Enhancing YOLO Models for Handwritten Text Recognition

Nikita Lomov
HSE University
Moscow, Russia
nikita-lomov@mail.ru

***Abstract*—This paper is devoted to the development of computer vision models capable of solving problems of simultaneous detection and recognition of handwritten text. As a starting point, the YOLOv8 family of architectures for object detection is considered. We formulate line detection as different tasks depending on the text shape: straight lines as object detection, slanted lines as oriented bounding box (OBB) detection, and curved lines as instance segmentation. For each version of the model, a suitable pooling procedure is developed that extracts a feature description within a bounding box or a mask.**

For the instance segmentation problem, a modification of the segmentation mechanism is proposed that takes into account the features of lines as graphic objects and operates on a geometric principle. To recognize the formatting of handwritten text, in particular, to determine strikethrough and underlining, a transition to an extended alphabet is carried out with the prediction of two components—a symbol and its style—separately.

The effectiveness of the developed methods is estimated on real original data—a set of diary pages of the Russian statesman Modest Andreevich Korf (1800-1876), which is a valuable historical source. All models successfully cope with the task and demonstrate a character error rate (CER) of about 3-4%, which makes the recognized text easily readable by a person. At the same time, the quality of recognition increases with increasing complexity of the model, which justifies the consideration of various variants of the problem.

The code is available at https://github.com/nlomov/yolo-htr.

I. Introduction

In recent years, the widespread use of digital technologies and machine learning tools has had a significant impact on the nature of the work of researchers in the humanities. Not only has the digitalization of archives and document collections made it possible to access them remotely, but methods for intelligently analyzing the documents themselves, including searching and navigating them, summarizing and categorizing them, make it possible to cover previously unimaginable volumes of information. Handwriting recognition, aimed at converting a scanned or photographed document image into a textual transcription, is a fundamental challenge in this field.

Several software tools offer complex handwriting recognition functionality, including Kraken OCR [1], Transkribus [2] and OCR4all [3]. They provide broad opportunities for solving problems associated with HTR, such as extracting, binarizing and normalizing text lines, layout analysis, collecting training samples and training your own models. Among the resources focused on recognizing documents in Russian, we can highlight Yandex Vision OCR [4], commercial system with access via API. Also, handwriting recognition systems based on multimodal language models are easy-to-use and offer enormous potential [5].

At the same time, even large datasets do not eliminate the problem of handwriting diversity, which makes it difficult to apply a pre-trained model to previously unseen writing styles. Even greater obstacles are associated with insufficient amounts of data for training in the case of non-trivial languages and alphabets other than Latin. For these reasons, the development of computer vision models that can train on small amounts of data, for example, handwriting samples of a particular person, still seems to be a relevant task. The study [6] shows that when tuning for a specific handwriting, the volume of training data of several hundred lines is insufficient, since stabilization of quality metrics is not achieved, so it is more appropriate to talk about thousands of lines.

Thus, the task of developing handwriting recognition models that are both user-friendly and easy to train remains a pressing issue. The actual survey [7] on the problem considers two groups of approaches: up-to-line level, including word and line recognition, and beyond line-level, concerning paragraph- and document-level challenges. However, the input data is rarely represented by images of lines, so using the methods in the first class requires extracting the strings from the image, and often doing significant preprocessing of those crops. It is natural to encapsulate all subtasks within a single model, which defines our choice in favor of page-level approaches that, owing to advances in computing power, have been attracting increasing research interest.

As an additional task we consider the analysis of handwritten text formatting, in particular the highlighting of underlined and crossed-out fragments, interlinear insertions, and characteristic abbreviations. Formatting features help to trace the progress of work on the manuscript, compare its various versions, record what the author himself paid special attention to, and what he considered generally applicable. For example, such problems were addressed by researchers of the works of Lope de Vega [8] and Charles Dickens [9].

Our work is aimed at creating page models that combine search and recognition of lines of formatted handwritten text, for various formulations of the search problem—such as classical detection, oriented bounding box detection and instance segmentation.

The main achievements of this work are as follows:

- Three main architectures from the YOLOv8 family are modified in such a way as to support text recognition within the corresponding area.
- We propose a new principle of instance segmentation, which works on a geometric principle and copes better with visually similar objects.
- We demonstrate a method for recognizing text formatting that uses an extended alphabet of symbol-style pairs.

II. Related Work

Models aimed at recognizing line-level images are quite well developed and varied in architecture. They can be fully convolutional [10], use multivariate recurrent connections [11], or be based on cutting-edge architectures like Visual Transformer [12].

There are a number of works in which a full-fledged text recognition system at the page level is assembled from several modules, one of which is the extraction of text lines, which allows for end-to-end inference, but not end-to-end training. In particular, the system described in [13] includes not only separate networks for searching and recognizing marginalia, and the second of them contains a learning module based on thin-plate spline for straightening words. The system presented in [14] goes down from the page level to the level of individual words, different neural network architectures are compared to solve subtasks. In [15], the authors combined YOLOv8-driven detector of text regions with a simple RNN-based handwritten text recognition model.

Some models work at the paragraph level, assuming that lines of text are stacked neatly on top of each other and not too warped. To distinguish individual lines, vertical attention [10] or sequential vertical upsampling and horizontal downsampling to collapse the horizontal dimension [16] is used. It is worth noting that models that work directly at the page level are still relatively rare [17]–[19] and are quite difficult to train—for example, they usually involve pre-training on synthetic datasets.

The first successful attempt to implement the end-to-end paragraph handwritten text recognition was undertaken in [20], where text scanning was carried out using the attention mechanism implemented by a multi-dimensional Long Short-Term Memory (LSTM) network. In the Start, Follow, Read method [21] three networks are trained together at once: to search for the starts of lines, to track them with the formation of a straightened image of the line, and to recognize its text.

One way to make the network end-to-end learnable is to connect the output of the detector to the input of the recognizer using some pooling procedure. In [22], a RoI Pooling operation is used, which consists of resampling the detected word from the original image by a grid of a fixed height. In [23], a similar operation is called Text Pooling and extracts crops of feature masks for lines of text, but not words. In [24], features are extracted within the rotated box, and an adversarial feature learning network is used to approximate the distributions of these features and the synthetic image's ones. The [25] uses a spatial transformer network to detect words of non-handwritten text in natural scene images and allows a wider class of affine transformations during resampling.

The tasks of detecting text elements—strings of words and letters (usually in the case of hieroglyphic writing)—can be combined with various recognition tasks. Thus, the work [26] demonstrates an example of simultaneous recognition of the letter itself and its box, for which the CTC loss is modified. In the work [27], the problems of detecting handwritten words, classifying them by types of named entities and direct text recognition are simultaneously solved.

In [28], various types of text formatting, including strikethroughs, underlines, and multi-line text, are detected using a network with an architecture based on the Generative Adversarial Network (GAN). The tasks of formatting recognition are often accompanied by image restoration tasks. Thus, in the work [29] the word detection problem is solved, strikethrough words are presented as a separate class. Then the appearance of the word without strikethrough is restored by a combination of U-Net and Bi-LSTM networks. Search and removal of user marks, such as underlining, is implemented in the work [30] using line slope analysis and morphological operations.

III. Architecture Design

The work [31] proposed the YOLO-HTR architecture, the idea of which is to replace the YOLOv8 network encoder with an encoder from a network designed to recognize text lines. The features obtained by the Vertical Attention Network (VAN) [10] have proven themselves to be good for text recognition. The network encoder consists of 10 consecutive blocks, like the YOLOv8 network encoder, therefore, as in the YOLOv8 network, the outputs of the 5th, 7th and 10th blocks are sent to the head part, which is directly responsible for the regression of the box parameters. Next, the features from the last layer of the encoder were masked by boxes and freed from vertical dimension using a special box pooling procedure.

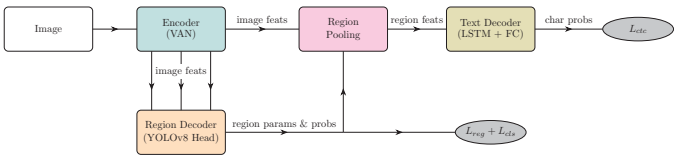


Fig. 1. General scheme of the YOLO-HTR architecture. The region can be a bounding box, an oriented bounding box, or a bounding box with a mask. In the first two cases, L_{reg} is equal to L_{box}, in the last case—to $L_{box} + L_{seg}$. Hidden layer size in LSTM in Text Decoder is 512.

We will follow the general scheme of this approach, shown in Figure 1. Note that in the original version of YOLO-HTR only a network for classical object detection with straight boxes was considered. We will develop networks for oriented bounding boxes' detection and instance segmentation, too, with heads taken from the corresponding YOLO variants. We also need to modify the connection between the encoder and the region decoder in order to better handle lines—objects that are usually wide and low.

Indeed, it should be noted that the text recognition encoder is local and pays attention to a small neighborhood of the pixel, sufficient to recognize an individual character and its neighbors. At the same time, for successful detection of the entire line, the network must “see” it in full width with some neighborhood, and therefore must have a sufficiently wide receptive field. Let us estimate its size for all layers of the VAN encoder and three detection blocks—in small (S), medium (M) and large (L) scales—with a “naive” replacement of the encoder.

TABLE I. THE SIZES OF THE RECEPTIVE FIELD WITH A ”NAIVE” ENCODER REPLACEMENT

Block	Scaled	Original	Block	Scaled	Original
Encoder 1	7 x 7	7 x 7	Encoder 8	32 x 22	256 x 352
Encoder 2	8 x 13	16 x 13	Encoder 9	38 x 28	304 x 448
Encoder 3	8 x 11	32 x 22	Encoder 10	44 x 34	352 x 544
Encoder 4	8 x 10	64 x 40			
Encoder 5	14 x 10	112 x 80	Detector S	80 x 104	640 x 832
Encoder 6	20 x 10	160 x 160	Detector M	53 x 65	848 x 1040
Encoder 7	26 x 16	208 x 256	Detector L	39 x 45	1248 x 1440

Table I presents the receptive field sizes in the downsampled resolution (columns ”Scaled”) and in the original one (”Original”). The results show that the maximum line that the detector can capture is 1248 pixels wide. This may not be enough to describe long lines, especially with random scaling during augmentation. Note that the original YOLOv8 network has field sizes of Detector S/M/L in original pixels equal to 1728, 1728, and 2368, the same in width and height. To expand the network’s field of view, we add separately to each of the layers read out (5th, 7th and 10th) the corresponding layer from the YOLO encoder. As a result, we increase the field sizes of Detector S/M/L to 1248x1440, 1648x1840 and 2080x2272, which is quite enough for training on large images of about 2000 pixels.

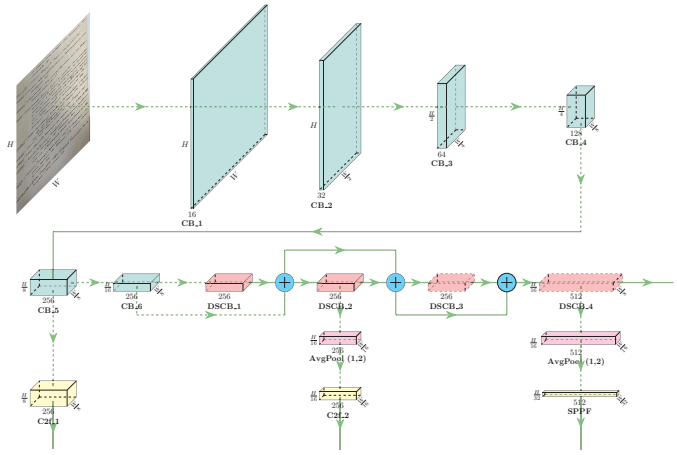


Fig. 2. Encoder of YOLO-HTR+ architecture. The arrows pointing down lead to the region (Box/OBB/Mask) decoder, the right arrow leads to the region pooling.

A detailed diagram of the new YOLO-HTR+ network encoder is shown in Figure 2. Minor modifications were made to the VAN encoder in terms of layer sizes and downsampling rates to enable more successful processing of closely spaced lines based on the results of previous work on YOLO-HTR. Also, the decoder structure in the form of a combination of LSTM and a fully connected (FC) layer was taken from the page-level, rather than the line-level, version of VAN, since it was recognized more suitable for processing complex handwriting.

IV. REGION POOLING PROCEDURE

The aim of the region pooling procedure, which does not have trainable parameters, is to preserve the features of only those points that fall within the predicted box and to eliminate the vertical dimension in preparation for text recognition using CTC loss. The specific type of region pooling operation depends on the region representation—as a straight bounding box, an oriented bounding box, or a straight box with a mask.

The input to the procedure is a set of boxes $\mathbf{b} = \{b_i\}_{i=1}^{m}$ and a feature map $F \in \mathbb{R}^{H' \times W' \times n}$, at the output we have a tensor $G \in \mathbb{R}^{m \times W' \times n}$. Since the procedure is carried out independently by features and boxes, it is sufficient to examine the map of a single feature $F_t \in \mathbb{R}^{H' \times W'}$ and the single box b, the obtained result $F''_t \in \mathbb{R}^{1 \times W'}$ will be a slice of G.

A. Straight Boxes

For straight boxes, the region pooling procedure is described in [31]. Assuming that the box coordinates $b = (x, y, w, h)$ are specified in the scale of the feature map, for each of its $H' \times W'$ cells we determine the fraction of the cell that falls within the box:

$$w_x(j) = 1 - \text{clip}\left(\left(x - \frac{w}{2}\right) - j - 1\right) - \text{clip}\left(j - \left(x + \frac{w}{2}\right)\right), \quad j = 1, \ldots, W',$$

$$w_y(i) = 1 - \text{clip}\left(\left(y - \frac{h}{2}\right) - i - 1\right) - \text{clip}\left(i - \left(y + \frac{h}{2}\right)\right), \quad i = 1, \ldots, H',$$

where $\text{clip}(a) = \min(\max(a, 0), 1)$. Considering that $w_x \in \mathbb{R}^{1 \times W'}$, $w_y \in \mathbb{R}^{H' \times 1}$, the map of the t-th feature F_t masked by the box, can be represented in matrix form $F'_t = (w_y w_x^T) \odot F_t$. For aggregation, we average the values inside the box vertically, noting that the sum of w_y is equal to h in the case when the box fits completely into the image: $F''_t = \frac{1}{h}[1, ..., 1] \times F'_t$.

Developing analogues of the pooling procedure for detection tasks in other formulations will provide a more accurate description of line shapes. This, in turn, will enable the extraction of more relevant text features, leading to improved text recognition quality.

B. Rotated Boxes

In this case, the box is augmented by an angle α: $b = (x, y, w, h, \alpha)$. We define the centers of all cells in the feature map F_t:

$$\bar{x}_j = j - 0.5, \quad \bar{y}_i = i - 0.5,$$
$$C = \{(\bar{x}_i, \bar{y}_j)\}, \; j = 1, \ldots, W', \; i = 1, \ldots, H'.$$

When both the box itself and the points of the set C are rotated by an angle $-\alpha$:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = M \begin{bmatrix} x \\ \beta y \end{bmatrix}, \ \begin{bmatrix} \bar{x}'_{ij} \\ \bar{y}'_{ij} \end{bmatrix} = M \begin{bmatrix} \bar{x}_j \\ \beta \bar{y}_i \end{bmatrix},$$
$$M = \begin{bmatrix} \cos\alpha & \sin\alpha, \\ -\frac{1}{\beta}\sin\alpha & \frac{1}{\beta}\cos\alpha \end{bmatrix},$$

the box $b' = (x', y', w, h, 0)$ becomes straight. The parameter β is needed here to correct different scales along the X and Y axes—in particular, since $W' = \frac{W}{8}$, $H' = \frac{H}{16}$, $\beta = 2$, we need to stretch the map along the Y axis twice for proportionality.

Again, we can determine the fraction of each cell that falls within the box:

$$\begin{aligned} w_x(i,j) &= 1 - \text{clip}\left(\left(x' - \frac{w}{2}\right) - \bar{x}'_{ij} + 0.5\right) + \\ &\quad - \text{clip}\left(\bar{x}'_{ij} + 0.5 - \left(x' + \frac{w}{2}\right)\right), \\ w_y(i,j) &= 1 - \text{clip}\left(\left(y' - \frac{h}{2}\right) - \bar{y}'_{ij} + 0.5\right) + \\ &\quad - \text{clip}\left(\bar{y}'_{ij} + 0.5 - \left(y' + \frac{h}{2}\right)\right), \\ w(i,j) &= w_x(i,j) \cdot w_y(i,j), \\ &\quad i = 1, \ldots, H', \\ &\quad j = 1, \ldots, W'. \end{aligned}$$

The map of a single feature masked by the box is computed as $F'_t = W \odot F_t$, vertical aggregation: $F''_t = \frac{\cos\alpha}{h}[1, ..., 1] \times F'_t$ (due to the rotation, the intersection of the box with the vertical line has length $\frac{h}{\cos\alpha}$, and not just h).

C. Straight Boxes with Mask

The shape of an object in this case is described not only by a straight box $b = (x, y, w, h)$, but also by a mask $P \in [0,1]^{H' \times W'}$ of pixels belonging to the object. Let's crop this mask to the bounding box using the same w_x and w_y as in the case of a straight box:

$$\begin{aligned} &P' = (w_y \times [1, \cdots, 1]) \odot P, \quad F'_t = P' \odot F_t, \\ &F''_t = (F'_t w_x) \oslash ([1, ..., 1] \times P'). \end{aligned} \tag{1}$$

D. Mask Thinning

When constructing ground truth data for boxes, we must keep them high enough so that the cell centers fall within them, and these anchors are assigned to predict boxes. Obviously, the boxes themselves can overlap due to serious line bends and writing density—this will lead to the same features being "blurred" across different objects, although their text is obviously different. To reduce the damage from this, only at the region pooling stage we can consider all boxes to be one pixel high—this means that in each column information will be read from either one pixel or two adjacent ones. So,

- in the case of a straight box h is replaced by 1;
- in the case of a rotated box h is replaced by $\cos\alpha$;
- in the case of a straight box with a mask we calculate average vertical level:

$$\tilde{y} = ([0.5, 1.5, \ldots H' - 0.5] \times P') \oslash ([1, 1, \ldots, 1] \times P')$$

and adjust

$$\tilde{w}_y(i) = 1 - \text{clip}((\tilde{y} - 0.5 - i) - \text{clip}(i + 1 - (\tilde{y} + 0.5)),$$

then recalculating P' and F''_t in 1 with $\tilde{w}_y$ instead of w_y.

V. Parameterization of Rotated Rectangles

A. Dealing with Parameterization Ambiguity

Although different parameterization methods are possible, the conventional bounding box is completely defined by four numbers, for example, the coordinates of its center (x, y), width w, and height h. At first glance, it seems that to define a rotated rectangle, it is enough to add one more parameter to these parameters—the rotation angle α. However, this immediately raises the problem of ambiguity of such a representation, illustrated by Fig. 3: a rotated rectangle can be considered low and wide (height h, width W, $h \ll W$) with a rotation angle $\frac{\pi}{4}$, or high and narrow (height $H = W$, width $h = w$) with a rotation angle $-\frac{\pi}{4}$. At the same time, the Intersection over Union-based box loss will not suffer, since it is determined by comparing the geometric figures themselves, and not their parameterizations, but the choice of parameterization method can be critical for the Distribution Focal Loss (DFL) [32] if we swap disproportionate height and width.

When using DFL, the task of predicting the parameters of the box is set not as a regression problem, but as a soft classification problem. The desired size v is expressed as a weighted sum of the basic sizes $\{v_i = i \cdot d\}_{i=0}^{N}$:

$$v = \sum_{i=0}^{N} w_i v_i, \ 0 \le w_i \le 1, \ \sum_{i=0}^{N} w_i = 1.$$

The idea of DFL is that among all possible sets of weights $\{w_i\}$ that yield the desired v, the optimal one is considered to be a sparse distribution over the two closest to v base sizes v_i and v_{i+1}, $i = \lfloor \frac{v}{d} \rfloor$, $\tilde{w}_i = 1 - \frac{v - v_i}{d}$, $\tilde{w}_{i+1} = \frac{v - v_i}{d}$, $\tilde{w}_j = 0$ if $j \notin \{i, i+1\}$. DFL is defined as

$$l_{dfl}(w_0, \ldots, w_N) = -\tilde{w}_i \ln w_i - \tilde{w}_{i+1} \ln w_{i+1}$$

and attains a minimum at the optimal set $\{\tilde{w}_i\}$. So the value of DFL depends critically on what is considered width and what is considered height.

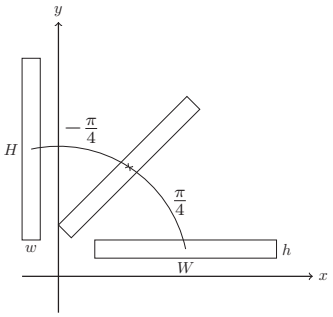


Fig. 3. Ambiguity of parameterization of a rotated rectangle

Fixing the range of angles (it is logical to make it symmetrical with respect to 0, which means no rotation, i.e. a straight line) within $(-\frac{\pi}{4}; \frac{\pi}{4}]$, although it leads to an unambiguous parameterization, does not remove the problem during training: imagine two low wide rectangles, one with a rotation angle of $\frac{\pi}{4} - \varepsilon$, the other with an angle of $\frac{\pi}{4} + \varepsilon$. Then the first of them will receive a parameterization $(x, y, w, h, \frac{\pi}{4} - \varepsilon)$, and the second $(x, y, h, w, -\frac{\pi}{2} + \varepsilon)$, while visually they will be almost indistinguishable. This means that our loss function will be discontinuous in the input, which is the image pixels.

A possible solution here is to consider both parameterizations as acceptable, both with a positive and negative angle:

$$\bar{l}_{dfl}(b) = \min\left(l_{dfl}(\cdot|w,h), l_{dfl}(\cdot|h,w)\right).$$

An even finer correction of DFL allows us to pessimize the alternative parameterization at α values close to 0:

$$\hat{l}_{dfl}(b) = q(\alpha) \cdot l_{dfl}(\cdot|w,h) + \\ + (1 - q(\alpha)) \cdot \min\left(l_{dfl}(\cdot|w,h), l_{dfl}(\cdot|h,w)\right).$$

In this case, $q(\alpha)$ should be equal to 1 when $\alpha = 0$ and equal to 0 when $\alpha = \pm\frac{\pi}{4}$. The function $q(\alpha) = \cos^2(2\alpha)$ is suitable for this.

B. Transition between Parameterizations

Recall that in DFL we predict distances from the anchor $(\bar{x}, \bar{y})$ to the sides of the rectangle. Let us generalize the necessary formulas to the case of a rotated rectangle specified by the parameters (x, y, w, h, α). When rotating by $-\alpha$, we move to the case of a straight box with the anchor

$$\begin{bmatrix}\bar{x}' \\ \bar{y}'\end{bmatrix} = M \begin{bmatrix}\bar{x} \\ \bar{y}\end{bmatrix}$$

and the center of the box

$$\begin{bmatrix}x' \\ y'\end{bmatrix} = M \begin{bmatrix}x \\ y\end{bmatrix},$$

where

$$M = \begin{bmatrix}\cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha\end{bmatrix}.^{1}$$

Assuming that

$$d'_x = x' - \bar{x}',\ d'_y = y' - \bar{y}',$$

we need to predict the values, shown in Fig. 4:

$$l = \frac{w}{2} - d'_x,\ r = \frac{w}{2} + d'_x,\ t = \frac{h}{2} - d'_y,\ b = \frac{w}{2} + d'_y.$$

Finally, we will also need the inverse transformation from (t, l, b, r) to to decode the prediction results using DFL:

$$w = l + r,\ h = t + b,\ d'_x = \frac{r - l}{2},\ d'_y = \frac{b - t}{2},$$

$$\begin{bmatrix}x \\ y\end{bmatrix} = \begin{bmatrix}\bar{x} \\ \bar{y}\end{bmatrix} + M^T \cdot \begin{bmatrix}d'_x \\ d'_y\end{bmatrix}.$$

[1]Note that in the original implementation of YOLOv8 from Ultralytics a serious mistake is made—this rotation is simply not performed, and it is assumed that $d_x = x - \bar{x}$, $d_y = y - \bar{y}$. This is the reason for the very unconvincing quality of the rotated box detectors even in tasks that do not look particularly complex, an example can be seen in figures (a)-(d) on the page https://docs.ultralytics.com/ru/datasets/obb/#yolo-obb-format

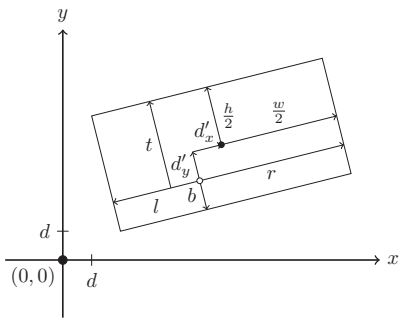


Fig. 4. Relationship between (t, l, b, r) and (x, y, w, h) parametrizations in an oriented bounding box. The filled dot is the center of the box, the unfilled dot is the anchor.

VI. Instance Segmentation by Center Prediction

A. Segmentation Principle

In the classic version of YOLOv8, in the instance segmentation task, not only a bounding box coordinates are predicted for each object, but also a feature description Q of length 32. The same feature description K must be obtained for each downsampled pixel of the image. To assess the correspondence of a pixel falling within the box to its object, the value is calculated: $v = \text{sigmoid}(Q^T K)$, this method of assessment resembles the well-known attention mechanism. Note that if a pixel belongs to a unique mask but falls within the boxes of two close objects, their features Q_1 and Q_2 must differ drastically from each other so that $Q_1^T K$ and $Q_2^T K$ have opposite signs. However, it is difficult to single out any visual characteristics that fundamentally distinguish one line of text from another, since they all represent a strip of contrasting thin strokes.

For more successful segmentation, we will completely change the principle of assessing pixels to objects and will predict for each pixel (x, y) its offset to the corresponding center of the object, again in the form of soft classification:

$$v = \sum_{i=-N}^{N} w_i \cdot v_i, \quad v_i = di, \quad \sum_{i=-N}^{N} w_i = 1,$$

where v is the offset along x or y. Note that, unlike the distances to the edge of the box considered earlier, these values can be both positive and negative. Also, for each pixel its objectness p is calculated. As a result, we do not calculate keys K at all, and change the length of Q to $1 + 2(2N + 1)$ while maintaining the rest of the architecture.

The pixel is considered to belong to the object with the center (x, y) and the box b if:

- it falls within b,
- $p > t$,
- there is no object with a closer center in terms of the metric: $(\frac{\bar{x} + d_x - x}{\beta})^2 + (\bar{y} + d_y - y)^2$.

Thus, each pixel is tied to no more than one object. The β parameter is needed for more accurate processing of horizontally elongated objects: the deviation along y is considered more critical, and β is set equal to 10.

The proposed approach is illustrated by Figure 5 and is based on the work of [33].

Fig. 5. Instance segmentation mechanism. Black and white background reflects the visibility p of pixels, lines show the offset vectors (d_x, d_y), dark dots show predicted box centers for different pixels

B. Loss Function

Based on the segmentation principle, the loss function should reflect

- whether an image pixel belongs to an object (denote the union of all object masks by F) or to the background, for which binary cross-entropy is used:

$$L_{fg} = -\frac{1}{WH}\left(\sum_{(\bar{x},\bar{y})\in F} \ln p(\bar{x},\bar{y}) + \right.$$
$$\left. + \sum_{(\bar{x},\bar{y})\notin F} \ln(1-p(\bar{x},\bar{y}))\right),$$

- accuracy of predicting the box center (x, y) using the mean square error:

$$L_{dist} = \frac{1}{|F|}\sum_{(\bar{x},\bar{y})\in F}\left(\frac{\bar{x}+d_x-x(\bar{x},\bar{y})}{\beta}\right)^2 +$$
$$+ (\bar{y}+d_y-y(\bar{x},\bar{y}))^2,$$

- also for a more concentrated prediction of offsets to the center, the DFL considered earlier is used:

$$L_{dfl} = \frac{1}{\beta^2|F|}\sum_{(\bar{x},\bar{y})\in F} l_{dfl}(w^{(x)}_{-N}(\bar{x},\bar{y}),\dots,w^{(x)}_{N}(\bar{x},\bar{y})) +$$
$$+ \frac{1}{|F|}\sum_{(\bar{x},\bar{y})\in F} l_{dfl}(w^{(y)}_{-N}(\bar{x},\bar{y}),\dots,w^{(y)}_{N}(\bar{x},\bar{y}))\Big).$$

VII. Formatting Recognition

Let us assume that we are recognizing not only a symbol from the alphabet A, but also its style from the set S. In this case, the symbol and the style are recognized independently, i.e. $p(a, s) = p(a)p(s)$, $a \in A$, $s \in S$. For this, the output of the last layer V before feeding into the CTC loss has the dimension $|A|+|S|+1$, the first $|A|+1$ values is responsible for the symbol, including the empty symbol ε, the last $|S|$— for its style. We transform these values to get the probability of the combination:

$$\begin{aligned} w_i &= \log \operatorname{softmax}(v_i \mid v_1,\dots,v_{|A|+1}),\\ i &= 1,\dots,|A|+1,\\ u_j &= \log \operatorname{softmax}(v_j \mid v_{|A|+2},\dots,v_{|A|+|S|+1}),\\ j &= |A|+2,\dots,|A|+|S|+1,\\ p_t &= w_{\text{mod}(t-1,|A|)+1} + u_{\text{div}(t-1,|A|)+1},\\ t &= 1,\dots,|A|\cdot|S|,\\ p_{|A|\cdot|S|+1} &= w_{|A|+1}, \end{aligned}$$

that is, CTC loss is calculated in the alphabet of $|A| \cdot |S| + 1$ "symbols-styles" using the matrix P, an empty symbol is not divided into styles. This approach allows predicting with CTC loss even those symbols that were not present in the required style in the training sample, since the symbol and style are predicted separately.

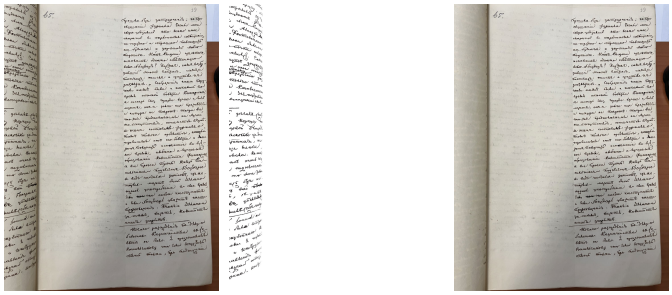
VIII. Experiments

A. Data preparation

The initial data are the diaries of Modest Andreevich Korf (1800–1876), one of the outstanding representatives of Russian conservative thought in the mid-19th century, a member of the State Council, senator, and actual privy councilor. Korf's diary archives are extremely extensive and contain about 9,000 pages. Of particular interest is the excellent systematization of the diary: each volume is preceded by an alphabetical index listing the entities mentioned—persons, places, events—with corresponding page numbers. The bulk of the entries are in a single format—the text is given in a narrow column on the right half of the page, the left is used for notes. Text transcriptions for the 100 pages of the third volume, dating back to 1840, were restored from the edition [34]. The text was broken into lines, a total of 4532, and checked for assignment to bounding boxes by an expert. Also, for a more stable allocation of boxes in the training, 40 pages with an alphabetical index of 1305 lines without a text transcript from the 1st, 2nd and 3rd volumes were used. The test and validation samples included 20 images each from different volumes.

To avoid fragments of lines from other pages in the frame that do not have markup, these fragments were colored in the background color, for which segmentation into an object and background was carried out using Kraken OCR [1]. An example of such a correction is shown in Figure 6.

To describe the shape of the lines, the same Kraken OCR was used to extract the baselines of the text with subsequent



(a) (b) (c)

Fig. 6. Removing unnecessary text fragments. (a) Original page, (b) mask of redundant fragments, (c) corrected page.

manual correction. Let the baseline b be given as a polyline $\{(x_i, y_i)\}_{i=1}^{n}$. Let us define the corresponding point of the upper and lower envelope for each vertex:

$$x_i^{(t)} = x_i - d_t \cos\alpha_i,\ y_i^{(t)} = y_i - d_t \sin\alpha_i,$$
$$x_i^{(b)} = x_i + d_b \cos\alpha_i,\ y_i^{(b)} = y_i + d_b \sin\alpha_i,$$

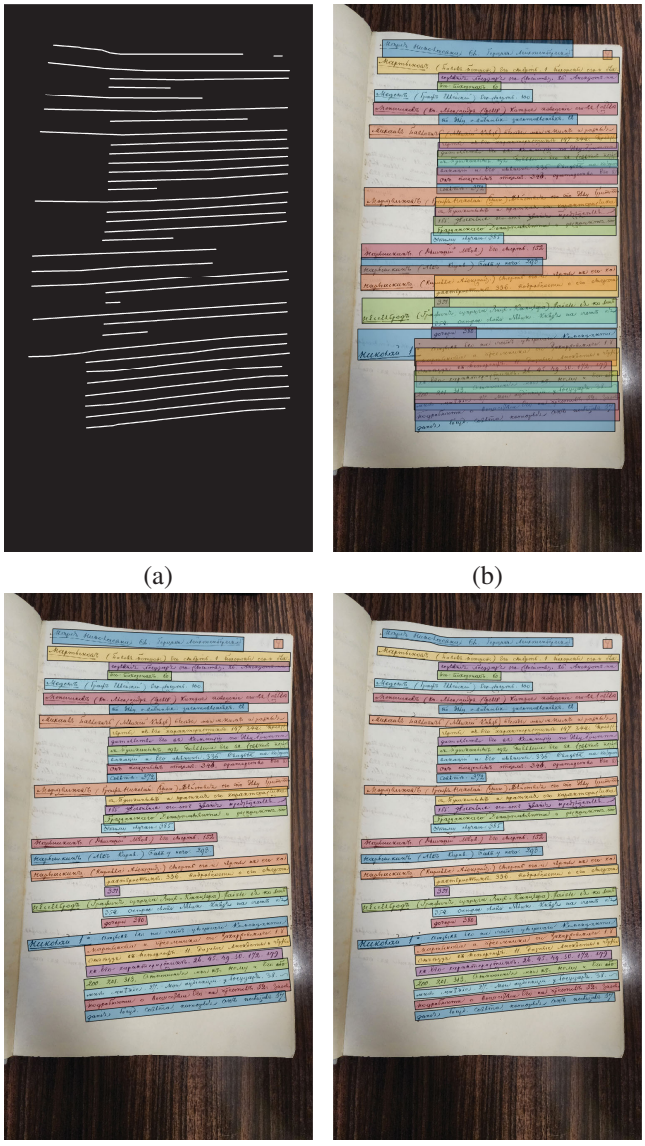
where α_i are calculated as bisectors of two links of the polyline adjacent to the point (x_i, y_i) or as perpendiculars of the extreme links for the ends of the polyline. The offsets d_t and d_b are equal to $\frac{9}{1000}$ and $\frac{7}{1000}$ from the image height, respectively. A mask of arbitrary shape is defined by a polygon obtained by going over the points $(x_1^{(t)}, y_1^{(t)}), \ldots, (x_n^{(t)}, y_n^{(t)}), (x_n^{(b)}, y_n^{(b)}), \ldots, (x_1^{(b)}, y_1^{(b)})$. The rotated box is constructed as a minimal rectangle covering all these points, and the straight box is constructed by the minimal and maximal x and y. All types of defining the shape of lines are demonstrated in Figure 7.

B. Training process

The need to develop specialized networks adapted to specific handwriting is caused by the lack of ready-made models that provide sufficient recognition quality. In particular, the Russian generic handwriting 2 model[2], the most relevant for recognizing Russian text among Transkribus models, showed results of 31.10% CER and 70.53% WER, on our test sample. The TrOCR-ru transformer model[3] produces more than 50% errors in characters while operating extremely slowly. Large language models like ChatGPT 5.0 and Gemini 2.5 Pro require at least thorough prompt engineering combined with few-shot learning, otherwise, with a basic prompt, they produce only partially relevant, albeit readable, text.

To reduce the time costs and demonstrate the possibilities of transfer learning, the YOLO-HTR model with straight boxes, previously trained for recognizing the handwriting of Admiral F.P. Litke [31], was considered as a starting point. In places where the model architecture changed, new layers were left initialized randomly. In the last layer of the model, which calculates the probabilities of symbols, the weights of the symbols that Litke had were copied from the original model, the weights of the symbols found only in Korf were left random, as were the weights responsible for the style—recognition of the style in Litke was not considered. The values of the weights of the loss function termns were set as $L_{box} = 7.5$, $L_{cls} = 0.5$, $L_{dfl} = 1.5$, $L_{ctc} = 0.1$, $L_{seg} = 7.5$. Due to the large size of the images, equal to 2048, they were presented individually during training, without combining them into batches. For images without ground truth text, the CTC loss was not calculated. All other parameters were taken from the YOLOv8 implementation by Ultralytics [35], including the Adam optimizer with a learning rate of 0.002 and a momentum of 0.9.

The training process, visualized in Figure 8, shows the ability to successfully search for lines in any format on an



(a) (b) (c) (d)

Fig. 7. Ground truth line segmentation by baselines (a) for (b) detection, (c) OBB detection, (d) segmentation problems

[2]https://app.transkribus.org/models/public/text/russian-generic-handwriting-2

[3]https://huggingface.co/kazars24/trocr-base-handwritten-ru

image. At the same time, searching for rotated boxes turns out to be slightly more difficult than searching for straight ones, which is evident from the most complex metric mAP50-95 (mean average precision, averaged over IoU in the range from 0.5 to 0.95)—apparently due to sensitivity to errors in restoring the rotation angle.

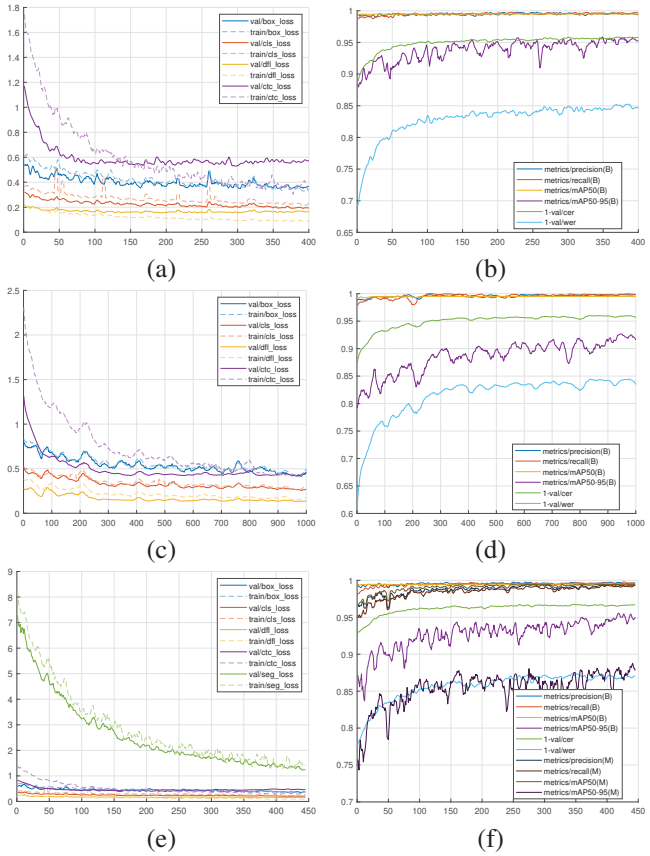


Fig. 8. Values of the loss function (a,c,e) and quality metrics (b,d,f) for (a-b) Detect, (c-d) OBB, (e-f) Segment models. Metrics related to boxes are marked (B), and to masks (M). Only Segment model has metrics related to masks.

It is noteworthy that since the task of predicting the mask is not posed at all in the original model, training actually begins from scratch and the loss function starts with a high value, as Figure 8e shows.

The performance of the models was measured for two configurations, with the results presented in Table II:

- Server: Intel Xeon Platinum 8255C / 196GB RAM / NVIDIA Tesla A100 PCIe 40GB
- Laptop: Intel Core i7-9750H / 16GB RAM / NVidia GeForce GTX 1660 Ti 6GB

The laptop does not have enough video memory for training. Server-based training allows for 600–900 training iterations per day, depending on the model type.

C. Effectiveness of Architectural Solutions

We will separately study the impact of box and mask prediction accuracy on the quality of text recognition. To do this, we will compare the character and word error rates (CER and WER) for ideal boxes/masks and for actually predicted ones. In Table III, the indicators for ideal and actual boxes, when only lines with IoU > 0.5 with the ground truth data are taken into account, do not differ significantly. The slight advantage of actual boxes over ground truth ones can be explained by the fact that when calculating CER and WER for the former, missed difficult boxes are simply ignored, and they, as a rule, contain more challenging text, for example, interline insertions.

TABLE II. TIME CONSUMPTION IN SECONDS PER IMAGE.

Stage + Hardware	Detect	OBB	Segment
Training (Server)	0.887	0.918	1.326
Inference (Server)	0.292	0.304	0.340
Inference (Laptop)	1.262	1.296	1.443

Also, the original line-level VAN model, which is essentially YOLO-HTR+ without a box decoder, was trained on a sample of lines obtained by straightening their polygons relative to the baseline using piecewise perspective transformation (Fig. 9). The VAN was unable to outperform the segmentation model in quality, which speaks, on the one hand, to the importance of straightening lines (YOLO-HTR+Segment does this in the feature space), and on the other hand, to the benefit of taking into account the context that is lost when switching to line-level sampling. Also note that when considering VAN, the line detection problem itself is ignored, and its solution is considered ideal.

TABLE III. THE IMPACT OF LINE DETECTION ACCURACY ON THE QUALITY OF TEXT RECOGNITION. THE INDICATORS, AS BELOW, ARE GIVEN IN %.

Model	CER		WER	
	predicted	ground truth	predicted	ground truth
Detect	3.846	4.060	14.17	14.69
OBB	3.533	3.713	13.45	13.93
Segment	3.227	3.281	12.64	12.66
Line-level (VAN)	–	3.440	–	13.25

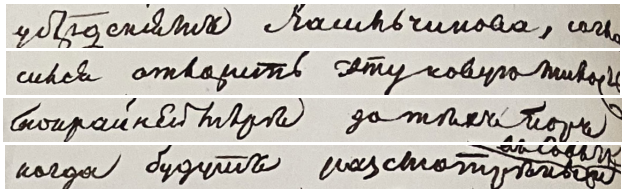


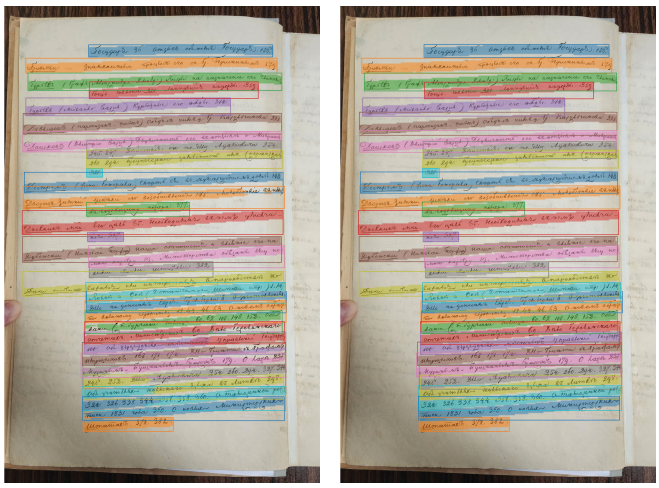
Fig. 9. Samples from the line-level dataset

We will also evaluate the usefulness of the move with reducing the height of the lines to 1 pixel, comparing the results with and without this normalization. The results in Table IV indicate the effectiveness of this technique, especially for straight boxes, which is explained by the strong overlap of the regions of interest without normalization. For rotated lines, the effect is least significant, presumably because the shape of the object itself changes the least with normalization.

TABLE IV
THE EFFECT OF BOX THINNING ON TEXT RECOGNITION QUALITY

Model	CER		WER	
	norm.	non-norm.	norm.	non-norm.
Detect	3.846	10.46	14.17	32.83
OBB	3.533	3.848	13.45	14.92
Segment	3.227	3.921	12.64	14.87

The justification for developing a new mechanism for instance segmentation is demonstrated by Figure 10 with an example of processing an image with strongly inclined lines. As can be seen from the sequence of degraded, broken masks at the bottom of the left image, the original YOLOv8 mechanism insufficiently distributes pixels along lines with intersecting frames. This is confirmed by the numerical estimates from Table V, where a deviation from the near-perfect segmentation of our method results in an increase in CER of nearly 2.5 percentage points.



(a) (b)

Fig. 10. The result of the instance segmentation model: (a) Initial, based on the key-query mechanism, (b) proposed, based on the prediction of the box center.

TABLE V
COMPARISON OF INSTANCE SEGMENTATION METHODS. SEGMENTATION QUALITY METRICS ARE GIVEN FOR MASKS.

Model	CER	WER	Precision	Recall	mAP50	mAP50-95
Original	5.680	17.27	97.54	97.46	97.23	73.13
Proposed	3.227	12.64	99.38	99.15	99.42	88.19

Text formatting in Korf's diaries includes four options: regular text, strikethrough, underline, and superscript. Since ignoring formatting simplifies the task, the percentage of text recognition errors decreases, as can be seen from Table VI. Note that since the share of non-trivially formatted text is 2.22%, and the difference in CER is 1.0-1.3%, we can estimate the share of formatting omissions at 8-10%.

Finally, we check which images show the difference in text recognition quality by analyzing the data from Table VII (only 14 out of 20 images in the test sample have reference text).

TABLE VI
TEXT RECOGNITION QUALITY WITH AND WITHOUT FORMATTING

Model	CER		WER	
	char-style	char only	char-style	char only
Detect	3.846	3.717	14.17	13.90
OBB	3.533	3.394	13.45	13.21
Segment	3.227	3.125	12.64	12.50

TABLE VII
RECOGNITION QUALITY FOR INDIVIDUAL IMAGES (CER IN %)

#	Image	Detect	OBB	Segment
1	3_12rev.	3.127	2.971	2.971
2	3_16	3.465	3.612	3.371
3	3_19rev.	7.316	5.918	5.270
4	3_61	3.918	3.437	3.184
5	3_30	2.763	2.016	2.390
6	3_105rev.	3.156	3.024	3.156
7	3_26rev.	5.651	4.751	4.083
8	3_68	3.152	2.189	1.751
9	3_71rev.	4.134	4.690	4.372
10	3_64rev.	4.009	3.480	3.933
11	3_75	3.655	3.421	3.188
12	3_33rev.	2.524	2.004	1.856
13	3_23	2.986	2.403	2.403
14	3_37	3.987	3.255	3.255

It turns out that the greatest increase in recognition quality when moving to a more complex model is achieved with a fairly complex format of the original lines, namely:

- the presence of curvature, and not just a slope;
- the presence of abundant interline insertions;
- the presence of fragments with a tight arrangement of lines.

An example of such a page is 3_19rev., shown in Figure 11. It can be noted that the segmentation model recognizes the formatting more successfully.

IX. CONCLUSION

The paper explored the principles of constructing neural network architectures capable of solving the problems of handwritten text line search and recognition within a common model. The YOLOv8 was chosen as a model, whose backbone was replaced with layers better suited for extracting handwritten character features. Since these layers have a fairly narrow receptive field, the basic architecture was modified so that the cells responsible for line detection could see the lines in full width with a small number of additional parameters.

The architectures for three different problems—object detection, oriented bounding box detection, and instance segmentation—were modified to match three line shape options of increasing complexity—straight, rotated, and curved lines. At the same time, theoretical problems associated with the ambiguity of coding rotated rectangles were discovered and solved.

Based on the features of the detected objects—an elongated shape and visual similarity—a new method of instance segmentation was proposed, based on a purely geometric principle. The performance of the designed models was analyzed

(a) (b) (c)

Fig. 11. Test image recognition for different network variants: (a) Detect, (b) OBB, (c) Segment

on a collection of historical documents of significant research interest—the diaries of Modest Andreevich Korf, totaling about 9,000 pages. When trained on less than 100 pages with expert decoding, all three models made only 3-4% of errors in characters—such a result ensures good readability of the text both by humans and modern large language models. Note that not only the text but also its formatting (e.g., strikethrough) was recognized, enabling deeper text analysis and contextual search.

ACKNOWLEDGMENT

The research was funded by the Russian Science Foundation (project No. 22-68-00066).

REFERENCES

[1] B. Kiessling, “Kraken OCR system,” https://kraken.re/, 2024, [Online; accessed 03-September-2025].

[2] P. Kahle, S. Colutto, G. Hackl, and G. Mühlberger, “Transkribus - A Service Platform for Transcription, Recognition and Retrieval of Historical Documents,” in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 04, 2017, pp. 19–24.

[3] C. Reul, D. Christ, A. Hartelt, N. Balbach, M. Wehner, U. Springmann, C. Wick, C. Grundig, A. Büttner, and F. Puppe, “OCR4all—An Open-Source Tool Providing a (Semi-)Automatic OCR Workflow for Historical Printings,” *Applied Sciences*, vol. 9, no. 22, 2019.

[4] “Text Recognition & OCR Service — Yandex Vision — yandex.cloud,” https://yandex.cloud/en/services/vision, [Accessed 07-09-2025].

[5] L. Li, “Handwriting Recognition in Historical Documents with Multimodal LLM,” 2024. [Online]. Available: https://arxiv.org/abs/2410.24034

[6] V. Pippi, S. Cascianelli, C. Kermorvant, and R. Cucchiara, “How to Choose Pretrained Handwriting Recognition Models for Single Writer Fine-Tuning,” in *Document Analysis and Recognition - ICDAR 2023*, G. A. Fink, R. Jain, K. Kise, and R. Zanibbi, Eds. Cham: Springer Nature Switzerland, 2023, pp. 330–347.

[7] C. Garrido-Munoz, A. Rios-Vila, and J. Calvo-Zaragoza, “Handwritten Text Recognition: A Survey,” 2025. [Online]. Available: https://arxiv.org/abs/2502.08417

[8] S. Boadas Cabarrocas, “Techniques and instruments for Studying the Autograph Manuscripts of Lope de Vega,” *Hipogrifo*, vol. VIII, no. 2, pp. 509–531, dec 2020, funding information: This work has been carried out thanks to TheaTheor project (n. 794064) funded by Marie Skłodowska-Curie Actions of Horizon 2020 programme (MSCA-IF), and is part of PGC2018-094395-B-I00 Research Project, “Edición y estudio de 36 comedias de Lope de Vega”, funded by the Spanish Ministry of Economy and Competitiveness.

[9] “Deciphering Dickens · V&A — vam.ac.uk,” https://www.vam.ac.uk/research/projects/deciphering-dickens, [Accessed 07-09-2025].

[10] D. Coquenet, C. Chatelain, and T. Paquet, “End-to-End Handwritten Paragraph Text Recognition sing a Vertical Attention Network,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 45, no. 01, pp. 508–524, jan 2023.

[11] P. Voigtlaender, P. Doetsch, and H. Ney, “Handwriting Recognition with Large Multidimensional Long Short-Term Memory Recurrent Neural Networks,” in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 228–233.

[12] Y. Li, D. Chen, T. Tang, and X. Shen, “HTR-VT: Handwritten text recognition with vision transformer,” *Pattern Recognition*, vol. 158, p. 110967, 2025.

[13] L. Cheng, J. Frankemölle, A. Axelsson, and E. Vats, “Uncovering the Handwritten Text in the Nargins: End-to-end Handwritten Text Detection and Recognition,” in *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, Y. Bizzoni, S. Degaetano-Ortlieb, A. Kazantseva, and S. Szpakowicz, Eds. St. Julians, Malta: Association for Computational Linguistics, mar 2024, pp. 111–120.

[14] B. V. Kasuba, D. Kudale, V. Subramanian, P. Chaudhuri, and G. Ramakrishnan, “PLATTER: A Page-Level Handwritten Text Recognition System for Indic Scripts,” 2025. [Online]. Available: https://arxiv.org/abs/2502.06172

[15] N. Garg, N. Sharma, G. Jain, V. Jain, and V. Upreti, “ Handwriting Recognition System Using YOLO and CTC ,” in *2023 International Conference on Advanced Computing & Communication Technologies (ICACCTech)*. Los Alamitos, CA, USA: IEEE Computer Society, Dec

2023, pp. 496–502.
[16] M. Yousef and T. E. Bishop, "OrigamiNet: Weakly-Supervised, Segmentation-Free, One-Step, Full Page Text Recognition by learning to unfold," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun 2020, pp. 14 698–14 707.
[17] D. Coquenet, C. Chatelain, and T. Paquet, "DAN: a Segmentation-free Document Attention Network for Handwritten Document Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, pp. 8227–8243, 2023.
[18] D. Castro, B. L. D. Bezerra, and C. Zanchettin, "An End-to-End Approach for Handwriting Recognition: From Handwritten Text Lines to Complete Pages," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024, pp. 264–273.
[19] S. S. Singh and S. Karayev, "Full Page Handwriting Recognition via Image to Sequence Extraction," in *Document Analysis and Recognition – ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part III*. Berlin, Heidelberg: Springer-Verlag, 2021, p. 55–69.
[20] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 838–846.
[21] C. Wigington, C. Tensmeyer, B. Davis, W. Barrett, B. Price, and S. Cohen, "Start, Follow, Read: End-to-End Full-Page Handwriting Recognition," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 372–388.
[22] M. Carbonell, J. Mas, M. Villegas, A. Fornés, and J. Lladós, "End-to-End Handwritten Text Detection and Transcription in Full Pages," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 5, 2019, pp. 29–34.
[23] W. Sui, Q. Zhang, J. Yang, and W. Chu, "A Novel Integrated Framework for Learning both Text Detection and Recognition," *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 2233–2238, 2018.
[24] Y. Huang, Z. Xie, L. Jin, Y. Zhu, and S. Zhang, "Adversarial Feature Enhancing Network for End-to-End Handwritten Paragraph Recognition," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 413–419.
[25] C. Bartz, H. Yang, and C. Meinel, "STN-OCR: A single Neural Network for Text Detection and Text Recognition," *CoRR*, vol. abs/1707.08831, 2017. [Online]. Available: http://arxiv.org/abs/1707.08831
[26] C. Wigington, "Multi-Task CTC for Joint Handwriting Recognition and Character Bounding Box Prediction," in *Proceedings of the ACM Symposium on Document Engineering 2023*, ser. DocEng '23. New York, NY, USA: Association for Computing Machinery, 2023.
[27] M. Carbonell, A. Fornés, M. Villegas, and J. Lladós, "A neural model for text localization, transcription and named entity recognition in full pages," *Pattern Recognition Letters*, vol. 136, pp. 219–227, 2020.
[28] D. Zhong, S. Palaiahnakote, U. Pal, and Y. Lu, "Struck-out handwritten word detection and restoration for automatic descriptive answer evaluation," *Signal Processing: Image Communication*, vol. 130, p. 117214, 2025.
[29] A. Poddar, A. Chakraborty, J. Mukhopadhyay, and P. K. Biswas, "Detection and Localisation of Struck-Out-Strokes in Handwritten Manuscripts," in *Document Analysis and Recognition – ICDAR 2021 Workshops*, E. H. Barney Smith and U. Pal, Eds. Cham: Springer International Publishing, 2021, pp. 98–112.
[30] S. Pratihar, P. Bhowmick, S. Sural, and J. Mukhopadhyay, "Removal of hand-drawn annotation lines from document images by digital-geometric analysis and inpainting," in *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, 2013, pp. 1–4.
[31] N. Lomov, D. Stepochkin, and D. Kropotov, "YOLO-HTR: Page-Level Recognition of Historical Handwritten Document Collections," in *Analysis of Images, Social Networks and Texts*. Cham: Springer Nature Switzerland, 2025, pp. 208–220.
[32] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," 2020. [Online]. Available: https://arxiv.org/abs/2006.04388
[33] Y. Li, X. Bian, M.-C. Chang, L. Wen, and S. Lyu, "Pixel Offset Regression (POR) for Single-shot Instance Segmentation," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 11 2018, pp. 1–6.
[34] M. A. Korf, *Diary for 1840 [in Russian]*, ser. Historical monument. Moscow: Quadriga, 2007.
[35] "GitHub - ultralytics/ultralytics: Ultralytics YOLO — github.com," https://github.com/ultralytics/ultralytics, [Accessed 08-09-2025].