining on line images and using the learned-stop strategy.

To our knowledge, all the proposed approaches to recognize Russian handwritten at line level have been conducted on Digital Peter dataset only as shown on the Table I. Thus, there are no results reported in the literature on the any Russian datasets at paragraph level and comparisons with approaches under similar conditions can't be conducted as this work is the first attempt to recognize Russian handwritten dataset at paragraph level.

TABLE I. EVALUATION OF DIGITAL PETER DATASET ON THE TEST SET WITH THE LINE-LEVEL RECOGNITION APPROACHES.

Model	CER%↓	WER%↓
CRNN [5]	7.1%	39.7%
Resnet+BLSTM [6]	2.50%	14.60%

This section is dedicated to the evaluation of the VAN on Russian notebooks dataset at paragraph level, the dataset splits with the associated number of characters are shown in Table II.

TABLE II. DATASET SPLITS IN TRAINING, VALIDATION, TEST SETS WITH ASSOCIATED NUMBER OF CHARACTERS.

Dataset	Level	training	validation	test	charset size
Russian Notebooks dataset	Line	3962	404	391	147
	Paragraph	1344	113	117	

Different images from the dataset are shown in Figure 4. As we can see, the number of lines per image can vary a lot and this exhibits more layout variability such as multi-column text images. From the other hand, the Van is designed and limited to process single-column multi-line text documents with relatively horizontal text lines. The next step would be to focus on processing images with more complex layouts such as multi-column text images.

For all our experiments on VAN model, we use the Adam optimizer with an initial learning rate (lr = 10^{-4}). The model had been trained with mini-batch size of 16 for line-level training and mini-batch size of 4 for training at paragraph level. First, we pretrain the model on hybrid Russian and German (READ2016 [14]) datasets then we use the weights to initialize the training on Russian notebook dataset (the pre-training process is carried out for 7 days while the training process is carried out for 5 days). Pre-training and training are performed on a single GPU RTX 3090 (24 GB). To summary the training process. First, the input images are pre-processed and augmented.

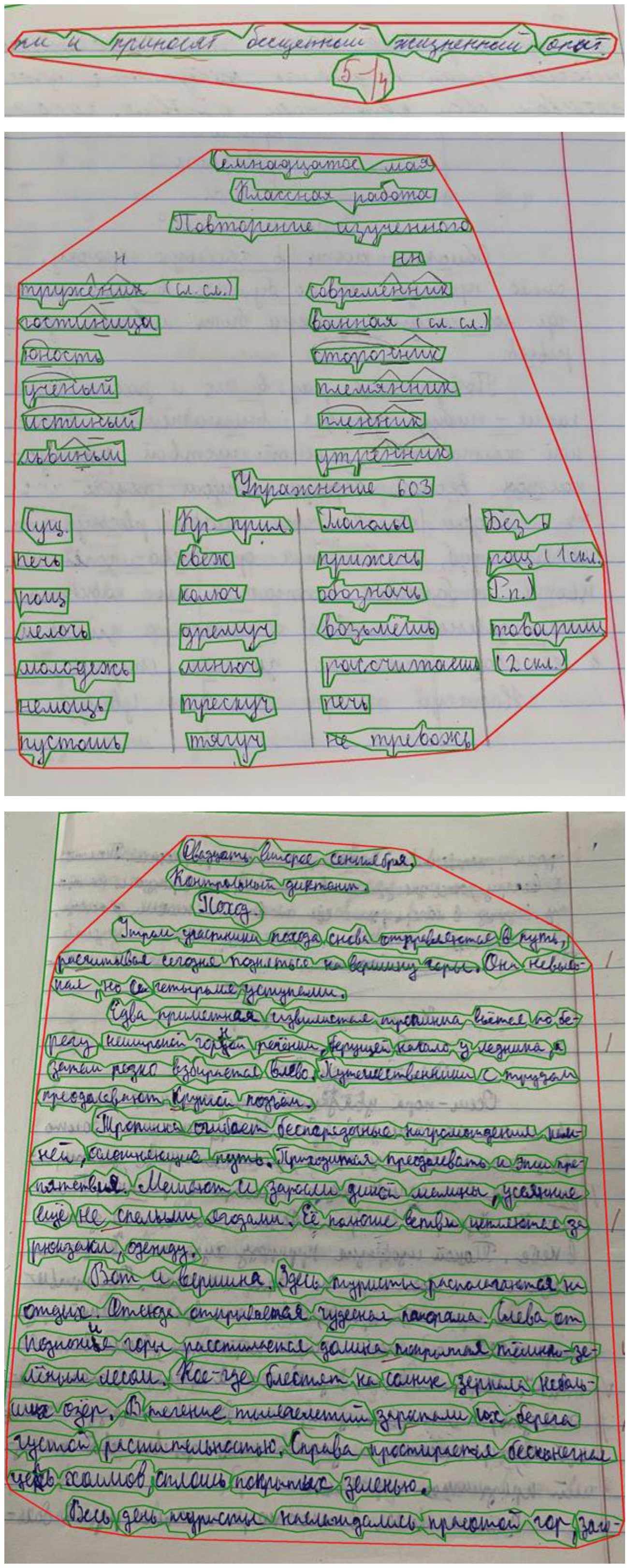

Fig. 4. Samples from Russian notebooks dataset.

Then, the encoder extracts feature map f_{2D} from them. The attention module recurrently generates line features l_t until max number of lines per paragraph $l_{\max}$ is reached or, for the learned-stop approach, until d_t probabilities are in favor of stopping the process. The decoder outputs character probabilities from each line features l_t , which are then decoded through the CTC algorithm, and merged, separated by a whitespace character. We finally get the whole paragraph transcription.

The recognition results on Russian notebooks dataset at paragraph level are presented in the Table III.

TABLE III. RECOGNITION RESULTS OF THE VAN ON RUSSIAN NOTEBOOKS DATASET AT LINE AND PARAGRAPH LEVEL.

Russian notebooks dataset	**CER%↓**	**WER%↓**	**CER%↓**	**WER%↓**
	Test set		**Valid set**	
Line level	7.35 %	20.78%	4.96%	15.59%
Paragraph level	20.87%	38.52%	17.69%	32.61%

We can notice that the values of CER, WER are slightly worse for the line level dataset, and this can be explained by the different complex layouts and irregularity such as multi-column text lines. The challenges in this domain (Paragraph recognition) are substantial, stemming from the variations in layout, line spacing, and textual arrangements within handwritten documents. In future work, we will work on overcoming these challenges.

VI. CONCLUSION

Key contributions of this work include the preparation of a Russian dataset at the paragraph level and fine-tuning the VAN model for whole paragraph recognition. Comprehensive experiments have been conducted at line level and paragraph level. We couldn't reach state-of-the-art recognition results at line level, but the results still indicate promising performance in paragraph recognition, the complexity of handwritten remains a challenge for future research.

In summary, this work pioneers paragraph-level recognition for Russian handwritten documents and lays the foundation for further advancements in layout-aware handwritten paragraph recognition. We hope our work will benefit researchers in the Russian handwritten recognition field at paragraph level.

ACKNOWLEDGMENT

This work was supported by the Russian State Research FFZF-2023-0001.

REFERENCES

[1] A. Corbelli, L. Baraldi, C. Grana, and R. Cucchiara, "Historical document digitization through layout analysis and deep content classification," 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 2016, pp. 4077-4082.

[2] E. Hsu, I. Malagaris, Y.-F. Kuo, R. Sultana, and K. Roberts, "Deep learning-based NLP data pipeline for Ehr-Scanned Document Information Extraction," JAMIA Open, vol. 5, no. 2, 2022.

[3] N. Subramani, A. Matton, M. Greaves, and A. Lam. "A survey of deep learning approaches for ocr and document understanding," arXiv preprint arXiv:2011.13534, 2020.

[4] J. Philips and N. Tabrizi, "Historical document processing: A survey of techniques, tools, and Trends," Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2020, pp. 341–349.

[5] M. Potanin et al., "Digital peter: New dataset, competition and handwriting recognition methods," The 6th International Workshop on Historical Document Imaging and Processing, 2021, pp. 43–48.

[6] A. Shonenkov, D. Karachev, M. Novopoltsev, M. Potanin, and D. Dimitrov, "StackMix and Blot Augmentations for Handwritten Text Recognition," 2021.

[7] D. Coquenet, C. Chatelain, and T. Paquet, "End-to-end handwritten paragraph text recognition using a vertical attention network," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 1,2023, pp. 508–524.

[8] S. Inunganbi, "A systematic review on handwritten document analysis and recognition," Multimedia Tools and Applications, vol. 82, Jun. 2023.

[9] S. Srivastava, A. Verma and S. Sharma, "Optical Character Recognition Techniques: A Review," 2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), BHOPAL, India, 2022, pp. 1-6

[10] L. Kumari and A. Sharma, "A review of deep learning techniques in document image word spotting," Archives of Computational Methods in Engineering, vol. 29, no. 2, 2021, pp. 1085–1106.

[11] S. Ares Oliveira, B. Seguin, and F. Kaplan, "DhSegment: A generic deep-learning approach for document segmentation," 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2018, pp. 7–12.

[12] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. L. Giles, "Learning to extract semantic structure from documents using multimodal fully convolutional neural networks," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, January 2017, pp. 4342–4351.

[13] Grosicki and H. E. Abed, "ICDAR 2011 - French handwriting recognition competition," in International Conference on Document Analysis and Recognition, ICDAR, 2011, pp. 1459–1463.

[14] J. Sánchez, V. Romero, A. H. Toselli, and E. Vidal, "ICFHR2016 competition on handwritten text recognition on the READ dataset," in 15th International Conference on Frontiers in Handwriting Recognition ICFHR, 2016, pp. 630–635.

[15] J. Michael, R. Labahn, T. Gruning, and J. Zollner, "Evaluating sequence-to-sequence models for handwritten text recognition," 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 1286–1293.

[16] J. Jo, H. I. Koo, J. W. Soh, and N. I. Cho, "Handwritten text segmentation via end-to-end learning of Convolutional Neural Networks," Multimedia Tools and Applications, vol. 79, no. 43–44, 2020, pp. 32137–32150.

[17] Y. Xu et al., "LayoutLMv2: Multi-modal pre-training for visually rich document understanding," ACL-IJCNLP 2021 - 59th Annu. Meet. Assoc. Comput. Linguist. 11th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., 2021, pp. 2579–2591.

[18] S. Biswas, A. Banerjee, J. Lladós, and U. Pal, "DocSegTr: An Instance-Level End-to-End Document Image Segmentation Transformer," 2022.

[19] J. Li, Y. Xu, T. Lv, L. Cui, C. Zhang, and F. Wei, DiT: Self-supervised Pre-training for Document Image Transformer, vol. 1, no. 1. Association for Computing Machinery, 2022.

[20] Ai-Forever, "Ai-forever/school_notebooks_ru · datasets at hugging face," ai-forever/school_notebooks_RU · Datasets at Hugging Face, https://huggingface.co/datasets/ai-forever/school_notebooks_RU (accessed Sep. 10, 2023).

[21] N. Teslya and S. Mohammed, "Deep learning for handwriting text recognition: Existing approaches and challenges," 2022 31st Conference of Open Innovations Association (FRUCT), Apr. 2022, pp. 339–346.