Document image analysis and recognition: a survey

V.V. Arlazarov [1,2], E. I. Andreeva [2], K.B. Bulatov [1,2], D.P. Nikolaev [3], O.O. Petrova [2], B. I. Savelev [2], O.A. Slavin [1]
[1] Federal Research Center "Computer Sciences and Control" Russian Academy of Sciences, 117312, Moscow, Russia, prosp. 60-letiya Oktyabrya, 9;
[2] LLC "Smart Engines Service", 117312, Moscow, Russia, prosp. 60-letiya Oktyabrya, 9;
[3] Federal Publicly Funded Institution of Science, Institute for Information Transmission Problems n.a. A.A. Kharkevich of Russian Academy of Science, 127051, Moscow, Russia Bolshoy Karetny per. 19

Abstract

This paper analyzes the problems of document image recognition and the existing solutions. Document recognition algorithms have been studied for quite a long time, but despite this, currently, the topic is relevant and research continues, as evidenced by a large number of associated publications and reviews. However, most of these works and reviews are devoted to individual recognition tasks. In this review, the entire set of methods, approaches, and algorithms necessary for document recognition is considered. A preliminary systematization allowed us to distinguish groups of methods for extracting information from documents of different types: single-page and multi-page, with text and handwritten contents, with a fixed template and flexible structure, and digitalized via different ways: scanning, photographing, video recording. Here, we consider methods of document recognition and analysis applied to a wide range of tasks: identification and verification of identity, due diligence, machine learning algorithms, questionnaires, and audits. The groups of methods necessary for the recognition of a single page image are examined: the classical computer vision algorithms, i.e., keypoints, local feature descriptors, Fast Hough Transforms, image binarization, and modern neural network models for document boundary detection, document classification, document structure analysis, i.e., text blocks and tables localization, extraction and recognition of the details, post-processing of recognition results. The review provides a description of publicly available experimental data packages for training and testing recognition algorithms. Methods for optimizing the performance of document image analysis and recognition methods are described.

Keywords: document recognition, image normalization, binarization, local features, segmentation, document boundary detection, artificial neural network, information extraction, document sorting, document comparison, video sequence recognition.

Citation: Arlazarov VV, Andreeva EI, Bulatov KB, Nikolaev DP, Petrova OO, Savelev BI, Slavin OA. Document image analysis and recognition: a survey. Computer Optics 2022; 46(4): 567-589. DOI: 10.18287/2412-6179-CO-1020.

Introduction

Paperwork and subsequent document flow have accompanied the organized activity of mankind since ancient recorded times. In the 1980s, with the development of electronic computing devices, the migration of paper document management to electronic form began. Despite consistent predictions that electronic document flow will soon replace paper one, after 40 years people around the world still use the mixed electronic and paper document flow. In such a situation it is important to have technologies that help easily and accurately translate a document from one form to another. The methods of preparing electronic documents for printing can be considered from the scientific point of view, taking into account the need for automatic layout, but they are not considered in this review. Here we concentrate on relevant methods of automatic analysis and recognition of document images (document image analysis, DIA) that are necessary for the electronic processing of paper documents.

For further discussion, we need to introduce several basic notions. We define a document as a set of unchangeable (for a fixed class of documents) elements and information attributes. The attribute values are interpreted by an information system to perform operations on a document. Examples of such operations are registration, control, and cancellation of a document, synchronization of document attributes and data in the electronic archive. Documents that are part of the document flow can be called business documents. Attributes of business documents, unlike other documents, are parameters of document flow processes. Examples of documents of various types are shown in fig. 1.

For further discussion, we need to introduce several basic notions. The graphical representation of any document contains static graphic elements and content elements: information fields and confirmation elements. A set of static elements is referred to as the document template. Static graphic elements are usually represented by text fragments within the document: headings and field names ("name", "address", etc.). Other types of static elements are boundary lines and checkboxes. Information fields usually contain text, although there are also bar-

code fields. Interestingly, in some cases, a barcode can be a static element, which facilitates document classification. There are also multi-line text fields, parts of which can be transferred to another page of a multi-page document. Confirmation elements include signatures, seals, and handwritten notes. The documents may include sections that combine logically connected groups of elements and fields. One of the most important elements of certain templates is the complex background. For ID or registration documents, text on stamped paper with a so-called "guilloché" background is typical. Such background is a classical static element. The background of bank plastic cards can be a random image that changes from series to series. It is not a static element in the strict sense, but as a rule, it is not an information attribute either.

Fig. 1. Samples of ID forms (on top, from set of data MIDV-500 [1]) and flexible forms (on bottom, from set of data Funsd [2])

For further discussion, we need to introduce several basic notions. The graphical representation of any document contains static graphic elements and content elements: information fields and confirmation elements. A set of static elements is referred to as the document template. Static graphic elements are usually represented by text fragments within the document: headings and field names ("name", "address", etc.). Other types of static elements are boundary lines and checkboxes. Information fields usually contain text, although there are also barcode fields. Interestingly, in some cases, a barcode can be a static element, which facilitates document classification. There are also multi-line text fields, parts of which can be transferred to another page of a multi-page document. Confirmation elements include signatures, seals, and handwritten notes. The documents may include sections that combine logically connected groups of elements and fields. One of the most important elements of certain templates is the complex background. For ID or registration documents, text on stamped paper with a so-called "guilloché" background is typical. Such background is a classical static element. The background of bank plastic cards can be a random image that changes from series to series. It is not a static element in the strict sense, but as a rule, it is not an information attribute either.

By document recognition, we denote the extraction of attribute values from the document image. Usually, prior to recognition, some information about document structure and characteristics of its attributes is known. Among the variety of documents, the following classes can be considered based on the document structure:

- rigid forms created in a uniform polygraphic way, such as ID documents (passport, ID card), driver's license;
- flexible forms created based on the known templates, e.g. standard questionnaires, notifications, declarations, bank plastic cards;
- documents without a strict template, e.g. contracts, letters of attorney;
- documents that do not have a common template, such as business letters.

There are single-page and multi-page documents. Each page of a multi-page rigid form is often treated as a single-page document. In other document classes, static elements and content text can be transferred from one page to another, hence separation into single pages is not constructive in this case.

It is also possible to classify document types based on their application and attribute characteristics. For example, IDs, registration documents, financial and credit documents, contract documents are considered.

As we will show below, there are still tasks within this field that have not been solved sufficiently enough for a fully automated process of document transformation. In particular, it is related to the problems of classification and semantic analysis of poorly structured documents. Besides, technological progress has changed the emphasis within the paperwork, though in a different way than it was predicted before. As telephones have been replaced by internet-enabled smartphones with cameras and powerful processors, the demand for document processing via such devices has increased. While paper documents are still ubiquitous, flatbed scanners have become far less common and are no longer perceived as an essential attribute of document digitization. The transition from the analysis of scanned images to the analysis of document photographs required the development of new approaches, resistant at least to projective distortions and uneven illumination. This brought the field of DIA closer to computer vision.

The main ways of document image capturing are scanning and photographing, including capturing via mobile devices. Very noisy photocopies of pages can be still found nowadays. Any digitization method can potentially cause a defect of image loss around the frame border. The above-mentioned variability of document content and peculiarities of modern digitization methods result in a great variety of document images, which complicates DIA.

A large number of works published, including recently confirms the relevance of the research in field of DIA. The following specialized conferences are fully or in separate sections dedicated to DIA problems, models, and algorithms:

- ICDAR – International Conference on Document Analysis and Recognition (https://icdar2021.org, https://icdar2020.org, …);
- ICPR – International Conference of Pattern Recognition (https://icpr2020.net);
- ICMV – International Conference on Machine Vision (http://icmv.org);
- ASPDAC – Asia and South Pacific Design Automation Conference (https://aspdac2022.github.io/index.html);
- ICIP – International Conference on Image Processing (2020.ieeeicip.org, 2019.ieeeicip.org, …);
- CCVPR – Conference on Computer Vision and Pattern Recognition (http://www.ccvpr.org).

High interest in DIA is also shown by an increase in the number of citations for papers (for example see fig. 2).

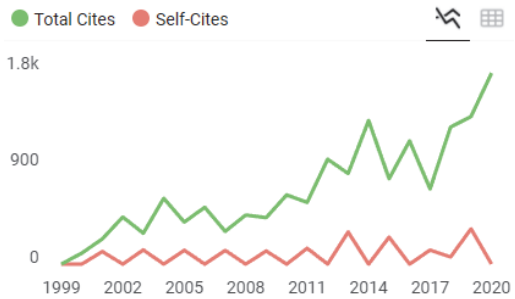


Fig. 2. Changes in the number of citations of ICDAR publications [25]

The long history of research in the field of DIA is reflected in numerous conference proceedings, reviews [3,4,6,7 – 21], and books [22 – 24]. But older publications do not account for the technological shift of recent 10 years, when the full DIA process from image or video capture to full recognition and results presentation became possible directly on mobile or autonomous devices. At the same time, to the best of our knowledge, publications of recent years considered only separate tasks (such as document image classification [3], extraction of information from poorly structured documents [4], etc.) or advances of particular methods (mostly machine [16] and deep [20, 21] learning).

Thus, in document image analysis, some long-standing issues remain unresolved while new problems are emerging. On the other hand, the methodology of data analysis and recognition has undergone significant changes over the past 40 years, and machine learning methods, primarily the neural network approach, are steadily replacing the classical algorithms. In addition, a common practice nowadays is to experimentally compare various methods on publicly available massive datasets. Hence, it is important to review the main relevant tasks of DIA, the corresponding datasets, and the most promising methods in a single review.

1. A review of document image analysis methods

1.1. Document image normalization

Due to human and technical factors, digital images may contain distortions. Elimination of distortions, or normalization, is a classical task of digital document imaging. A particular case of such distortions is the skew (rotation) of the acquired document image. Correction of such skew is considered to be one of the main stages of document image preparation for further analysis [26].

To this moment, many approaches have been developed to solve the problem of document skew estimation and correction. In the fundamental work of Hull published in 1996 [5] several dozens of different algorithms were considered. A more modern review of current methods can be found, for example, in [6].

To get an idea of document skew detection algorithms let consider one of the simplest and most popular methods, which is based on Principal Component Analysis, or PCA. Given a set of points, the first principal component corresponds to a line that minimizes the sum of squares of the distances of the points from this line. By method`s assumption the input document image contains some linear structures which in general share the same direction which corresponds to this best-fitting line, and consequently matches the skew angle.

To find principal components of the input binary image the set of its black pixels is analyzed. Firstly, the covariance matrix C is calculated for these pixels coordinates. The eigen vectors of C are corresponding directly to the principal components and the largest one thus is corresponding to the skew angle required to find. The method is very computationally efficient, but requires preliminary binarization of the input image. In depth it is described in work [27]. Other methods can be roughly divided into five main groups.

The first group of methods is based on the analysis of horizontal parallel projections of the document image. For an image I of size $M \times N$ the parallel horizontal projection is defined as follows:

$$\pi_I(j) = \sum_{i=0}^{M} I(i,j). \quad (1)$$

Skew detection in this case is reduced to the successive rotation of the original image by a predetermined set of angles, and the best projection is chosen. Obviously, to rank the obtained projections, some criterion $f(\pi)$ should be provided. The angle ϕ at which the global extremum of $f(\pi)$ is reached is selected as a result. The sum of squared projection element values is often used as a criterion, i.e.,

$$f(\pi) = \sum_{j=0}^{N} \pi^2(j). \quad (2)$$

The second group of methods relies on the Hough transform applied to the document image. The concept of this transformation is to count the number of pixels along

all possible straight lines in the original image. The normal (from "normal" in geometry) parameterization (ρ, ϕ) is usually employed to define the lines. With this choice, the problem is reduced to the search for such lines of direction ϕ, where the given criterion reaches maximum. The criterion is usually introduced as the sum of Hough image derivative squared values [26]. The main problem of this group of methods is high computational complexity. To overcome this drawback, it was proposed [28] to use a fast discrete Radon transform algorithm to calculate the Hough image, which reduced the computational complexity from $\Theta(n^3)$ to $\Theta(n^2 \log n)$.

The third group of methods is based on the analysis of connectivity components. Their idea is to cluster the extracted components and identify linear clusters corresponding to the lines of the document. After determining the skew in each of the found clusters, a single "average" skew is selected.

The fourth group of methods is based on document image segmentation algorithms. First, various graphic primitives, such as text lines, elements of the document structure, individual characters are detected. After that, the estimation of the skew for each of the received components is performed, and the document skew is calculated.

The fifth group includes methods based on the use of artificial neural networks. A noteworthy approach was proposed in [29]. Here, the problem document skew estimation is treated as a classification problem with 360 classes, each corresponding to a certain angle.

With such a large variety of methods, the question is which one to choose. For this, in 2013, a special competition was organized. It is devoted solely to the problem of document skew estimation [30]. For this competition, a data set modeling the main problems encountered in document skew detection was created. For all images within this dataset, experts manually specified the expected skew detector response. Method performance evaluation criteria were introduced. According to the results of the competition, the method based on the Fourier transform with special image preprocessing [31] proved superior.

Document skew detection for subsequent normalization is required not only for correcting the entire document. The processing of document images requires the recognition of the text images by optical character recognition (OCR) modules. Text images normalization improves the accuracy of OCR.

It should be noted that the normalization tasks for printed and handwritten attributes are significantly different. In the first case, it is a matter of finding a single dominant direction on the entire fragment. The handwritten text images skew can vary within a word fragment, which leads to much more complex schemes of fragment processing. Examples of such schemes based on the principles of dynamic programming can be found in [32].

1.2. Document image binarization

Document image binarization separates pixels of document contents from pixels of background. This classification of pixels allows for a significant reduction in the memory required to store the image, and it simplifies all the subsequent OCR steps.

Binarization methods can be grouped according to the threshold surface introduction. Threshold surface $T(x,y)$ is a two-dimensional function defined on the domain of the original image I, which specifies the binarization threshold value for the corresponding pixel of the original image. Hence, bi-tonal image B is constructed as follows: $B(x,y) = [I(x,y) > T(x,y)]$.

The simplest group of methods are the so-called global binarization methods, in which the threshold surface T is the same at each point. Such algorithms are characterized by very high performance, but are extremely sensitive to the image content. The most widespread of such methods is the Otsu method, proposed back in 1979 [33]. The Otsu method determines the threshold value based on the image brightness histogram, minimizing the weighted intraclass variance. Global methods yield poor results in the cases of unevenly illuminated documents, outlier noise, and other distortions. Hence, various improvements have been proposed. Special image processing applied prior to global binarization can significantly improve the results. One popular technique for such preprocessing is image background estimation and normalization [34].

In methods within the group of local (or adaptive) binarization algorithms [35], the threshold surface values depend not only on the brightness value of the pixel itself but also on the brightness of its neighbors in some given neighborhood. Such algorithms are much more robust to both document imaging conditions and to the presence of noise in the resulting image. On the other hand, local methods require tuning parameters, and the selection of these parameter values is a nontrivial task. Therefore, local methods with automatic determination of parameter values, including the use of the Otsu criterion generalization, were proposed [36].

Classical binarization methods are characterized by high performance but do not allow for consistently high accuracy in all applications. To increase the accuracy, some works suggest using several different binarization results and make a final classification decision for each pixel based on these results, but this is clearly a mere palliative. The most popular group of methods at the moment is based on pixel-by-pixel classification using machine learning, mainly neural networks of different architecture [37, 38]. Such algorithms provide high binarization accuracy in a number of cases, they usually do not require any preprocessing of the document image but are computationally complex.

Currently, choosing a suitable binarization method is a non-trivial task. The document image binarization

(DIB) platform [39] has been launched to track the progress of document binarization. It consolidates all available knowledge and artifacts related to document binarization. The platform includes datasets such as Document Image Binarization Competition (DIBCO) datasets (all years), Nabuko, LiveMemory, synthetic data, and many others. The platform has also begun to host relevant contests, the distinguishing feature of which is the consideration of the running time of the proposed solutions.

1.3. Document classification and localization based on keypoints

In image analysis, keypoints are image points, local neighborhoods of which include distinctive features compared to other neighborhoods. In the last few years, methods based on detection, analysis, and comparison of keypoints have shown good results for object detection, image classification, panoramic image stitching, facial recognition and etc.

Keypoints detectors are methods for search of Region of Interest (ROI) considered as reference points/areas for local descriptors that describe the keypoint and the features within its vicinity. Hence, a set of descriptors allows characterizing the local areas of the image. There are various methods for descriptors detection and calculation. Usually, these methods are proposed in pairs and have the same name, but this is not necessarily the case.

In document image recognition, the keypoints mechanism is used to classify and localize a document or its part by comparison with a reference. The localization of documents with a template of fixed geometry can be carried out by this method while the document is being captured by a camera, in the presence of projective distortions. In this case, the algorithms of the random sample consensus (RANSAC) [40] family are usually employed for comparing the constellations of keypoints. This approach fully determines the internal coordinate system of the document within the image. In this case, an important property of keypoints is affine invariance (the projective transformation is locally affine). When the number of keypoints is large, classification is performed via the Bag-Of-Features model using fast search trees [41].

The simplest examples of keypoints are corners, ends of segments, and other topological features of the morphological skeleton of images. Keypoints have an advantage over other image features, such as edges and regions, because they are better localized, yet have highly informative content [42]. Keypoints tend to be stable in terms of image transformations and transformations that change the viewpoint. The disadvantages of using keypoints include a decrease in the probability of correct classification when the volume of possible classes increases, and instability to strong defocusing. Among the fast methods of keypoint detection are the Harris corner detector [43], FAST [44], Difference-of-Gaussian (DOG) [45] and YAPE [46].

The Scale Invariant Feature Transform (SIFT) detection algorithm is not sensitive to image scaling and rotation and is partially invariant to illumination and viewpoint changes [45]. The SIFT algorithm is highly distinctive, i.e., with high probability it allows a single feature to be correctly compared against a large feature database, providing the basis for object and scene recognition. The computational cost of keypoints detection in SIFT is optimized using cascade filtering, where more expensive operations are applied only to locations that have passed the initial validation. Despite this, the main drawback of SIFT is its high computational complexity.

The Speeded Up Robust Features (SURF) method, which is based on Gaussian multiscale image analysis, was introduced in [47] in 2008. The SURF detector is based on the Hesse matrix determinant and uses integral images to increase the speed of feature detection. The 64-bit SURF descriptor represents each detected keypoint with a Haar wavelet response distribution within a particular neighborhood. For each of the 44 subregions, each feature vector contains four parts:

$$v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|), \quad (3)$$

where the wavelet responses dx and dy are summed for each subregion, and the absolute values of the responses $|dx|$ and $|dy|$ provide the polarity of the intensity changes. The SURF algorithm is invariant to rotation and scale but has relatively weak affine invariance. However, the descriptor can be extended to 128 bits in order to deal with large changes in the viewpoint. The main advantage of SURF over SIFT is its low computational cost.

The Oriented FAST and Rotated BRIEF (ORB) algorithm was proposed in 2011. ORB is a combination of a modified FAST (Features from Accelerated Segment Test) detector [44] and a directed-normalized BRIEF (Binary Robust Independent Elementary Features) descriptor. FAST features are detected in each layer of the multiscale pyramid, and the quality of the detected points is evaluated using the Harris corner detector. Since the BRIEF method is extremely sensitive to rotations, a modified version of the BRIEF descriptor was used. The orientation of FAST features is estimated by the intensity centroid, the corner's intensity offset from its center. This offset measures the orientation: the vector between the object location and the centroid [48]. In [48], the centroid is defined as:

$$C = (m_{10}/m_{00}, m_{01}/m_{00}), \quad m_{pq} = \sum x^p y^q I(x,y). \quad (4)$$

The ORB method is robust to scaling, rotation, and limited affine distortions.

The Binary Robust Invariant Scalable Keypoints (BRISK) algorithm is described in [49]. It detects corners using the AGAST method [50] and filters them using the Harris corner detector while searching for maxima in a multiscale spatial pyramid. The BRISK descriptor is based on determining each keypoint's characteristic direction, which allows for the invariance to rotation. To compute the orientation of keypoint k, BRISK uses local gradients between sampling-point pairs:

$$g(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{\left| p_j - p_i \right|^2}, \quad (5)$$

where (p_j, p_i) is a sampling-point pair. The smoothed intensity values at these points are $I(p_j, \sigma_j)$ and $I(p_i, \sigma_i)$. To allow for illumination invariance, results of simple brightness tests are concatenated, so the descriptor is constructed as a binary string. The BRISK method is rotation, scale, and limited affine distortions invariant.

The development of new methods for keypoints detection and description continues: along with methods that have already proven themselves useful, many new experimental methods are based on neural networks [51].

In a review of keypoints detection and description methods [52], the experimental results describing the accuracy and processing time of various methods (see tab. 1). The authors generated 1630 videos [52] based on the book from the open WikiBook dataset [53], which contains 700 A4-sized pages of text. The quality was measured based on the accuracy of correctly detected regions of interest within the images. For methods evaluation, other open datasets, such as MIDV-500 [1], MIDV-2019 [54], Tobacco dataset [55], can be employed.

Tab. 1. Detection quality and processing time of keypoints detection methods

Descriptor	Detector	Quality (for different rotation angles) (%)			Average processing time for 15 frames of the video, s
		minimum	maximum	Average	
SIFT	SIFT	88.4	83.9	86.4	1.21
SIFT	SURF	98.6	93.2	96.4	2.76
SIFT	BRISK	98.4	98.7	98.6	1.4
SIFT	ORB	97.1	98.3	97.9	1.5
SIFT	AKAZE	61.6	64.7	63.6	0.7
SURF	SURF	10.0	92.5	59.5	3.06
SURF	SIFT	5.0	55.2	31.0	1.2
SURF	BRISK	69.9	94.0	84.2	1.8
SURF	ORB	80.0	96.9	90.4	1.3
SURF	AKAZE	0.3	1.0	0.7	0.8
ORB	ORB	79.0	81.2	79.9	6.81
AKAZE	AKAZE	68.7	73.1	70.4	9.9
KAZE	KAZE	0.4	0.7	32.3	2.1

1.4. Document boundary detection

Localization via keypoints detection assumes that a document's template has unique features, and they should be known in advance. Often, this is not the case. Office documents (contracts, etc.) may have no template, while bank cards have extremely varied templates and are usually unknown in advance. In this case, at the geometric normalization step, a rectangle is chosen as a model for a document. The model can be augmented with a known aspect ratio of the document. When scanning a document, it is sometimes beneficial to augment the model with absolute dimensions. The image of a document in scanned images is a rectangle that differs from the preimage by shift, rotation, and isotropic compression, and the compression ratio is usually known (as it is set by the scanning resolution). In mobile camera images, the document image is an almost arbitrary convex quadrangle, since the image is subjected to projective distortion.

Works on document boundary detection consider both scanned images and photographs. The second case is more general, because such images may have a non-uniform background, which increases the probability of false positives. In addition, the borders of the document may be partially obscured. We will consider the problem of document detection in camera images. The methods described below can be also applied to scanned images by introducing additional geometric constraints on the resulting quadrangle.

There are three main approaches to document boundary detection: contour analysis, image segmentation, and document corner detection.

The most widespread approach is the analysis of contours in the document image. One of the classic works using this approach is [56], which introduces a method for the detection of a whiteboard in an image. The algorithm proposed by the authors includes the following steps:

- edge detection in an image converted to grayscale;
- straight lines detection on edge map via Hough transform;
- formation of the quadrangle candidates satisfying geometric conditions;
- estimation of each quadrangle B consistency: *consistency (B)* = $\Sigma_{b \in B}$ *consistency(b),* where *b* is a quadrangle side, and *consistency(b)* is calculated as the fraction of the side's pixels which have a match on edges map;
- the best quadrangle selection and its refinement.

This approach is the basis of many algorithms for document boundary detection. The authors of [57], while following the described steps, propose to omit grayscale conversion before border detection due to the possible loss of information. Instead, the image is converted to the CIELAB color space. A separate edge map is generated for each channel, then the results are combined using the OR logical operation.

Some works propose to complement the described approach with high-frequency noise filtering methods, which can reduce the number of false positives of the straight lines detector. The paper [58] describes a text filtering method: on the edge map, the connectivity components are selected, their bounding rectangles are calculated, and each component is estimated by the aspect ratio of the rectangle, its size relative to the region of interest, and the number of pixels.

For quadrangle candidates formation, the classical approach assumes the use of exhaustive search with cutoffs. The same technique is used by most works described here.

Some works introduce new ways of quadrangle consistency estimation. In [59], the aspect ratio of the document P_d is considered known, and the quadrangle evaluation verification takes into account the deviation of the evaluated aspect ratio P_e from the given one. Also, the

weight of the corresponding lines within the Hough space W_l is considered, and for each corner, the penalty P_c is calculated: the sum of pixel intensities on the edge map within the corresponding sides and outside of them. This allows for the filtering out the quadrangles for which one of the sides is formed by an edge that is beyond the quadrangle boundaries:

$$consistency(B) = \left|1 - P_d / P_e\right| \left(\sum_{i=1}^{4} W_{l_i} - \sum_{i=1}^{4} P_{c_i} \right). \quad (6)$$

The paper [60] also considers the issue of reducing the number of false positives when localizing a quadrangle in the image, which is especially important in the case of a complex background, elements of which can obscure the document or form areas similar to a quadrangle. The authors propose to complement the classical approach with the calculation of consistency via the evaluation of the contrast between the inner and outer regions of the quadrangle.

It is worth noting that the basic assumption of the contour analysis approach is that all sides of the document are visible in the image and have a strong contrast to the background. However, when used in real-world document recognition systems, this assumption may not hold: one of the sides of the document may be outside the frame or be obscured, making it nearly impossible to form a quadrangle of 4 sides.

The second approach treats document boundary detection as an image segmentation problem. This approach imposes weaker constraints on the visibility of the document boundaries, but in most cases, the algorithms are more computationally complex. The authors of [61] use the "local-global variational energy" functional based on probabilistic distributions of coordinate vectors and color characteristics of pixels for document regions and background. The segmentation problem is solved by optimizing this functional.

A separate group of works (e.g., [62]) employs a tree of shapes, which is a hierarchical representation of an image based on the inclusion of its level lines to detect objects in the image. It is worth noting that the algorithm described in [62] performed best in the first test of the ICDAR 2015 SmartDoc thematic competition.

In [63], a few background points are selected in the image, areas similar to these points are segmented, and the document quadrangle is found iteratively, minimizing the number of background-like pixels inside the quadrangle, and maximizing their number outside it.

A neural network approach is also used for document segmentation: U-net architecture [64]; OctHU-PageScan - Fully Octave Convolutional Neural Network based on the same architecture; HoughEncoder [65], autoencoder which employs direct and transpose Hough transforms.

The third approach involves the detection of document corners. In [66], a two-stage neural network is used for this purpose. The first neural network roughly estimates the position of document corners in the input image; then each of the corners is iteratively refined by a second neural network, which takes the local neighborhood of each document corner as input. The limitations of this approach are the requirement of visibility and high contrast to the background of all document corners, which is often not possible when capturing an image with a mobile camera.

It should be noted that there are also works that propose to combine the methods of anchoring based on keypoints, and detection of the document borders quadrangle to increase the accuracy [40].

There are several public datasets that are used to measure the quality of document boundary detection: the dataset of the ICDAR 2015 SmartDoc competition [41], the MIDV-500 dataset [1], the MIDV-2019 dataset [54], the SEECS-NUSF dataset [66], the CDPhotoDataset described in [67] which will be available after the ICDAR2021 competition is concluded. The most commonly used metric for evaluating the quality of document boundary detection is the Jaccard index. Particularly, it was used in the ICDAR 2015 SmartDoc competition.

1.5. Visual appearance based document image classification and localization

There is another approach to document localization and classification. It is used when the document template does not have unique and stable local features, but the document can still be easily recognized by its visual appearance. In such cases, some machine learning methods can be applied to classify the document. Classification methods that do not provide simultaneous document localization are applied after the previously discussed algorithms of geometric normalization by localizing boundaries. A rather detailed review of such methods can be found in section 2.2.2 of the paper [3].

Moreover, there is at least one machine learning method that solves both problems. A method of object detection based on fast computation of local contrasts (the so-called Haar features) was proposed by Viola and Jones for facial recognition in photographs [68]. When they used the concept of an integral image, the contrast calculation for an arbitrary rectangular sub-area was performed on average in a few clock cycles of the central processor. Hence, quick application of the detectors based on the calculation of Haar features to each rectangular sub-region of the image taken with all possible scales from a predetermined set, was possible. As a result, the Viola and Jones method allowed for running capabilities of 15 frames per second on a personal computer in 2000 without the use of a graphics processing unit.

Another advantage of the Viola and Jones method as compared to neural network deep learning methods is that it is less demanding in terms of the amount of training data. This is important for the tasks associated with the recognition of identity documents when training data collection is associated with serious legal constraints. In [69], several Viola and Jones classifiers were trained to

localize several types of identity documents in scanned images, with a positive sample for each class consisting of only a few hundred examples. The method is robust to noisy elements such as guilloché patterns, security fibers, handwritten marks, etc., because it employs low-resolution features that form the visual image "as a whole," ignoring small details.

The Viola and Jones method is also applicable for searching and classifying the reference elements of a template or content when the document does not contain a sufficient number of keypoints. Another interesting object is round seals, the detection and removal of which plays a special role in document recognition. First, seal detection is necessary to verify the authenticity of the document. Second, seals contain meaningful information, the recognition and control of which is important [70].

1.6. Document structure analysis algorithms

After preliminary image processing and localization of document boundaries in the image, the most important stage is the analysis of its structure and segmentation of the document into its constituent parts. The tasks included in the document structure analysis block can be divided into the following groups:

- detection and localization of text elements (lines, words) of the document;
- analysis of the text blocks structure and the segmentation of the document into separate text blocks, columns, paragraphs, or text fields; examples of text blocks are shown in fig. 3;
- finding the text reading sequence of text blocks (columns, paragraphs, fields) in a document;
- detection and localization of graphic elements within the document, i.e. shapes, figures, formulas, seals, stamps, etc.

Fig. 3. An example of the structure of text blocks (from [3])

Since the segmentation of a document image into information fragments largely depends on both the structure of the document and the specifics of the particular recognition systems, a wide range of methods and approaches proposed to solve associated problems, grouped in different ways, can be found in the literature.

In terms of document features, document structure analysis methods are divided into rigid templates structure analysis methods (i.e., documents with variable elements located at the same places throughout samples), single-column and multi-column Manhattan structure analysis (i.e. documents that can be decomposed into non-overlapping orthotropic rectangular regions, each containing either a text block or a figure, a table, etc.), and analysis of documents with arbitrary structure (i.e. documents where graphical elements and text blocks can be of arbitrary shape and with arbitrary text direction). Methods of document structure parsing are also divided into methods involving a priori knowledge of the template ("grammar") of the document components structure, and methods applicable to documents with a priori unknown structure.

Algorithms for document structure analysis in terms of approaches can be divided into three groups: bottom-up, top-down, and hybrid approaches.

The bottom-up approaches rely on the preliminary identification and analysis of individual document components, such as pictures, text lines and words, static texts, keypoints, markers, etc. Based on these components, such approaches restore the overall structure. Bottom-up group includes methods based on groups of connected components analysis, line segments identification [71], morphological image analysis for text components search [72], textual key points and relations between them [73], and methods based on Voronoi diagrams and Delaunay triangulation.

The top-down approaches involve the analysis of the document image as a whole and the preliminary segmentation of the document structure into blocks, each is subsequently analyzed independently. Such methods include analysis of image projections on horizontal and vertical axes and gap analysis [74], document decomposition into blocks by optimal superposition of Gaussian kernels [75] or other types of functions.

The third group includes hybrid approaches, the joint application of several techniques, combining top-down and bottom-up approaches with additional modifications, determined by the task or the document specificity. This group also includes methods based on machine learning and the application of deep neural networks, which have been actively developing in recent years. Thus, models based on convolutional and recurrent neural networks, including deep VGG-16 and YOLOv2/v3 architectures are used to detect, search, and classify individual document components such as shapes and figures [76], and formulas, as well as textual elements of documents and the links between them. Deep convolutional models are used to classify blocks of documents [77], full convolutional neural networks, Trainable Multiplication Layers (TML) convolutional neural networks, and Siamese networks are employed to solve the problem of semantic document segmentation [78].

The problem of detection and localization of arbitrary text lines and individual words, which is widely discussed in the literature together with methods of document structure analysis (although, strictly speaking, it is a more general task that also takes place outside the document processing systems), should be pointed out sepa-

rately. The terms "text in the wild", "word spotting", or "text spotting" describe this problem. To solve it, various methods are created. Such methods allow for the selection of sections of the input images containing characters, graphemes, words, and whole text lines under the conditions of pixel (brightness, color) and geometric distortions. Methods include both structural analysis of image sections followed by the combinatorial selection of sections with text features, use of local text feature learning techniques, and global use of deep convolutional neural networks for text pixel extraction in arbitrary images [79].

The broad topic of document structure analysis and such subtopics as document image segmentation and searching for graphical or textual elements kindles the interest of the scientific community in developing new algorithms and methods for related tasks. Scientific contests, such as the ICDAR Recognition of Documents with Complex Layouts and others, are held regularly. To evaluate the newly published algorithms, the traditional open datasets such as PRImA [80] (document structure analysis), COCO-text [81] (text detection and recognition from natural images), and the datasets of the international project for the development of document analysis systems MAURDOR [82] are used. In addition to them, international teams create new datasets reflecting the specifics and characteristics of individual document types, for example, the BID [83] and MIDV-500 [1] datasets for the analysis of identity documents.

1.7. Table recognition methods

Tables are an important element of many types of documents. Despite the long history of research, the problem of their recognition remains relevant, and the bulk of the results are given for closed datasets, which complicates the analysis and verification of the results. Below we will consider the main results of table recognition in printed documents (unconstrained documents), as they are supported by regular competitions [84 – 86] and are applicable to table recognition in documents.

General approaches and methods for table recognition can be found, for example, in the review section of [87]. When processing tables, algorithms can rely on more heuristics, and additionally use such characteristic structural elements as layout lines, corners and junctions, alignments, etc. Nevertheless, in recent years, there has been a shift in researchers' interest from engineering problem-solving methods to learning ones.

In various sources, tables are formalized in different ways [88], and this is the result, apparently, of the fundamental impossibility of formulating a single strict definition of the concept of the table. The main subtasks of optical table recognition can be formulated as follows:

- table detection (localization), i.e. detection of the borders of tabular structured information areas, usually in the form of orthotropic rectangles, in the image;
- table segmentation (structure recognition), i.e. recognition of the table structure as a graph of adjacent cells, specifying the relative position of each table cell;
- table recognition, i.e. recognition of the table structure and the contents of each cell.

For the documents with a template, during the segmentation, the specification of the table type according to the description can be accessed. For the documents without a template, different variants of the table reconstruction tasks may be formulated (table understanding, table reconstruction, information extraction), which serve to restore the original structure of the table information content for interpretation, transformation, etc. In general, the table image may contain zones of different types: header zones, matrix core, identification and information zones. In practice, all types of zones are rarely used simultaneously. The main attention of research on table recognition is focused on the analysis of the header zones and matrix kernel zones, and the identification and information zones are often left out.

To assess the performance of the table detection algorithms, the detection results are compared to the reference rectangles using the Intersection over Union (IoU) coefficient. Accuracy, completeness, and F-measure values averaged over the dataset can be evaluated either at several fixed IoU levels [86] or averaged over several IoU levels [85, 86]. The segmentation error classification system proposed in [88] can be used for a more detailed analysis. To assess the segmentation quality of the table area [85], a comparison with the benchmark structure of vertical and horizontal links between adjacent filled cells in the matrix structure is often used. In [89], this approach is criticized for its lack of sensitivity to errors caused by empty cells and mismatched cells outside of immediate neighbors, and for the fact that it does not accurately assess the quality of cell content (text) recognition. The proposed new function of the TEDS (Tree-Edit-Distance-based Similarity) quality of table recognition evaluation is based on the representation of the table in the form of a hierarchical tree structure (HTML table tree). The quality score is calculated based on the cost of editing (edit distance) of such a tree-like recognition result to bring it to the reference, which allows taking in to account both errors related to the structure and to the values of the terminal cells.

When considering open reference datasets for table recognition tasks, we can distinguish three main types:

- tables in digital (pdf, latex, word) sources (digitally-born documents), which can be represented as digital source files themselves or as generated (rendered) images of rectified pages;
- images of tables in printed documents;
- images of tables in handwritten and/or historical documents.

The images for the last two types were obtained mainly by scanning. Tab. 2 provides information on the main datasets related to table recognition.

Tab. 2. Information on datasets related to table recognition

Datasets	Year	Number of tables	Source	Format	Official split train/val/test
FUNSD [2] (IIT-CDIP [90])	2006, 2019	199 (15k)	scan	image	yes
UNLV [91]	2010	558	scan	image	no
Marmot [92]	2012	958	digital	image	no
ICDAR-2013 [84]	2013	254	digital	pdf	yes
CamCap [93]	2015	75	camera	image	no
ICDAR-2017 [85], TabStructDB [94]	2017, 2019	1081	digital	image	yes
ICDAR-2019 [86] modern	2019	340	digital	image	no
ICDAR-2019 [86] historical	2019	1099	scan	image	yes
ICDAR-2019 SROIE [95]	2019	1000	scan	image	yes
IIIT-AR-13K [96]	2020	16k+	digital	image	no

Open datasets mainly consist of images of tables in digital sources, such as articles and open business documents (IIIT-AR-13K), mostly financial reports. For paper documents, there are significantly fewer images, the documents themselves are archival, and the methods of obtaining images often are not modern, i.e. only monochrome and binarized scans (UNLV) are presented. The use of digital sources made it possible to dramatically increase the volume of sets, as well as to use automatically generated markup. A negative consequence of this approach is the insufficient accuracy of the data, as pointed out by the authors themselves. Therefore, in tab. 2, only the sets with manual markup are listed.

Note that the basic datasets listed above are of limited usefulness for studying the recognition of tables in documents. They do not contain images of tables on a complex background, typical for protected documents. There are very few obscured images with marks, signatures, seals, and stamps in the table area, etc. There are no open datasets with images of tables in documents obtained with mobile device cameras.

Extraction of table-structured information is still a relevant task and is of interest to researchers, despite the fact that the concept of a table itself is not clearly formulated. The use of neural network methods for detection has allowed reaching the industrial quality (with a probability of correct table recognition above 90 %), and the main focus of researchers has now shifted to the task of segmentation of tables in images of articles, books, reports, etc. For document recognition methods, the important tasks are the creation of representative open datasets with table images in documents, as well as the formulation of document-specific quality evaluation criteria.

1.8. Character and word recognition using artificial neural networks

Since the late 90s of the last century, most of the methods and approaches to optical character recognition are based on artificial neural networks (ANNs). In this section, we limit ourselves to the methods used for character string recognition in printed documents.

All methods used for line recognition can be divided into two large groups: methods with and without explicit segmentation of a text line into characters.

When segmenting a text line, a bounding rectangle is constructed for each character, and the contents of the former are passed to the classifier. In this group of methods, segmentation is considered a more complicated task than classification. All the standard segmentation algorithms experience difficulties when recognizing the linked or overlapping characters (when letters in some fonts merge into ligatures, for example, ff) or when processing characters consisting of more than one connected component (primitive). These problems are especially pronounced in the presence of a complex document background. This, in particular, is the basis for modern methods of CAPTCHA creation. Therefore, segmentation algorithms often contain heuristics that make them difficult to use for different languages and alphabets.

Text line segmentation algorithms can be divided into two groups:

- processing connected components;
- analysis of the projection on the horizontal axis.

Based on projection analysis, a heuristic redundant segmentation approach was developed. In this approach, several segmentation paths (several options for splitting a text line into characters) are constructed and the best one is chosen [97]. In this case, a text line can be deliberately divided into redundant parts (with character splitting), then each part is classified, and the best path is constructed based on the segmentation scores of the classifier [98]. Such methods sometimes use heuristics, such as the monospaced font feature [97] or the absence of characters made of several connectivity components [98]. ANN can be built into segmentation algorithms. In this case, there are two different approaches:

- using a classifier that knows how to distinguish a character from its part [99];
- using ANN to construct the projection itself, thereby eliminating the need to manually build additional heuristics and making the segmentation algorithm more robust to handwritten text [100].

After text line segmentation, the classification of character images is performed. Convolutional neural networks (CNN) are mostly used for the latter. There are also approaches based on cascade ANNs, when the final response is based on the estimates of all the networks in the cascade. To achieve the best results, cascades of networks with a very large number of parameters are now being composed. For example, the current lowest error rate on the MNIST dataset (0.14 %) is achieved by a cascade of 15 networks of 35.4×106 parameters each [101], while the human error rate is about 0.20 %. Thus, a lot of modern architectures contain an excessive number of parameters and are prone to overfitting [102].

With the emergence of claims about the instability of segmentation algorithms on distorted images, approaches without explicit character-by-character segmentation began to appear. Nowadays, most of such approaches are based on sliding window classifiers [103] and recurrent neural networks (RNNs) [104], particularly, LSTM (networks with long short-term memory blocks). The main advantage of RNNs is their ability to process and memorize sequences, but it can also be a disadvantage, for example, for text lines without language models, or when applying the network to another language with the same writing. Another approach to recognition without segmentation, whole word recognition, has two significant disadvantages: the number of classes becomes gigantic (by the number of words required), and it cannot be applied to text with a previously unknown set of words.

The text line recognition methods are often tailored to the peculiarities of writing. In this case, most studies have been and are being conducted for printed text with the English alphabet.

Languages based on Arabic writing present some difficulties for recognition due to the linked writing as well as the variety of forms [105]. In particular, the same word can be written with or without ligatures, depending on the font. One of the most difficult among modern languages in terms of recognition is Urdu in Nastaliq font, used in Pakistan, and used in identity documents as well.

A separate group is formed by languages where the number of classes to be recognized leads either to very heavy ANNs or to the processing of incomplete alphabets. Such languages primarily include Chinese, Japanese and Korean. Neural network methods of embedded learning [106] have been proposed for this group. Such ANNs also seem to be applicable to the alphabetic-syllabic writings of India and Southeast Asia, when ligatures of several characters are often formed.

Let us consider publicly available test sets for line recognition quality measurement. For accurate quality measurement, it is important that either the dataset consists of cut images of text lines [107], or the markup contains information on bounding rectangles (or general quadrangles) of lines [54], otherwise, errors of previous subsystems interfere in the quality measurements of text line recognition, as in the experiments with SmartDoc-2015. The aforementioned datasets contain examples for Latin. For Arabic, there is a number of datasets for text line recognition, for example, [108]. For languages of India and Southeast Asia, there are also separate datasets, but not all of them are publicly available [109]. The International Conference on Document Analysis and Recognition (ICDAR) regularly hosts various competitions involving text line recognition. In Thailand, the National Software Contest is regularly held, where datasets for Thai are provided. The closed datasets [97] are often used for experiments.

There may be handwritten or typewritten lines included in printed documents. Approaches to their recognition generally fall into two fundamentally different groups: online and offline approaches. In the first case, the text is recognized at the moment of its writing not only by image but also by hand movement and by the order of writing the line. But this mode is not typical for document recognition tasks. The second group is not fundamentally different from the approaches to the recognition of printed lines. The same approaches for recognition as described above are employed. At the same time, for handwritten text recognition, segmentation turns out to be even more difficult, and sometimes almost impossible to perform.

1.9. Recognition results post-processing

The result of text document recognition can contain various errors: incorrect recognition of one or more word characters, segmentation errors of word boundaries, punctuation errors. Recognition accuracy can be improved by recognition results post-processing. The post-processing algorithms usually include two stages: error detection and further error correction. Usually, these stages are closely related: detection involves detecting places where the recognition result does not satisfy the rules inherent in the algorithm, and correction, the choice of the best option that satisfies the rules.

The rules embedded in the post-processing algorithms usually employ the information on the syntactic and semantic structure of the data being recognized. Syntactic rules describe constructions that are acceptable in terms of the recognition language. Semantic rules are based on the semantic interpretation of the data. Thus, from the language point of view, errors can be divided into those that result in non-existing words, and those where the wrong recognition result is acceptable for the language but contradicts the grammatical rules or context.

Approaches to automatic post-processing of recognition results can work both on the level of characters, based on the syntax and semantics of the recognition language words, and on the level of words, taking into account the semantics of the links within recognized data. Mixed approaches may also be used. The automatic correction methods, which use semantic rules, can include dictionary methods [110]. Context-aware approaches are usually based on statistical language models, noisy channel models [111], N-grams [112]. Machine learning methods [112] can also be applied, and external resources can be used, such as, for example, calling the Google search engine with the spell check function [113] or the lexical database WordNet [114]. In addition, there are approaches that combine several post-processing methods to more accurately account for the specifics of the data being recognized.

Detection of recognition errors in the dictionary approach is to find out whether the recognized word is included in the accepted dictionary. At the correction stage, a list of variants of correct recognition is generated, from which the candidate with the highest score is then select-

ed. The evaluation of matches between the recognized word s_r and the dictionary word s_d can be performed, for example, based on the Levenshtein distance or the length of the longest common subsequence.

Dictionary methods are quite simple to implement but have a number of limitations. First, it is required to compile a dictionary covering all possible words and word forms that may occur in the recognized text. This significantly limits the applicability of dictionary methods for the correction of natural language texts, especially for languages with a complex morphological structure. Also, when compiling dictionaries, it is necessary to keep in mind the time period, the possibility of obsolete words and dictionary forms occurrence in the text. In addition, dictionary methods are not suitable for error correction when the wrong recognition result is present in the dictionary, but does not correspond to syntactic rules or context.

The editorial distance between words can be used in combination with the evaluation of the context similarity of two named entities, calculated according to some linguistic model. This approach is used to cluster incorrectly recognized words and phrases in order to further correct recognition errors [115].

Statistical N-gram post-processing models can work both at the characters level [116] and at the word level [112]. N-gram models are based on the assumption that the probability of occurrence of different sequences of characters or words in the recognized data is different. The main difficulties in the application of N-gram models are, first, in the compilation of the dictionary for N-gram models, i.e. a statistical model which adequately reflects the features of the recognized data. Machine learning approaches, such as the support vector machines [117], can be applied to improve the statistical model. Second, with large values of N, the computational complexity of the N-gram approach greatly increases. In practice, N-grams with an N value not exceeding 4 are usually employed.

Statistical methods of correction also include approaches based on the application of hidden Markov models [118]. This uses the assumption that the probability of occurrence of a character in a word or a word in the text depends on previous characters or words. Limiting factors for such statistical models are, first, that they are dependent on the language, and second, do not work well with texts that may contain non-canonical spelling, punctuation, etc. To solve the problem of error correction in such conditions, the recurrent neural networks, in particular, LSTM models, can be used [119].

The approach proposed in [120], based on Weighted Finite-State Transducers (WFST), combines a language model, an error model, and a hypothesis model. The approach based on the WFST allows building an algorithm for the correction of recognition results, not initially grounded under the specifics of the recognized data, and adapted by changing the semantic and syntactic rules. However, the difficulty of using this approach is the need for the information on the estimates distribution of characters belonging to recognition classes (for hypothesis model), but often this information is not available as the output of recognition. Another disadvantage of the approach is the high computational complexity of finding the optimal path in the transducer if the encoded language is complicated. In addition, the construction of a general language model in the form of a weighted finite-state transducer can also be quite difficult in some cases.

Easier to extend and implement, though less general, the approach where the language model is represented in the form of a verifying grammar. In this case, the problem of post-processing of recognition results is reduced to a discrete optimization problem. In [121], an effective algorithm for its solution is proposed.

Currently, there is a trend towards integrated approaches to the detection and correction of recognition errors, many of which use neural networks and machine learning techniques to tune post-processing algorithms to the data being recognized. At the same time, the use of large corpora, such as Google Web 1T, Google Book, and others, for the construction of language models remains relevant. Due to the widespread use of automatic text recognition technologies and, as a result, the high variability of the data, approaches that have the property of adaptivity, i.e. that make it possible to easily tune the algorithm for detection and correction of recognition results without changing the entire model, are of great interest [121].

2. Application of document image analysis and recognition methods

Let us consider several scenarios of document images recognition.

2.1. Extraction of attributes

The most popular scenario for processing the document recognition results is the extraction of attributes (fields). The extracted data is transferred to the document management system. For example, the extracted attributes together with the image can be stored in a digital archive.

To find the boundaries of the fields of documents with known geometric structure, methods of text segmentation based on descriptions (templates) can be used. For example, in [122] the application of a three-line template for text field extraction for bank card recognition is described.

For documents with a more flexible structure, algorithms of the "text in the wild" category that do not use a priori information about the geometry of the document can be applied. Such algorithms can be implemented using both artificial neural networks [123] and classical image processing methods [124], as well as their combinations.

In [125], the problem of a composite object segmentation with known constraints on the mutual arrangement of its elements is considered. In [72], the case when the constraint graph is a simple chain is considered, and by means of dynamic programming, the search of field boundaries for IDs and license plates is performed.

After the field boundaries are found, the lines contained within the field are recognized. Recognition depends on field attributes, such as alphabet, postprocessing type, and print or handwritten field characteristics. For all document types, if the table is found, the text is extracted from each table cell.

To extract fields from document images with a complex structure, such as free-form letters, algorithms for extracting data in OCR-recognized texts can be applied. The data to be extracted from texts usually includes the following objects:

- meaningful object: the name of a person, a company name, etc. for news reports; a subject area term for a special text; a reference to literature for scientific and technical documents, etc;
- attributes of the object, further characterizing it, for example, for a company this is the legal address, phone, name of the CEO, etc.
- the relationship between objects: for example, the relationship "to be the owner" connects the company and the owner, "to be part of" connects the faculty and the university;
- an event/fact that links several objects, e.g., the event "a meeting took place" includes meeting participants, as well as the place and time of the meeting.
- The main ways of data extraction are:
- named entities recognition and extraction;
- extraction of objects' attributes and semantic relations between them;
- extraction of facts and events covering several of their attributes.

2.2. Document sorting

In document management systems of large enterprises, one of the important tasks is sorting the flow of incoming documents in order to route them further. The task is usually complicated by the fact that the flow contains both single-page and multi-page documents. The purpose of document sorting is to divide a set of documents into subsets corresponding to different classes. Thus, the sorting task is based on page image classification. Subsequent processing of a document stream can be reduced to recognizing the image of a document of a known class. The page limits of documents in the incoming page stream may be unknown. In this case, classification of the first and the last page is required.

An extensive review of text document classification methods [3] considers streams generated by both stationary (scanners) and mobile devices. Inter-class similarity and intra-class document variability are pointed out as one of the main classification problems. Several groups of document classification methods are considered in [3]:

- textual methods;
- visual methods;
- structural methods.

Textual information analysis is based on global text descriptors such as Bag-of-Words (BOW), which is then analyzed by classical classifiers. The Bag-of-Words representation of a document can be replaced by more relevant models that take into account word order, such as statistically stable N-grams and vector word representations.

Effective methods of textual information analysis are the application of probabilistic topic models designed to determine the topics of a collection of documents by representing each topic by a discrete distribution of word probabilities and each document by a discrete probability distribution of topics [126]. When analyzing a document, topic modeling divides the document among several topics. Implicitly, it is assumed that the document contains a sufficient number of words to construct a discrete word probability distribution. In [127], additive regularization of topic models (BigARTM), i.e., multi-criteria optimization of a weighted sum of criteria, is described, which is necessary to refine and ensure the stability of the topic model construction over a collection of documents. Before constructing topic models of documents in natural language, normalization [127] is usually performed via several transformations: lemmatization, stemming, and others.

Datasets derived from recognized image sets or sets of articles originally existing in digital form can be used to train text-based methods. For example, the ever-expanding LitCovid dataset [128], currently consists of more than 130,000 COVID-19-related articles, categorized into 8 classes.

Structural methods explicitly use the structure of a document defined in its design. The image is segmented into several physical or logical components, then mapped to a set of templates or to a set of graph descriptions.

Template-matching methods use one or more templates that characterize each class. A similarity function between the image and the template is specified in advance. Byun and Lee [129] describe a method based on the location of lines in documents or in a given area of a document. In [130], Peng et al. propose to represent a document as a set of blocks (Component Block List, CBL). The comparison takes into account the size and location of the blocks, as well as the possible rotation of the document by an angle multiple of 90 degrees. Logical blocks such as title, author, or other text blocks can be represented as a fully connected Attributed Relational Graph (ARG) [131]. Further, graph matching techniques such as minimum weight edge cover or Earth Mover's Distance (EMD) are applied.

Unlike structural methods, visual methods implicitly describe the structure of a document image. Visual methods do not require preliminary image segmentation. Visual methods of image classification are divided into two categories: methods based on predefined features, and methods based on deep learning [3].

Predefined features are usually local features or keypoints, discussed in detail in Section 1.3. Depending on the expected composition of the document flow, these features are analyzed in the Bag-Of-Features model (also called BOVW - Bag-Of-Visual-Words in this task, by analogy with the BOW model for text descriptors). The

BOVW model allows for aggregation of local descriptors into a single vector but ignores the spatial location of "visual words", which leads to lower classification accuracy. In cases where it is important, keypoint coordinate anchoring methods are used to improve classification accuracy.

Identification documents flow is the best example in this regard [40]. Identification documents have a characteristic graphical structure in the form of fixed text zones (field labels: first name, last name, etc.), fields with variable text (personal data) and background elements of the template. The geometry of such documents is usually fixed. Therefore, at the training stage, keypoints corresponding to the fixed part of the image are selected for each class of documents based on their geometric position. In the simplest cases, one image of each class is sufficient for this purpose. The learned features together with the reference coordinates and class labels are indexed using tree search structures. During the recognition phase, the descriptors of the keypoints are constructed for the query image, which are then classified by the search tree. The final decision is made after checking the most likely classes to match the arrangement of local features. Usually, the RANSAC method is employed for this purpose because it allows for a high degree of invariance to projective distortions and at the same time provides robustness to the inevitable errors in the classification of individual keypoints.

But the most popular direction of document image classification is algorithms using convolutional neural networks. The typical architecture of CNNs used for this task includes two components. The first component consists of convolutional and subsampling layers and forms the input vector of deep features. The second component of CNNs is a multilayer perceptron, which classifies the feature vector.

In [132], four convolutional neural networks including AlexNet, VGG-16, GoogLeNet, and ResNet-50 were investigated for document image classification. It is argued that the model pre-trained on a set of ImageNet annotated images performs better than when using standard random initialization.

Obviously, for successful training of convolutional neural networks, a large number of samples is required. Below are short descriptions of known datasets:

- RVL-CDIP [133] includes 400,000 images;
- database NIST 2 (NIST-SPDB2) [134] includes 5590 binary images of tax forms of 20 classes with typed or handwritten filling;
- dataset Tobacco-3482 [135] includes 3482 images of 10 classes: report, memo, resume, scientific, letter, news, note, advertisement, form, and e-mail address.

Text-based methods employ optical character recognition (OCR) results. Due to text recognition errors, the use of text descriptors may result in decreased classification accuracy.

Hybrid methods combine textual methods with analysis of visual and/or structural features to classify document images. To date, such methods usually provide the highest classification accuracy due to such combination, since these two types of features contain complementary information. In [136], an architecture for page classification in a document stream usual in banks was proposed. Both visual and text descriptors were used. Visual descriptors were computed based on pixel intensity distribution. Text descriptors were generated based on latent semantic analysis to represent the content of the document as a combination of topics. The authors evaluated several off-the-shelf classifiers and various strategies for combining visual and textual representations. The proposed method was tested on a set of real business documents of 70,000 pages. The best accuracy obtained by combining several classification methods was 95.6 %.

2.3. Document comparison

Document images comparison is relevant when checking the correctness of signed paper documents, for example, when two parties sign contracts and agreements [137]. In this case, a detailed analysis of the document contents is required, as a change of even a single character may become critical for disputing the deal.

Possible changes to the document may relate to both the content of the document and the layout of the document (font style, color, and text size), spatial arrangement of elements (line spacing). It is possible to add and delete content elements, including text, figures, graphs, tables, handwritten content (notes, signatures), stamps.

One approach to comparing two document images is to use recognition. The simplest comparison method is text recognition using OCR followed by the diff utility based on the longest common subsequence (LCS) search to compare the recognition results for two documents [138]. The disadvantages of this method are a large number of false positives due to recognition errors and loss of information about the font, color, and text size. Also, text recognition cannot be used to compare seals, figures, and other non-text elements present in the document images.

Another approach, which does not rely on character recognition, uses descriptors with pre-segmentation of text into text lines. In [137], dense SIFT descriptors were used. Segmentation of the document image not only into text lines but also into characters, as proposed in [139], can also be used. The results presented in this work show that the proposed method can handle images of multilingual documents with different resolutions and font sizes.

Another method for document images comparison relies on the visual similarity of documents. In [140], a visual similarity measure is proposed for document comparison, which uses document layout and text characteristics derived from text primitives: text block complexity, based on entropy, and clarity, dealing with font boldness. This measure of document image similarity can be used to classify documents and sort out similar documents.

Document comparison is also employed for duplicates or near-duplicates search, for example, when constructing large datasets, documents corpora. In this case, different

definitions of duplicates can be considered. In some cases, duplicates may refer to versions of a document obtained under different conditions [141]. Near-duplicates are, for example, images with the same textual content but different handwritten notes [142].

All duplicate detectors perform two operations [143]. The first is the signature generation, which represents the image as a relatively small amount of information. The second operation is signature matching: all pairs of signatures are compared to detect duplicates. In [138, 141 – 147], the search for duplicates in a database of raster binarized images of documents, mostly fax documents, is considered. The primary coarse detector is based on the number of black pixels, and the secondary detector is based on pattern matching.

The methods used for duplicates or near-duplicates search can be divided into 3 groups [138, 141, 142, 144 – 146]:

- recognition-based methods that represent a document as a set of words [144];
- methods based on dividing a document image into paragraphs /columns to extract information about the shape of the objects within the document. Also, when dividing a document into lines and words, information about the peculiarities of the words writing in the image can be extracted [140];
- methods which rely on common image features [142].

There are four types of duplicates discussed in [144]: full-layout (if two documents are visually the same), full-content (if two documents have the same content but not necessarily the same template), partial-layout (if two documents have much of the same content with the same template), and partial-content (if two documents have common content but their template is not necessarily the same).

When comparing two documents, first, the recognition is performed, then the two unprocessed recognized texts are compared. In the case of content duplication (full-content and partial-content duplicates), the document can be represented as a string of characters sequence. In the case of template duplication (full-layout and partial-layout duplicates), the document is represented as a sequence of lines. Different variations of the algorithm for the largest common sequence (LCS) search as well as the edit distance are used to compare the recognized texts for these four cases.

In [142], the search for near-duplicates was performed in a database of Arabic documents, and the documents could be either printed with different handwritten notes in the duplicates or completely handwritten. Therefore, methods based on text segmentation into lines and words as well as methods using recognition are not applicable in this case. The approach proposed in this paper is based on keypoint matching using SIFT.

[141] considered the detection of near-duplicates: images obtained from the same document but under different conditions. To compare two document images, a multi-granularity tree of objects is constructed for each of them, and several graphs were generated. Then graphs are compared pairwise, and the similarity of the document images evaluation is reduced to the evaluation of maximum similarity of these graphs. Each level in the tree is associated with one possible object segmentation, while different levels are characterized by different degrees of object detail.

Document images comparison can be employed to determine whether an image is a forgery, according to a predetermined set of data [145, 146]. Images of documents coming from a single source are considered, e.g., clinic bills, payrolls, payslips, etc. Such documents do not contain additional security features (such as watermarks) but have a similar structure. Based on the available set of genuine documents, treated as a training sample, the features or peculiarities of the given document images are identified. Next, the algorithm receives some image as an input and it determines whether it is a fake document. It identifies certain features in the input image and compares them with those identified in the set of genuine documents. The most important distortion, which is considered in [145], is the uneven vertical scaling when reprinting a scanned document. In [146], an approach for aligning the documents from the same source is proposed.

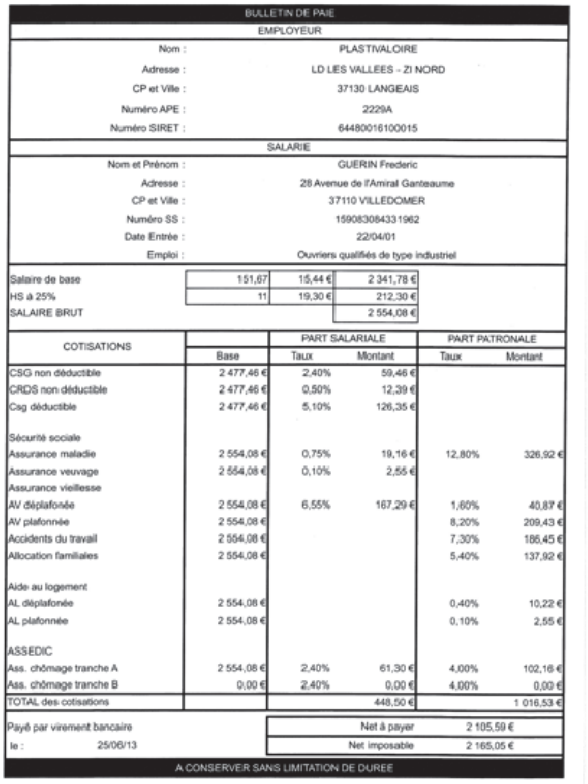
A common problem in counterfeit detection is the lack of public datasets for algorithms evaluation. First, it is natural that forgers do not want to disclose their methods and the types of forgeries they have made. Second, most of the documents that are being modified contain personal information and are confidential. In [147], the authors addressed this problem by synthesizing "real" documents that were subsequently forged by volunteers. The public dataset includes a corpus of 477 forged (modified) Payslips where about 6,000 characters were modified. Fig. 4 illustrates an example of image from this dataset.

2.4. Video sequence recognition

Currently, capturing document images with a smartphone camera or webcam [148] seems preferable from the user's point of view compared to using scanners. From the developer's point of view, by contrast, the mobile document recognition mode is much more complex.

When capturing with a camera, at the very least, the document is worse, and (usually) unevenly illuminated, and its image is subjected to projective distortion. In addition to these disadvantages, mobile cameras also have an advantage over scanners: they can capture a video stream which can contain frames with different illumination, from different angles, with different characteristics of focus, that allows reducing sporadic errors of OCR-system [1].

When working with a video stream, the problem of combining information extracted from different frames of a video sequence arises. Methods of combining frame-by-frame information can be divided into two groups: methods based on combining images to obtain a better representation of the object, and methods of combining extracted text recognition results.



BULLETIN DE PAIE

EMPLOYEUR

Nom : PLASTIVALOIRE
Adresse : LD LES VALLEES - ZI NORD
CP et Ville : 37130 LANGEAIS
Numéro APE : 2229A
Numéro SIRET : 64480016100015

SALARIE

Nom et Prénom : GUERIN Frederic
Adresse : 28 Avenue de l'Amiral Ganteaume
CP et Ville : 37110 VILLEDOMER
Numéro SS : 159083084331962
Date Entrée : 22/04/01
Emploi : Ouvriers qualifiés de type industriel

Salaire de base	151,67	15,44 €	2 341,78 €
HS à 25%	11	19,30 €	212,30 €
SALAIRE BRUT			2 554,08 €

COTISATIONS	Base	PART SALARIALE Taux	PART SALARIALE Montant	PART PATRONALE Taux	PART PATRONALE Montant
CSG non déductible	2 477,46 €	2,40%	59,46 €		
CRDS non déductible	2 477,46 €	0,50%	12,39 €		
Csg déductible	2 477,46 €	5,10%	126,35 €		
Sécurité sociale					
Assurance maladie	2 554,08 €	0,75%	19,16 €	12,80%	326,92 €
Assurance veuvage	2 554,08 €	0,10%	2,55 €		
Assurance vieillesse					
AV déplafonnée	2 554,08 €	6,55%	167,29 €	1,60%	40,87 €
AV plafonnée	2 554,08 €			8,20%	209,43 €
Accidents du travail	2 554,08 €			7,30%	186,45 €
Allocation familiales	2 554,08 €			5,40%	137,92 €
Aide au logement					
AL déplafonnée	2 554,08 €			0,40%	10,22 €
AL plafonnée	2 554,08 €			0,10%	2,55 €
ASSEDIC					
Ass. chômage tranche A	2 554,08 €	2,40%	61,30 €	4,00%	102,16 €
Ass. chômage tranche B	0,00 €	2,40%	0,00 €	4,00%	0,00 €
TOTAL des cotisations			448,50 €		1 016,53 €

Payé par virement bancaire	Net à payer	2 105,59 €
le : 25/06/13	Net imposable	2 165,05 €

A CONSERVER SANS LIMITATION DE DUREE

Fig. 4. An example of image for forgery detection (from Payslips dataset [127-141])

The first group includes methods for selecting the most informative frame [149], "super-resolution" methods that create a higher quality image based on several low-resolution frames [150], methods for tracking and combining images of a recognized object in a sequence of frames [151], methods of blur compensation by replacing blurred areas in one frame with their clearer versions taken from other frames or using deep learning techniques [152]. Also, data from various sensors of a mobile device, such as an accelerometer or gyroscope, can be used to better the recovery of a recognized document image. However, for modern mobile devices, the error in their measurements can be very significant and prevent the use of such data for image reconstruction. Disadvantages of methods within the first group include computational complexity, sensitivity to geometric distortions of frames, and poor scalability in terms of video sequences of arbitrary length.

The second group of methods involves combining the results of individual recognition of document images. A distinctive feature of the text objects recognition is that the text line is a composite object, i.e. it consists of several characters. In identity document recognition systems, the result of text recognition is treated as a composition of the character classification results. Such a representation implies a preliminary procedure of text segmentation into characters, i.e. the process of splitting the image of a text line into images of special characters. For such text representation, the model of combined per-frame recognition results has to deal with text lines obtained for different frames, and in case of segmentation errors such text lines would have different lengths, and the combining algorithm should be able to take this into account. One of the combining approaches, which allows the input of text lines with variable lengths, is the ROVER (Recognizer Output Voting Error Reduction) method [153]. This method was originally created to improve the quality of speech recognition by combining the recognition results from different systems. This method includes two steps. In the first step, all the combined recognition results are aligned by inserting an empty character and combined into a single transient network. In the second step, a voting procedure is used to select the best recognition result for each element of the combined object. The voting procedure can be considered as a classifiers combination problem, and various models for classifiers ensemble such as rules of sum, product, maximum, median, etc., can be employed as an extension of the voting procedure in ROVER. Thus, using the ROVER method to combine the recognition results obtained from several frames allows for correct recognition results, even if in some frames the text field has been incorrectly segmented into characters. In [154], combining algorithms for frame-by-frame text recognition, which take into account weights for input results, are described.

3. Optimizing the performance of document recognition algorithms

Document recognition is based on a combination of algorithms with different mathematical complexity. The performance of recognition implementation on any computing architecture should be optimized for several reasons. First, to save time and other resources, and second, to improve the ergonomics of user applications. And finally, for platforms with limited resources, such as Internet of Things (IoT) devices and mobile devices, optimizing performance is directly related to optimizing power consumption and heat dissipation.

One of the most popular classification mechanisms is artificial neural networks. ANNs can be used in almost all stages of document recognition. ANN models designed for classification with high accuracy usually require large computational resources and energy, since often ANNs are based on a large number of multiplication operations. Therefore, optimizing the performance of ANNs is an urgent task. Let us consider several popular approaches for ANNs performance optimization.

The first group of approaches relies on integer arithmetic instead of real arithmetic [155]. This approach improves performance and saves memory in exchange for some, usually insignificant, decrease of accuracy. Generally, the reduction of neural network models accuracy during their optimization remains a debatable issue. Currently, there are no theoretical estimates of such decrease for any of the proposed optimization technologies. In practical applications, results depend both on the size and architecture of the initial network, and on a particular

problem to be solved. Therefore, it is interesting to note, [156] shows that the use of 4-bit architectures particularly for character recognition gives a significant performance gain while retaining the acceptable accuracy.

Another group of approaches is focused on binary networks (BNN), in which some of the layers contain binary coefficients. A group of methods for avoiding overflows, such as Normalization Layer Design, Small-pipeline Rule, and Aggregated Convolutional Operation [157], was proposed to achieve acceptable accuracy in BNN training.

The application of low-precision real arithmetic (FP8) and low-precision hybrid arithmetic (HFP8) also allows for successfully trained DNN. [158] demonstrated that it is possible to quantize a pre-trained model to 8-bit format without loss of accuracy.

A promising approach is the replacement of the convolution and fully connected layers with low-rank tensor approximations. Methods such as tensor decomposition, tensor sequence, tensor ring, and modified polyadic tensor decomposition showed compression efficiency with a slight decrease in accuracy [159].

The discussed approaches are based on the quantization reduction of ANN coefficients and on an approximation of nonlinear functions. Additionally, low-level optimization for SIMD command systems is reasonable.

Discussion

Currently, document image recognition tools are characterized by a systematic approach to document image processing. While for some tasks (text line detection, document classification, document comparison) in addition to classical multi-step algorithms it has been already reasonably proposed to use end-to-end deep learning methods, document detection as a whole is almost invariably considered in the context of recognition systems. This is apparently due to the great variability of the document detection problem, i.e. various document image capturing conditions (scanning, photo, or video), and diverse constituents of a document stream. The latter includes various classes which differ significantly in volume and degree of structured information: identification documents often have a fixed template and content geometry, however, accuracy requirements are more stringent in the case of ID recognition; the invoices recognition quality to a large extent depends on the table analysis algorithms; multi-page weakly structured documents without the explicitly numbered pages (e.g., letters) significantly complicate even such seemingly simple tasks as determining the first and last page of the document within a document stream.

In the systematic approach, the task of DIA is divided into separate subtasks of data normalization (elimination of projective distortions, binarization), recognition (of a text line, class of document), or clarification due to a priori (statistical post-processing of text) or redundant (integration via a video stream) information, and each subtask is being solved relatively independently. This approach, first, avoids the combinatorial explosion when determining the limit of the applicability. For example, the projective normalization subtask hardly depends on the document language, and the text line recognition subtask does not really depend on the projective distortion degree prior to the normalization step. Such assumptions allow for a drastic reduction in the work effort when generating test (and training) datasets. Second, a well-structured unified system can be easily configured to handle document streams that are very different in constituents and capturing conditions. Modern end-to-end deep learning methods do not possess such qualities so far.

Thus, at the present day, DIA cannot be reduced exclusively to OCR. Hence, the considered groups of methods, such as document image normalization or boundary detection are not directly related to the topic of character recognition but are related to, for example, the computer vision for autonomous vehicles. Therefore, in the modern context, document detection should rather be considered one of computer vision directions.

Despite the observation on the systematic approach, the main trend of document image processing and recognition algorithms development is the use of machine learning methods, primarily ANNs. It is clear from the review that ANNs are successfully used for the majority of digital document imaging tasks. A new promising direction in the development of ANNs application is to consider and optimize the computational efficiency of the proposed models: due to the need to perform recognition with devices of limited resources or low performance.

Algorithms for document images processing and recognition have a long history of development. Nevertheless, a large number of papers are still published annually on this topic. There are specialized conferences (ICDAR) and journals (The International Journal on Document Analysis and Recognition, IJDAR) dedicated to document image processing and recognition. This demonstrates the relevance of the topic. Its development manifests in both the gradual improvement of known models and methods, and the emergence of new tasks. For example, digital document imaging via mobile devices has been introduced fairly recently. Therefore, some of the new methods are aimed at eliminating the distortions inherent to mobile images. Digitization of documents captured with a video camera provides an opportunity to improve the quality of recognition, but this requires the development of data integration and post-processing methods. As for the directions that have already become classic, for recognition and attribute extraction of poorly structured documents, especially of those containing tables, solutions with acceptable accuracy and robustness have not been proposed yet.

Both the current and further development of the automatic analysis and recognition of document images would be impossible without a gradual expansion of the open data sets available to the scientific community. We

have mentioned the most significant among the present ones. Currently, if a well-labeled large dataset is available, methods that allow for high accuracy on such a dataset usually appear in 1-2 years. A new trend in this area is the constant updating of datasets (e.g., for the DIB task or various MIDV recognition tasks). This approach allows to steadily expand the applicability domain of the proposed methods and, more importantly, to control the overfitting effects of the booming deep learning methods.

Acknowledgments

The reported study was funded by RFBR, project number 20-17-50177. The authors thank Sc. D. Vladimir L. Arlazarov (FRC CSC RAS), Pavel Bezmaternykh (FRC CSC RAS), Elena Limonova (FRC CSC RAS), Ph. D. Dmitry Polevoy (FRC CSC RAS), Daniil Tropin (LLC "Smart Engines Service"), Yuliya Chernysheva(LLC "Smart Engines Service"), Yuliya Shemyakina (LLC "Smart Engines Service") for valuable comments and suggestions.

References

[1] Arlazarov V, Bulatov K, Chernov T, Arlazarov VL. MIDV-500: a dataset for identity document analysis and recognition on mobile devices in video stream. Computer Optics 2019; 43(5): 818-824. DOI: 10.18287/2412-6179-2019-43-5-818-824.

[2] Jaume G, Ekenel HK, Thiran J. FUNSD: A dataset for form understanding in noisy scanned documents. Int Conf on Document Analysis and Recognition Workshops (ICDARW) 2019; 2: 1-6. DOI: 10.1109/ICDARW.2019.10029.

[3] Liu L, Wang Z, Qiu T, Chen Q, Lu Y, Suen CY. Document image classification: Progress over two decades. Neurocomputing 2021; 453: 223-240. DOI: 10.1016/j.neucom.2021.04.114.

[4] Baviskar D, Ahirrao S, Potdar V, Kotecha K. Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions. IEEE Access 2021; 9: 72894-72936. DOI: 10.1109/ACCESS.2021.3072900.

[5] Hull JJ. Document image skew detection: Survey and annotated bibliography. In Book: Hull JJ, Taylor SL, eds. Document analysis systems II. London: World Scientific Publishing Co; 1998: 40-64. DOI: 10.1142/9789812797704_0003.

[6] Rehman A, Saba T. Document skew estimation and correction: Analysis of techniques, common problems and possible solutions. Appl Artif Intell 2011; 25(9): 769-787. DOI: 10.1080/08839514.2011.607009.

[7] Chen D, Luettin J, Shearer K. A survey of text detection and recognition in images and videos. Institute Dalle Molle d'Intelligence Artificielle Perceptive Research Report 2000: 00-38.

[8] Nagy G. Twenty years of document analysis in PAMI. IEEE Trans Pattern Anal Mach Intell 2000; 22(1): 38-62. DOI: 10.1109/34.824820.

[9] Mao S, Rosenfeld A, Kanungo T. Document structure analysis algorithms: a literature survey. Proc SPIE 2003; 5010: 197-207. DOI: 10.1117/12.476326.

[10] Doermann D, Liang J, Li H. Progress in camera-based document image analysis. Seventh Int Conf on Document Analysis and Recognition 2003; 1: 606-616. DOI: 10.1109/ICDAR.2003.1227735.

[11] Zanibbi R, Blostein D, Cordy J. A survey of table recognition. Int J Doc Anal Recognit 2004; 7: 1-16. DOI: 10.1007/s10032-004-0120-9.

[12] Jung K, Kim K, Jain A. Text information extraction in images and video: A survey. Pattern Recognit 2004; 37: 977-997. DOI: 10.1016/j.patcog.2003.10.012.

[13] Liang J, Doermann D, Li H. Camera-based analysis of text and documents: a survey. Int J Doc Anal Recognit 2005; 7: 84-104. DOI: 10.1007/s10032-004-0138-z.

[14] Marinai S, Gori M, Soda G. Artificial neural networks for document analysis and recognition. IEEE Trans Pattern Anal Mach Intell 2005; 27(1): 23-35. DOI: 10.1109/TPAMI.2005.4.

[15] Chen N, Blostein D. A survey of document image classification: problem statement, classifier architecture and performance evaluation. Int J Doc Anal Recognit 2007; 10: 1-16. DOI: 10.1007/s10032-006-0020-2.

[16] Baharudin B, et al. A review of machine learning algorithms for text-documents classification. J Adv Inf Technol 2010; 1: 4-20.

[17] Dixit U, Shirdhonkar M. A survey on document image analysis and retrieval system. Int J Cybern Inform 2015; 4: 259-270. DOI: 10.5121/ijci.2015.4225.

[18] Eskenazi S, Gomez-Krämer P, Ogier JM. A comprehensive survey of mostly textual document segmentation algorithms since 2008. Pattern Recognit 2017; 64: 1-14.

[19] Binmakhashen GM, Mahmoud SA. Document layout analysis: A comprehensive survey. ACM Comput Surv 2019; 52(6): 109.

[20] Lombardi F, Marinai S. Deep learning for historical document analysis and recognition–A survey. J Imaging 2020; 6: 110. DOI: 10.3390/jimaging6100110.

[21] Bhatt J, Hashmi KA, Afzal MZ, Stricker D. A survey of graphical page object detection with deep neural networks. Appl Sci 2021; 11(12): 5344. DOI: 10.3390/app11125344.

[22] Doermann D, Tombre K. Handbook of document image processing and recognition. Springer Publishing Company Inc; 2014.

[23] Liu CL, Lu Y, eds. Advances in Chinese document and text processing. World Scientific; 2017. ISBN: 978-981-3143-67-8.

[24] Fischer A, Liwicki M, Ingold R. Handwritten historical document analysis, recognition, and retrieval – state of the art and future trends. World Scientific Publishing Co Pte Ltd; 2021.

[25] SJR. Scimago Journal & Country Rank. Proc Int Conf on Document Analysis and Recognition (ICDAR). Source: ⟨https://www.scimagojr.com/journalsearch.php?q=75898&tip=sid⟩.

[26] Bloomberg DS, Kopec GE, Dasari L. Measuring document image skew and orientation. Proc SPIE 1995; 2422: 302-316. DOI: 10.1117/12.205832.

[27] Steinherz T, Intrator N, Rivlin E. Skew detection via principal components analysis. Proc Fifth Int Conf on Document Analysis and Recognition. ICDAR '99 (Cat. No. PR00318) 1999: 153-156. DOI: 10.1109/ICDAR.1999.791747.

[28] Bezmaternykh P, Nikolaev DP. A document skew detection method using fast Hough transform. Proc SPIE 2020; 114330: 114330J. DOI: 10.1117/12.2559069.

[29] Akhter SSMN, Rege PP. Improving skew detection and correction in different document images using a deep learning approach. 2020 11th Int Conf on Computing,

Communication and Networking Technologies (ICCCNT) 2020: 1-6. DOI: 10.1109/ICCCNT49239.2020.9225619.
[30] Papandreou A, Gatos B, Louloudis G, Stamatopoulos N. ICDAR 2013 document image skew estimation contest (DISEC 2013). 2013 12th Int Conf on Document Analysis and Recognition 2013: 1444-1448. DOI: 10.1109/ICDAR.2013.291.
[31] Fabrizio J. A precise skew estimation algorithm for document images using KNN clustering and fourier transform. 2014 IEEE Int Conf on Image Processing (ICIP) 2014: 2585-2588. DOI: 10.1109/ICIP.2014.7025523.
[32] Uchida S, Taira E, Sakoe H. Nonuniform slant correction using dynamic programming. Proc Sixth Int Conf on Document Analysis and Recognition 2001: 434-438. DOI: 10.1109/ICDAR.2001.953827.
[33] Otsu N. Threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern Syst 1979; SMC-9(1): 62-66. DOI: 10.1109/tsmc.1979.4310076.
[34] Lu S, Su B, Tan CL. Document image binarization using background estimation and stroke edges. Int J Doc Anal Recognit 2010; 13(4): 303-314. DOI: 10.1007/s10032-010-0130-8.
[35] Gatos B, Pratikakis I, Perantonis SJ. Adaptive degraded document image binarization. Pattern Recognit 2006; 39(3): 317-327. DOI: 10.1016/j.patcog.2005.09.010.
[36] Ershov EI, Korchagin SA, Kokhan VV, Bezmaternykh PV. A generalization of Otsu method for linear separation of two unbalanced classes in document image binarization. Computer Optics 2021; 45(1): 66-76. DOI: 10.18287/2412-6179-CO-752.
[37] Calvo-Zaragoza J, Gallego A-J. A selectional auto-encoder approach for document image binarization. Pattern Recognit 2019; 86: 37-47. DOI: 10.1016/j.patcog.2018.08.011.
[38] Bezmaternykh PV, Ilin DA, Nikolaev DP. U-Net-bin: hacking the document image binarization contest. Computer Optics 2019; 43(5): 825-832. DOI: 10.18287/2412-6179-2019-43-5-825-832.
[39] Document image binarization. Source: ⟨https://dib.cin.ufpe.br⟩.
[40] Skoryukina N, Arlazarov V, Nikolaev D. Fast method of id documents location and type identification for mobile and server application. IEEE Int Conf on Document Analysis and Recognition (ICDAR) 2019: 850-857. DOI: 10.1109/ICDAR.2019.00141.
[41] Challenge 1: Smartphone document capture competition. Source: ⟨https://sites.google.com/site/icdar15smartdoc/challenge-1⟩.
[42] Schmid C, Mohr R. Local grayvalue invariants for image retrieval. IEEE Trans Pattern Anal Mach Intell 1997; 19(5): 530-535. DOI: 10.1109/34.589215.
[43] Harris C, Stephens M. A combined corner and edge detector. Alvey Vision Conference 1988: 147-151. DOI: 10.5244/C.2.23.
[44] Rosten E, Drummond T. Machine learning for high-speed corner detection. In Book: Leonardis A, Bischof H, Pinz A, eds. Computer vision – ECCV 2006. Part 1. Berlin, Heidelberg: Springer-Verlag; 2006: 430-443. DOI: 10.1007/11744023_34.
[45] Lowe DG. Distinctive image features from scale-invariant keypoints. Int J Comput Vis 2004; 60(2): 91-110. DOI: 10.1023/B%3AVISI.0000029664.99615.94.
[46] Lepetit V, Fua P. Towards recognizing feature points using classification trees. Technical report, Swiss Federal Institute of Technology (EPFL), 2004. Source: ⟨https://infoscience.epfl.ch/record/52666⟩.
[47] Bay H, EssTinne A, Tuytelaars T, Gool LV. Speeded-up robust features (SURF). Comput Vis Image Underst 2008; 110(3): 346-359. DOI: 10.1016/j.cviu.2007.09.014.
[48] Rosin PL. Measuring corner properties. Comput Vis Image Underst 1999; 73(2): 291-307. DOI: 10.1006/cviu.1998.0719.
[49] Leutenegger S, Chli M, Siegwart RY. BRISK: Binary robust invariant scalable keypoints. IEEE Int Conf on Computer Vision (ICCV) 2011: 2548-2555. DOI: 10.1109/ICCV.2011.6126542.
[50] Zhang H, Wohlfeil J, Grießbach D. Extension and evaluation of the AGAST feature detector. ISPRS Ann Photogramm Remote Sens Spat Inf Sci 2016; III(4): 133-137. DOI: 10.5194/isprsannals-III-4-133-2016.
[51] Verma R, Kaur R. Enhanced character recognition using surf feature and neural network technique. Int J Comput Sci Inf Technol Res 2014; 5(4): 5565-5570.
[52] Dang OB, Coustaty M, Luqman MMM, Ogier J-M. A comparison of local features for camera-based document image retrieval and spotting. Int J Doc Anal Recognit 2019; 22: 247-263. DOI: 10.1007/s10032-019-00329-w.
[53] Lewis D, Agam G, Argamon S, Frieder O, Grossman D. Building a test collection for complex document information processing. Proc 29th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval (SIGIR '06) 2006: 665-666. DOI: 10.1145/1148170.1148307.
[54] Bulatov K, Matalov D, Arlazarov VV. MIDV-2019: Challenges of the modern mobile-based document OCR. Proc SPIE 2019; 11433: 114332N. DOI: 10.1117/12.2558438.
[55] University of California, San Francisco: The Legacy Tobacco Document Library (LTDL) 2007. Source: ⟨http://legacy.library.ucsf.edu⟩.
[56] Zhang Z, He L-W. Whiteboard scanning and image enhancement. Digit Signal Process 2007; 17(2): 414-432. DOI: 10.1016/j.dsp.2006.05.006.
[57] Liu N, Wang L. Dynamic detection of an object framework in a mobile device captured image. US Patent 10134163 of November 20, 2018.
[58] Hartl A, Reitmayr G. Rectangular target extraction for mobile augmented reality applications. The 21st Int Conf on Pattern Recognition (ICPR 2012) 2012: 81-84.
[59] Skoryukina N, Nikolaev DP, Sheshkus A, Polevoy D. Real time rectangular document detection on mobile devices. Proc SPIE 2014; 9445: 94452A. DOI: 10.1117/12.2181377.
[60] Tropin DV, Ilyuhin SA, Nikolaev DP, Arlazarov VV. Approach for document detection by contours and contrasts. IEEE Int Conf on Pattern Recognition (ICPR) 2020: 9689-9695. DOI: 10.1109/ICPR48806.2021.9413271.
[61] Hua G, Liu Z, Zhang Z, Wu Y. Automatic business card scanning with a camera. IEEE Int Conf on Image Processing (ICIP) 2006: 373-376. DOI: 10.1109/ICIP.2006.312471.
[62] Xu Y, Carlinet E, Géraud T, Najman L. Hierarchical segmentation using tree-based shape spaces. IEEE Trans Pattern Anal Mach Intell 2017; 39(3): 457-469. DOI: 10.1109/TPAMI.2016.2554550.
[63] Attivissimo F, Giaquinto N, Scarpetta M, Spadavecchia M. An automatic reader of identity documents. IEEE Int Conf on Systems, Man and Cybernetics (SMC) 2019: 3525-3530. DOI: 10.1109/SMC.2019.8914438.
[64] Castelblanco A, Solano J, Lopez C, Rivera E, Tengana L, Ochoa M. Machine learning techniques for identity document verification in uncontrolled environments: A case study. Springer Mexican Conference on Pattern Recognition (MCPR) 2020: 271-281. DOI: 10.1007/978-3-030-49076-8_26.

[65] Sheshkus A, Nikolaev D, Arlazarov VL. Houghencoder: neural network architecture for document image semantic segmentation. IEEE Int Conf on Image Processing (ICIP) 2020: 1946-1950. DOI: 10.1109/ICIP40778.2020.9191182.

[66] Javed K, Shafait F. Real-time document localization in natural images by recursive application of a CNN. IEEE IAPR Int Conf on Document Analysis and Recognition (ICDAR) 2017: 105-110. DOI: 10.1109/ICDAR.2017.26.

[67] das Neves RB, Felipe Verçosa L, Macêdo D, Dantas Bezerra BL, Zanchettin C. A fast fully octave convolutional neural network for document image segmentation. IEEE Int Joint Conf on Neural Networks (IJCNN) 2020: 1-6. DOI: 10.1109/IJCNN48605.2020.9206711.

[68] Viola P, Jones M. Robust real-time object detection. Int J Comput Vis 2002; 57: 137-154.

[69] Usilin S, Nikolaev D, Postnikov V, Schaefer G. Visual appearance based document image classification. 2010 IEEE Int Conf on Image Processing 2010: 2133-2136. DOI: 10.1109/ICIP.2010.5652024.

[70] Roy PP, Pal U, Llados J. Seal detection and recognition: An approach for document indexing. 10th Int Conf on Document Analysis and Recognition 2009: 101-105. DOI: 10.1109/ICDAR.2009.128.

[71] Wang Y, Zhou Y, Tang Z. Comic frame extraction via line segments combination. 13th Int Conf on Document Analysis and Recognition (ICDAR) 2015: 856-860. DOI: 10.1109/ICDAR.2015.7333883.

[72] Povolotskiy MA, Tropin DV. Dynamic programming approach to template-based OCR. Proc SPIE 2019; 11041: 110411T. DOI: 10.1117/12.2522974.

[73] Slavin OA. Using special text points in the recognition of documents. In Book: Kravets AG, Bolshakov AA, Shcherbakov MV, eds. Cyber-physical systems: Advances in design & modelling. Cham: Springer Nature Switzerland AG; 2020: 43-53. DOI: 10.1007/978-3-030-32579-4_4.

[74] Shafait F, Breuel TM. The effect of border noise on the performance of projection-based page segmentation methods. IEEE Trans Pattern Anal Mach Intell 2011; 33(4): 846-851. DOI: 10.1109/TPAMI.2010.194.

[75] Melinda L, Ghanapuram R, Bhagvati C. Document layout analysis using multigaussian fitting. 14th IAPR Int Conf on Document Analysis and Recognition (ICDAR) 2017: 747-752. DOI: 10.1109/ICDAR.2017.127.

[76] Yi X, Gao L, Liao Y, Zhang X, Liu R, Jiang Z. CNN based page object detection in document images. 14th IAPR Int Conf on Document Analysis and Recognition (ICDAR) 2017: 230-235. DOI: 10.1109/ICDAR.2017.46.

[77] Kosaraju SC, Masum M, Tsaku NZ, Patel P, Bayramoglu T, Modgil G, Kang M. DoT-Net: Document layout classification using texture-based CNN. Int Conf on Document Analysis and Recognition (ICDAR) 2019: 1029-1034. DOI: 10.1109/ICDAR.2019.00168.

[78] He D, Cohen S, Price B, Kifer D, Giles CL. Multi-scale multi-task FCN for semantic page segmentation and table detection. 14th IAPR Int Conf on Document Analysis and Recognition (ICDAR) 2017: 254-261. DOI: 10.1109/ICDAR.2017.50.

[79] Wu Y, Wang W, Palaiahnakote S, Lu T. A robust symmetry-based method for scene/video text detection through neural network. 14th IAPR Int Conf on Document Analysis and Recognition (ICDAR) 2017: 1249-1254. DOI: 10.1109/ICDAR.2017.206.

[80] Antonacopoulos A, Bridson D, Papadopoulos C, Pletschacher S. A realistic dataset for performance evaluation of document layout analysis. 10th Int Conf on Document Analysis and Recognition 2009: 296-300. DOI: 10.1109/ICDAR.2009.271.

[81] Veit A, Matera T, Neumann L, Matas J, Belongie S. COCO-Text: Dataset and benchmark for text detection and recognition in natural images. arXiv Preprint 2016. Source: ⟨https://arxiv.org/abs/1601.07140⟩.

[82] Brunessaux S, Giroux P, Grilheres B, Manta M, Bodin M, Choukri K, Galibert O, Kahn J. The Maurdor Project: Improving automatic processing of digital documents. 11th IAPR Int Workshop on Document Analysis Systems 2014: 349-354. DOI: 10.1109/DAS.2014.58.

[83] Soares AS, Neves RB, Bezerra BLD. BID Dataset: a challenge dataset for document processing tasks. Conf on Graphics, Patterns and Images (SIBGRAPI) 2020. DOI: 10.5753/sibgrapi.est.2020.12997.

[84] Göbel M, Hassan T, Oro E, Orsi G. ICDAR 2013 table competition. 12th Int Conf on Document Analysis and Recognition 2013: 1449-1453. DOI: 10.1109/ICDAR.2013.292.

[85] Gao L, Yi X, Jiang Z, Hao L, Tang Z. ICDAR 2017 competition on page object detection. 14th IAPR Int Conf on Document Analysis and Recognition (ICDAR) 2017; 1: 1417-1422. DOI: 10.1109/ICDAR.2017.231.

[86] Gao L, et al. ICDAR 2019 competition on table detection and recognition (cTDaR). Int Conf on Document Analysis and Recognition (ICDAR) 2019: 1510-1515. DOI: 10.1109/ICDAR.2019.00243.

[87] Costa e Silva A, Jorge AM, Torgo L. Design of an end-to-end method to extract information from tables. Int J Doc Anal Recognit 2006; 8: 144-171. DOI: 10.1007/s10032-005-0001-x.

[88] Shafait F, Smith R. Table detection in heterogeneous documents. 9th IAPR Int Workshop on Document Analysis Systems 2010: 65-72. DOI: 10.1145/1815330.1815339.

[89] Zhong X, ShafieiBavani E, Yepes AJ. Image-based table recognition: data, model, and evaluation. arXiv Preprint 2019. Source: ⟨https://arxiv.org/abs/1911.10683⟩.

[90] Lewis D, Agam G, Argamon S, Frieder O, Grossman D, Heard J. Building a test collection for complex document information processing. 29th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval 2006: 665-666. DOI: 10.1145/1148170.1148307.

[91] Shahab A, Shafait F, Kieninger T, Dengel A. An open approach towards the benchmarking of table structure recognition systems. 9th IAPR Int Workshop on Document Analysis Systems 2010: 113-120. DOI: 10.1145/1815330.1815345.

[92] Fang J, Tao X, Tang Z, Qiu R, Liu Y. Dataset, ground-truth and performance metrics for table detection evaluation. 10th IAPR Int Workshop on Document Analysis Systems 2012: 445-449. DOI: 10.1109/DAS.2012.29.

[93] Seo W, Koo HI, Cho NI. Junction-based table detection in camera-captured document images. Int J Doc Anal Recognit 2014; 18(1): 47-57. DOI: 10.1007/s10032-014-0226-7.

[94] Siddiqui SA, Fateh IA, Rizvi STR, Dengel A, Ahmed S. DeepTabStR: Deep learning based table structure recognition. Int Conf on Document Analysis and Recognition (ICDAR) 2019: 1403-1409. DOI: 10.1109/ICDAR.2019.00226.

[95] Huang Z, Chen K, He J, Bai X, Karatzas D, Lu S, Jawahar CV. ICDAR 2019 competition on scanned receipt ocr and information extraction. Int Conf on Document Analysis and Recognition (ICDAR) 2019: 1516-1520. DOI: 10.1109/ICDAR.2019.00244.

[96] Mondal A, Lipps P, Jawahar CV. IIIT-AR-13K: A new dataset for graphical object detection in documents. In Book: Bai X, Karatzas D, Lopresti D, eds. Document analysis sys-

tems. Cham: Springer International Publishing; 2020: 216-230. DOI: 10.1007/978-3-030-57058-3_16.
[97] Jia F, Shi C, Wang Y, Wang C, Xiao B. Grayscale-projection based optimal character segmentation for camera-captured faint text recognition. 2017 Int Conf on Document Analysis and Recognition 2017: 1301-1306. DOI: 10.1109/ICDAR.2017.214.
[98] Roy PP, Pal U, Lladós J, Delalandre M. Multi-oriented touching text character segmentation in graphical documents using dynamic programming. Pattern Recognit 2012; 45(5): 1972-1983. DOI: 10.1016/j.patcog.2011.09.026.
[99] Saba T, Rehman A. Effects of artificially intelligent tools on pattern recognition. Int J Mach Learn Cybern 2013; 4: 155-162. DOI: 10.1007/s13042-012-0082-z.
[100] Chernyshova YS, Sheshkus AV, Arlazarov VV. Two-step CNN framework for text line recognition in camera-captured images. IEEE Access 2020; 8: 32587-32600. DOI: 10.1109/ACCESS.2020.2974051.
[101] Alvear-Sandoval RF, Sancho-Gómez JL, Figueiras-Vidal AR. On improving CNNs performance: The case of MNIST. Inf Fusion 2019; 52: 106-109. DOI: 10.1016/j.inffus.2018.12.005.
[102] Zhang C, Bengio S, Hardt M, Recht B, Vinyals O. Understanding deep learning (Still) requires rethinking generalization. Commun ACM 2021; 64(3): 107-115. DOI: 10.1145/3446776.
[103] Bahi E, Zatni A. Text recognition in document images obtained by a smartphone based on deep convolutional and recurrent neural network. Multimed Tools Appl 2019; 78(18): 26453-26481. DOI: 10.1007/s11042-019-07855-z.
[104] Rubner Y, Tomasi C, Guibas LJ. The earth mover's distance as a metric for image retrieval. Int J Comput Vis 2000; 40 (2): 99-121.
[105] Elarian Y, Ahmad I, Awaida S, Al-Khatib W, Zidouri A. Arabic ligatures: Analysis and application in text recognition. 2015 13th Int Conf on Document Analysis and Recognition (ICDAR) 2015: 896-900. DOI: 10.1109/ICDAR.2015.7333891.
[106] Ilyuhin SA, Sheshkus AV, Arlazarov VL. Recognition of images of Korean characters using embedded networks. Twelfth Int Conf on Machine Vision (ICMV 2019) 2020; 114330: 1143311. DOI: 10.1117/12.2559453.
[107] Kišš M, Hradiš M, Kodym O. Brno mobile OCR dataset. 2019 Int Conf on Document Analysis and Recognition (ICDAR) 2019: 1352-1357. DOI: 10.1109/ICDAR.2019.00218.
[108] Doush IA, AlKhateeb F, Gharibeh AH. Yarmouk arabic OCR dataset. 2018 8th Int Conf on Computer Science and Information Technology (CSIT) 2018: 150-154. DOI: 10.1109/CSIT.2018.8486162.
[109] Mathew M, Singh AK, Jawahar CV. Multilingual OCR for Indic Scripts. 2016 12th IAPR Workshop on Document Analysis Systems (DAS) 2016: 186-191. DOI: 10.1109/DAS.2016.68.
[110] Guo C-Y, Tang YY, Liu C-S, Duan J. A japanese OCR post-processing approach based on dictionary matching. Int Conf on Wavelet Analysis and Pattern Recognition 2013: 22-26. DOI: 10.1109/ICWAPR.2013.6599286.
[111] Kissos I, Dershowitz N. OCR error correction using character correction and feature-based word classification. 12th IAPR Workshop on Document Analysis Systems (DAS) 2016: 198-203. DOI: 10.1109/DAS.2016.44.
[112] Mei J, Islam A, Wu Y, Moh'd A, Milios EE. Statistical learning for OCR text correction. arXiv Preprint 2016. Source: ⟨http://arxiv.org/abs/1611.06950⟩.
[113] Bassil Y, Alwani M. OCR post-processing error correction algorithm using google online spelling suggestion. arXiv Preprint. Source: ⟨https://arxiv.org/abs/1204.0191⟩.
[114] Eutamene A, Kholladi MK, Belhadef H. Ontologies and bigram-based approach for isolated non-word errors correction in OCR system. Int J Electr Comput Eng 2015; 5(6): 1458-1467. DOI: 10.11591/ijece.v5i6.pp1458-1467.
[115] Jean-Caurant A, Tamani N, Courboulay V, Burie JC. Lexicographical-based order for post-OCR correction of named entities. Int Conf on Document Analysis and Recognition (ICDAR) 2018: 1192-1197. DOI: 10.1109/ICDAR.2017.197.
[116] Bulatov K, Manzhikov T, Slavin O, Faradjev I, Janiszewski I. Trigram-based algorithms for OCR result correction. Proc SPIE 2017; 10341: 103410O. DOI: 10.1117/12.2268559.
[117] Fonseca Cacho JR, Taghva K. OCR post processing using support vector machines. In Book: Arai K, Kapoor S, Bhatia R, eds. Intelligent computing. Proceedings of the 2020 computing conference. Vol 2. Cham: Springer Nature Switzerland AG; 2020: 694-713. DOI: 10.1007/978-3-030-52246-9_51.
[118] Bouchaffra D, Govindaraju V, Srihari SN. Postprocessing of recognized strings using nonstationary markovian models. IEEE Trans Pattern Anal Mach Intell 1999; 21(10): 990-999. DOI: 10.1109/34.799906.
[119] Saluja R, Punjabi M, Carman M, Ramakrishnan G, Chaudhuri P. Sub-word embeddings for OCR corrections in highly fusional indic languages. Int Conf on Document Analysis and Recognition (ICDAR) 2019: 160-165. DOI: 10.1109/ICDAR.2019.00034.
[120] Llobet R, Navarro-Cerdan JR, Perez-Cortes JC, Arlandis J. OCR post-processing using weighted finite-state transducers. Int Conf on Pattern Recognition 2010: 2021-2024. DOI: 10.1109/ICPR.2010.498.
[121] Bulatov KB, Nikolaev DP, Postnikov VV. General-purpose algorithm for text field OCR result post-processing based on validation grammars [In Russian]. Trudy Instituta Sistemnogo Analiza RAN 2015; 65(4): 68-73.
[122] Sheshkus A, Nikolaev DP, Ingacheva A, Skoryukina N. Approach to recognition of flexible form for credit card expiration date recognition as example. Proc SPIE 2015; 9875: 98750R. DOI: 10.1117/12.2229534.
[123] Wang K, Belongie S. Word spotting in the wild. In Book: Daniilidis K, Maragos P, Paragios N, eds. Computer vision – ECCV 2010. Berlin, Heidelberg: Springer-Verlag; 2010: 591-604. DOI: 10.1007/978-3-642-15549-9_43.
[124] Epshtein B, Ofek E, Wexler Y. Detecting text in natural scenes with stroke width transform. 2010 IEEE Computer Society Conf on Computer Vision and Pattern Recognition 2010: 2963-2970. DOI: 10.1109/CVPR.2010.5540041.
[125] Felzenszwalb PF, Zabih R. Dynamic programming and graph algorithms in computer vision. IEEE Trans Pattern Anal Mach Intell 2011; 33(4): 721-740. DOI: 10.1109/TPAMI.2010.135.
[126] Rubin TN, Chambers A, Smyth P, Steyvers M. Statistical topic models for multi-label document classification. Machine Learning 2011; 88(1): 157-208. DOI: 10.1007/s10994-011-5272-5.
[127] Vorontsov KV. Additive regularization for topic models of text collections [In Russian]. Doklady Mathematics 2014; 89(3): 301-304. DOI: 10.1134/S1064562414020185.
[128] Chen Q, Allot A, Lu Z. Keep up with the latest coronavirus research. Nature 2020; 579(7798): 193. DOI: 10.1038/d41586-020-00694-1.

[129] Byun Y, Lee Y. Form classification using DP matching. ACM Symposium on Applied Computing 2000; 1: 1-4. DOI: 10.1145/335603.335611.
[130] Peng HC, Long FH, Chi ZR, Siu W-C. Document image template matching based on component block list. Pattern Recognit Lett 2001; 22: 1033-1042. DOI: 10.1016/S0167-8655(01)00049-6.
[131] Liang J, Doermann D, Ma M, Guo J. Page classification through logical Labeling. 2002 Int Conf on Pattern Recognition 2002; 3: 477-480. DOI: 10.1109/ICPR.2002.1047980.
[132] Afzal MZ, Kölsch A, Ahmed S, Liwicki M. Cutting the error by half: Investigation of very deep CNN and advanced training strategies for document image classification. Int Conf on Document Analysis and Recognition 2017; 1: 883-888. DOI: 10.1109/ICDAR.2017.149.
[133] RVL-CDIP-I Dataset. Source: ⟨https://www.kaggle.com/nbhativp/first-half-training⟩.
[134] NIST Special Database 2. Source: ⟨https://www.nist.gov/srd/nist-special-database-2⟩.
[135] Tobacco-3482. Source: ⟨https://www.kaggle.com/patrickaudriaz/tobacco3482jpg⟩.
[136] Rusiñol M, Frinken V, Karatzas D, Bagdanov AD, Lladós J. Multimodal page classification in administrative document image streams. Int J Doc Anal Recognit 2014; 17: 331-341. DOI: 10.1007/s10032-014-0225-8.
[137] Jain R, Doermann D. Localized document image change detection. 13th Int Conf on Document Analysis and Recognition (ICDAR) 2015: 786-790. DOI: 10.1109/icdar.2015.7333869.
[138] Lopresti DP. A comparison of text-based methods for detecting duplication in scanned document databases. Inf Retr J 2001; 4: 153-173. DOI: 10.1023/A:1011471129047.
[139] Lin Y, Li Y, Song Y, et al. Fast document image comparison in multilingual corpus without OCR. Multimed Syst 2017; 23: 315-324. DOI: 10.1007/s00530-015-0484-3.
[140] Eglin V, Bres S. Document page similarity based on layout visual saliency: application to query by example and document classification. Seventh Int Conf on Document Analysis and Recognition 2003: 1208-1212. DOI: 10.1109/ICDAR.2003.1227849.
[141] Liu L, Lu Y, Suen CY. Near-duplicate document image matching: A graphical perspective. Pattern Recognit 2014; 47(4): 1653-1663. DOI: 10.1016/j.patcog.2013.11.006.
[142] Vitaladevuni S, Choi F, Prasad R, Natarajan P. Detecting near-duplicate document images using interest point matching. 21st Int Conf on Pattern Recognition (ICPR2012) 2012: 347-350.
[143] Caprari RS. Duplicate document detection by template matching. Image Vis Comput 2000; 18(8): 633-643. DOI: 10.1016/s0262-8856(99)00086-4.
[144] Lopresti DP. Models and algorithms for duplicate document detection. Fifth Int Conf on Document Analysis and Recognition, ICDAR '99 (Cat. No. PR00318) 1999: 297-300. DOI: 10.1109/ICDAR.1999.791783.
[145] Ahmed AGH, Shafait F. Forgery detection based on intrinsic document contents. 11th IAPR Int Workshop on Document Analysis Systems 2014: 252-256. DOI: 10.1109/DAS.2014.26.
[146] Beusekom J, Shafait F, Breuel TM. Document signature using intrinsic features for counterfeit detection. In Book: Srihari SN, Franke K, eds. Computational forensics. Berlin, Heidelberg: Springer-Verlag; 2008: 47-57. DOI: 10.1007/978-3-540-85303-9_5.
[147] Sidere N, Cruz F, Coustaty M, Ogier JM. A dataset for forgery detection and spotting in document images. Seventh Int Conf on Emerging Security Technologies (EST) 2017: 26-31. DOI: 10.1109/EST.2017.8090394.
[148] Ôn Vũ Ngoc M, Fabrizio J, Géraud T. Document detection in videos captured by smartphones using a saliency-based method. Int Conf on Document Analysis and Recognition Workshops (ICDARW) 2019: 19-24. DOI: 10.1109/ICDARW.2019.30059.
[149] Zhanzhan C, Jing L, Yi N, Shiliang P, Fei W, Shuigeng Z. You only recognize once: Towards fast video text spotting. 27th ACM Int Conf 2019: 855-863. DOI: 10.1145/3343031.3351093.
[150] Deudon M, Kalaitzis A, Goytom I, Arefin MdR, Lin Z, Sankaran K, Michalski V, Kahou SE, Cornebise J, Bengio Y. HighRes-Net: Multi-frame super-resolution by recursive fusion. ICLR 2020 Conf. Source: ⟨https://openreview.net/forum?id=HJxJ2h4tPr⟩.
[151] Cheng Z, Lu J, Xie J, Niu Y, Pu S, Wu F. Efficient video scene text spotting: Unifying detection, tracking, and recognition. arXiv Preprint 2019. Source: ⟨https://arxiv.org/abs/1903.03299⟩.
[152] Zhang S, Li P, Meng Y, Li L, Zhou Q, Fu X. A video deblurring algorithm based on motion vector and an encorder-decoder network. IEEE Access 2019; 7: 86778-86788. DOI: 10.1109/ACCESS.2019.2923759.
[153] Fiscus JG. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings 1997: 347-354. DOI: 10.1109/ASRU.1997.659110.
[154] Bulatov K, Arlazarov V, Chernov T, Slavin O, Nikolaev D. Smart IDReader: Document recognition in video stream. 14th IAPR Int Conf on Document Analysis and Recognition (ICDAR) 2017; 6: 39-44. DOI: 10.1109/ICDAR.2017.347.
[155] Elhoushi M, Chen Z, Shafiq F, Tian YH, Li JY. DeepShift: Towards multiplication-less neural networks. arXiv Preprint 2020. Source: ⟨https://https://arxiv.org/pdf/1905.13298.pdf⟩.
[156] Trusov AV, Limonova EE, Slugin DG, Nikolaev DP, Arlazarov VV. Fast implementation of 4-bit convolutional neural networks for mobile devices. 2020 25th Int Conf on Pattern Recognition (ICPR) 2021: 9897-9903. DOI: 10.1109/ICPR48806.2021.9412841.
[157] Li J, Wang Y, Liu B, Han Y, Li X-W. Simulate-the-hardware: training accurate binarized neural networks for low-precision neural accelerators. 24th Asia and South Pacific Design Automation Conf 2019: 323-328. DOI: 10.1145/3287624.3287628.
[158] Sun X, Choi J, Chen C-Y, Wang N, Venkataramani S, Srinivasan VV, Cui X, Zhang W, Gopalakrishnan K. Hybrid 8-bit floating point (HFP8) training and inference for deep neural networks. Adv Neural Inf Process Syst 2019; 32: 4901-4909.
[159] Phan AH, et al. Stable low-rank tensor decomposition for for compression of convolutional neural network. In Book: Vedaldi A, Bischof H, Brox T, Frahm J-M, eds. Computer Vision – ECCV 2020. Part XXIX. Cham: Springer Nature Switzerland AG; 2020: 522-539. DOI: 10.1007/978-3-030-58526-6_31.

Authors' information

Vladimir Viktorovich Arlazarov, (b. 1976), graduated from Moscow Institute of Steel and Alloys in 1999, majoring in Applied Mathematics. Received his Ph.D. degree in 2005. Currently he works as a head of division 93 at the Federal Research Center «Computer Science and Control» of RAS. Research interests are pattern recognition, computer vision, intelligent systems, and machine learning. E-mail: *vva@smartengines.com* .

Elena Igorevna Andreeva, (b. 1995), graduated from Moscow Institute of Physics and Technology with Master Degree in 2019, majoring in Applied Mathematics and Physics. Currently she works as the researcher-programmer at the LLC "Smart Engines Service". Research interests: computer vision, image processing, intelligent document processing. E-mail: *andreeva@phystech.edu* .

Konstantin Bulatovich Bulatov, (b. 1991), graduated from National University of Science and Technology «MISiS» in 2013, majoring in Applied Mathematics, and received his Ph.D. degree in 2020. Currently he works as a senior researcher a the Federal Research Center «Computer Science and Control» of RAS. Research interests include pattern recognition, combinatorial optimization, computer vision, document analysis.
E-mail: *kbulatov@smartengines.com* .

Dmitry Petrovich Nikolaev, (b. 1978), graduated from Lomonosov Moscow State University (MSU) in 2000, majoring in Physic. Ph. D. in Physics and Mathematics, is a head of the vision systems laboratory at the Institute for Information Transmission Problems. Research interests are machine vision, algorithms for fast image processing, pattern recognition. E-mail: *dimonstr@iitp.ru* .

Olga Olegovna Petrova, (b. 1994), graduated from Moscow Institute of Physics and Technology (State University) in 2019, majoring in Applied Mathematics and Informatics. Currently she is a postgraduate student at the Federal Research Center «Computer Science and Control» of RAS. Research interests are computer science, document analysis and recognition. E-mail: *opetrova@smartengines.com* .

Boris Igorevich Savelev, (b. 1995), graduated from National University of Science and Technology MISIS with Master Degree in 2018, majoring in Innovative IT Projects. Currently he works as the researcher-programmer at the LLC "Smart Engines Service" (Moscow, Russian Federation), lead programmer at the FRC CSC RAS (Moscow, Russian Federation). Research interests are computer vision, image processing, intelligent document processing.
E-mail: *saveliev@smartengines.com* .

Oleg Anatolevich Slavin, (b. 1963.), graduated from Moscow Institute Radiotechnics, Electronics and Automation (MIREA), majoring in Systems Engineering. Sc.D. in Technics Currently he works as a head of division 92 at the Federal Research Center «Computer Science and Control» of RAS. Research interests are pattern recognition, computer vision and information systems. E-mail: *oslavin@isa.ru* .

Received August 5, 2021. The final version – October 26, 2021.