https://doi.org/10.2352/ISSN.2470-1173.2017.2.VIPC-412

Document Image Classification on the Basis of Layout Information

Sergey Zavalishin; Samsung R&D Institute Russia; Moscow, Russia
Andrey Bout; Kaspersky Lab; Moscow, Russia
Ilya Kurilin; Samsung R&D Institute Russia; Moscow, Russia
Michail Rychagov; Samsung R&D Institute Russia; Moscow, Russia

Abstract

In this paper, we propose a document image classification framework based on layout information. Our method does not use OCR; hence, it is completely language independent. Still we are able to exploit text data by extracting text regions with a novel MSER-based approach. Our MSER formulation provides great robustness against text distortions in comparison to the existing one. We introduce two types of novel image descriptors supplemented with Fisher vectors, based on Bernoulli mixture model. Classifiers, based on aforementioned descriptors, are assembled into meta-classification system that is able to classify document in complex cases when individual classifier accuracy is poor. Our meta-classification system demonstrates low processing time comparable to a single classifier. We show that our method outperforms the existing ones by the means of classification accuracy for a wide range of documents of both well-known and machine-generated document datasets.

1. Introduction

Scanned document classification is an important task in various document management systems, such as business processing workflows, digital libraries, multifunctional devices and so forth (fig. 1). Many of existing approaches [1] focus on textual information, as this is an essential data. However, there are many situations for business documents when the amount of text in a document is relatively small or even absent or includes multi-language and handwritten text, which is difficult to recognize. Thus, some prior papers propose using text information along with the visual one to improve classification accuracy [2].

The most of business documents have pre-defined structure, which makes it possible to classify them on the basis of layout similarity. Existing methods rely on layout extraction in the form of XY-trees or region features. These methods are good for binary document images, but they lack of robustness in the case of complex documents with complicated background and distortions.

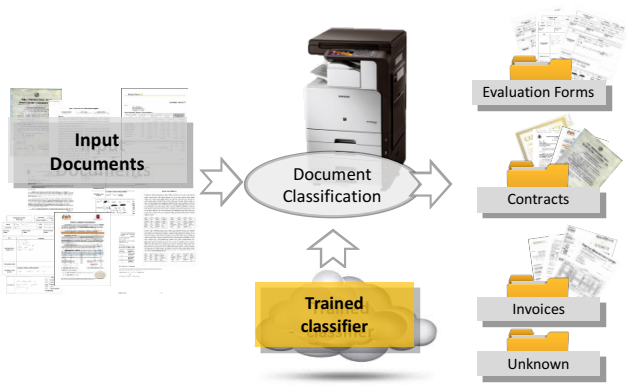

Fig. 1. Document classification in content management systems

In this paper, we propose a robust method for document images classification based on novel image descriptors of three types: a) Spatial Local Binary Pattern (SLBP), b) Grayscale Runlength Histogram (GRLH) and c) BRISK descriptors aggregated with Fisher vectors based on Bernoulli Mixture Model (BMMFV). These descriptors efficiently encode spatial document structure that provides layout-based classification without necessity to extract document layout tree. Our framework is able to extract text in OCR-free manner. It is achieved by using a novel formulation of MSER, MSER-SI, which can extract small or highly distorted text characters. A meta-classifier model aggregates proposed descriptors in order to improve classification performance in complex cases, when individual classifiers cannot provide sufficient accuracy. We show that proposed classification framework demonstrates low processing time comparable to a single classifier.

2. Related Work

There are several existing approaches [3] for OCR-free document image classification assembled into two major groups. The first group includes structure-based methods, which attempt to extract document layout directly and encode it with graphs, trees, or feature vectors. The second group uses general image representation methods, such as local and global image descriptors, convolutional neural networks, Fisher vectors and so forth.

Structure-based methods use XY-trees [4, 5] to encode document structure mainly. The major advantage of XY-trees is that they can encode the structure directly. A root of the tree is the document itself, leaves are text or image blocks and edges are block relationships. The disadvantages are obvious: block extraction relies on binarization, which is hard to achieve in the case of documents with complex backgrounds.

Another disadvantage is complex tree comparison: method [6] introduces a special grammar that makes it possible to compare trees in the form of text strings. Methods [7-9] convert XY-trees into a fixed-length feature vectors; hence, comparison is performed in the same way as for general image feature vectors. Finally, method [10] creates unique network model for each document class using Winnow algorithm and compares each document to these models one by one.

The recent methods are mainly based on general image features enhanced with document-specific spatial information. The most trivial ones utilize gray pixel density supplemented with connected components [11], document lines [12], table sizes and positions [13], text strings [14], Viola-Jones features [15] or runlength histograms [16]. These methods are fast and simple, but they rely on document binarization mainly, which makes them useless in the case of complex background presence.

Method [17] uses DTMSER transform to encode document structure, which is a combination of MSER regions and distance

transform. [18] is based on SIFT descriptors that are clusterized on the basis of regular grid. Similar, [19] exploits SURF descriptors clusterized using horizontal and vertical stripes. These methods are robust enough, but as we show further, using local descriptors only may lead to accuracy degradation in the case of different document backgorunds.

Finally, one of the most advanced methods [20] is based on Fisher vectors (FV) encoding. First, it extracts SIFT descriptors and trains Gaussian Mixture Model (GMM). Using GMM, SIFTs are clusterized into descriptor histograms that are utilized for FV computation. In addition, authors propose calculating FV for several sub-images with further concatenation of them into a single feature vector, which provides more spatial information. Evolution of this method introduces a hybrid scheme [21], where extracted FVs are used as an input for a pre-trained deep network. Both of these approaches demonstrate state-of-the-art classification accuracy, but extraction of such complex features is a time-consuming task.

That is, we propose a novel document classification framework that introduces an extremely flexible scheme, which combines OCR-free text extraction, image-based feature vector extraction and classifier ensembling. Our ensembling model is able to classify the most of the documents using a single classifier. It uses several classifiers for complex documents only that significantly reduces processing time.

The rest of the paper is organized as follows. In section 3 we introduce our approach to document classification. In subsection 3.1 we give proposed classification framework overview; in subsection 3.2 we describe our approach for text extraction; in subsection 3.3 we explain proposed GRLH and SLBP descriptors and outline Fisher vectors based on Bernoulli Mixture Model; and in subsection 3.4 we give detailed explanation of our classification framework. Section 4 is dedicated to test setup and results. Subsection 4.1 describes datasets we use, including well-known ones and generated by ourselves. Finally, in subsection 4.2 we demonstrate the advantages of the proposed algorithms.

3. Proposed Approach

3.1. Processing pipeline overview

In this section, we will discuss proposed processing pipeline. Our pipeline consists of three main stages: 1) text extraction; 2) preliminary classification with a single classifier; and 3) meta-classification, which ensembles decisions of several classifiers in order to provide the final decision (fig. 2).

Text extraction is an important pre-processing step. We propose a novel MSER-based approach for text regions detection with further classification of detected regions into text and non-text. The classification is performed using Grayscale Runlength Histogram (GRLH) descriptor, which is our modification of well-known RLH descriptor [16]. In contrast to RLH, GRLH extracts features directly from grayscale images, while RLH needs the binary ones; hence, it does not suffer from binarization errors.

Document classification is based on three different visual feature types: GRLH, Spatial Local Binary Pattern (SLBP), and Fisher Vectors based on Bernoulli Mixture Model (BMMFV). All three descriptors use spatial pyramid extracted from grayscale image; hence, we totally avoid binarization in our pipeline.

Intuition behind using these particular descriptors is the following: run length histograms were proven to be fast and simple, yet robust, for document classification, so they are good choice for classifying relatively simple documents. Fisher vectors are rather complex, but more robust to class intra variety and document distortions. SLBP is our extension to a classic Local Binary Pattern (LBP) descriptor, which is primarily used for local pixel patterns encoding. We improve it by extracting spatial pyramid of sub-images and further scaling each sub-image to the same, relatively small, size. Scaling provides implicit histogram normalization, which significantly improves classification accuracy in comparison to L1 or L2 normalization performed on non-scaled images.

SLBP provides lower classification accuracy than GRLH and BMMFV, so we use it for ensembling only. Using the third classifier is necessary due to ambiguity in the cases when two classifiers yield very different results.

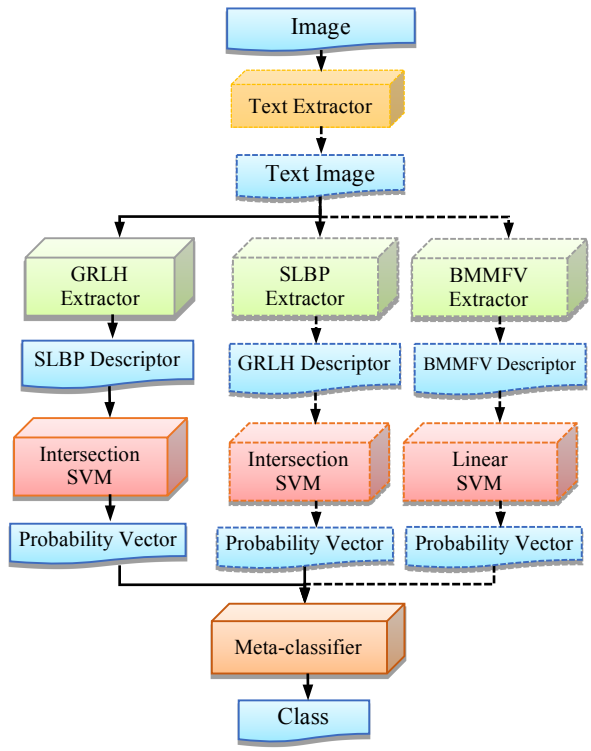

Fig. 2. General flowchart of the proposed method. Optional blocks are outlined using dashes.

Meta-classification addresses two problems. First, it attempts to provide the highest possible classification accuracy rate by utilizing classifier ensembling and, second, preserve high computational speed comparable to the single classifier. We achieve this goal by two-step processing: on the first step we perform classification with GRLH and estimate obtained prediction likehood. In the case if the likehood is low, we estimate prediction probabilities using SLBP and BMMFV based classifiers and pass them to a Support Vector Machine (SVM) classifier as concatenated feature vector. SVM yields the final prediction.

In contrast to the most of the existing algorithms, our classification pipeline may also mark document as “unclassified” and pass it to the user for manual classification. That is, we implement active learning paradigm. In real document classification system, the number of initial training images is small. It is hard to provide a large collection of documents for each class, especially if the number of classes is large; hence, active

learning provides ability to significantly improve classification accuracy by adding misclassified samples to the training set.

Another advantage of the proposed pipeline is flexibility: in our implementation, the only mandatory classifier is GRLH-based one, while all the other blocks, including text extractor, meta-classifier and other individual classifiers may be omitted for simple documents.

3.2. Text extraction

A typical way to extract text regions from image is using MSER [22]. It was shown that MSER regions are good representation of text characters [23]; hence using them for text candidate detection is a widely adopted technique.

MSER detects Extremal Regions (ERs) that are defined as following:

$$\mathrm{Q} = \{\forall \mathrm{p} \in \mathrm{Q}, \forall \mathrm{q} \in \partial \mathrm{Q}\colon \mathrm{I(p)} < \mathrm{I(q)}\}, \qquad (1)$$

where Q is a region, ∂Q is a boundary of Q, and I(x) is a brightness of pixel x. Maximal Stable Extremal Regions (MSERs) are those regions among enclosed ERs $\mathrm{Q_1} \subset \cdots \subset \mathrm{Q_{i-1}} \subset \mathrm{Q_i} \subset \cdots$, which meet the following condition:

$$\mathrm{q(i)} = |\mathrm{Q_{i+\Delta} \backslash Q_{i-\Delta}}|/|\mathrm{Q_i}|. \qquad (2)$$

MSER calculation is based on image binarizaion with different brightness thresholds. Connected component blobs found for each threshold are considered as ERs. ERs, which areas remain almost the same for several thresholds, are called stable. That is, they are MSERs by definition. This formulation is valid for a large-scale text, but document text characters may be too small to be stable (fig. 3).

Fig. 3. Comparison of photography text (left) and low quality document text (right).

To overcome this issue, we propose using different formulation of ER regions, MSER-SI (MSER Supremum/Infimum):

$$\mathrm{Q'} = \{\forall \mathrm{p} \in \mathrm{Q}, \forall \mathrm{q} \in \partial \mathrm{Q}\colon \mathrm{inf(p)} < \mathrm{inf(q)}\}, \qquad (3)$$

where infimum is taken in a NxN window around the pixel.

Typically, MSERs are detected for both normal image and its inverse in order to find dark and light characters. In the last case, the difference between boundary and non-boundary pixels is determined as $\sup(\mathrm{p}) > \sup(\mathrm{q})$.

One can see that this formulation leads to detection of region that encloses character, but not the character itself. Characters can be found by binarization of each region with Otsu algorithm, but there is no need to perform it as we use grayscale-based descriptors.

Found text regions are considered as text candidates. Thus, we need to classify them into text and non-text. We propose using GRLH descriptor that is calculated as following.

First, we scan image lines in four directions, D_h, D_v, D_{d+}, D_{d-}, and look for runs that defined as following:

$$\mathrm{R} = \begin{Bmatrix} a \coloneqq \min(R) \\ b \coloneqq \max(R) \\ \forall i \in [a,b) \end{Bmatrix} \begin{Vmatrix} |I(a-1) - I(a)| > T \\ |I(b+1) - I(b)| > T \\ |I(i) - I(i+1)| \le T \end{Vmatrix}. \qquad (4)$$

Here R is a set of run pixel positions, T=50 is a run separation threshold and directions are determined according to the following recurrent equations, where $I(i) = I(x,y)$ assumed to be current run pixel and $I(i+1)$ is the next run pixel:

$$\begin{aligned} I_{D_h}(i+1) &= I(x+1, y), \\ I_{D_v}(i+1) &= I(x, y+1), \\ I_{D_{d+}}(i+1) &= I(x+1, y+1), \\ I_{D_{d-}}(i+1) &= I(x-1, y-1). \end{aligned} \qquad (5)$$

Once the runs are found, we fill run length histograms for each direction. If the run length l_i < 128, histogram bin is determined according the following equation:

$$h_i = \log_2 l_i + \left[c_i / c_{max} \cdot q\right] \cdot h_{max}, \qquad (6)$$

and according (7), otherwise:

$$h_i = \log_2 l_i + \left[c_{max} \cdot q\right] \cdot h_{max}, \qquad (7)$$

where c_i is an average run brightness level, l_i is run length, c_{max} = 255 is maximal brightness possible, q = 4 is a number of quantization levels, and h_{max} = 6 is a number of histogram bins. Obtained histograms are concatenated into a single feature vector comprised of 96 features and normalized in [0,1] interval.

Classification is performed with SVM with $\chi 2$ kernel:

$$K(x,y) = -\gamma \sum_{i=1}^{N} \frac{(x_i - y_i)^2}{x_i + y_i}, \qquad (8)$$

where γ is set to 0.005.

3.3. Document layout descriptors

We encode document layout using three different descriptors: GRLH, SLBP and BMMFV. GRLH extraction is described in the previous section. One can note that we use it twice: for text extraction and for document classification. That is, we can find run lengths only once and fill both document descriptor and text region descriptor histograms simultaneously.

GRLH document descriptor is computed slightly different in comparison to GRLH text descriptor. Similar to [16], we downscale input document image to 5×10^5 pixels and divide it into plurality of sub-images using a spatial pyramid containing 21 images in total (fig. 4), which gives the best performance according to our experiments. For each sub-image we extract grayscale runlegnth histogram with 9 bins and 4 quantization levels. Hence, the total number of GRLH features increases up to 3024.

One can see that quantization level is the same for document and text descriptors. We utilize it to compute both descriptors simultaneously. An example is given in algorithm 1.

Algorithm 1. GRLH run extraction example for the single line

Input:
L ← Current image line;
D ← Chosen direction;
$\boldsymbol{S_j} \in S = \{S_1, \dots, S_M\}$ ← Sub-image regions;
$\boldsymbol{Q_k} \in Q = \{Q_1, \dots, Q_N\}$ ← Text candidate regions;

Output:
$\boldsymbol{H_{S_j}}, \boldsymbol{H_{Q_k}}$ ← Histograms for each S_j and Q_k, respectively;

while not EndOfLine(L) **do**:
  R_i := GetNextRun(L, D);
  if $\exists S_j: R_i \cap S_j \neq \emptyset$ **then**
    $l_i \coloneqq Length(R_i)$;
    if $l_i < 128$ **then**
      $h_i \coloneqq \log_2 l_i + \left[c_i / c_{max} \cdot q\right] \cdot h_{max}$,
    else
      $h_i \coloneqq \log_2 l_i + \left[c_{max} \cdot q\right] \cdot h_{max}$,
    end if
  end for
  AddToHistogram(H_{S_j}, h_i);
  if $\exists Q_k: R_i - Q_k = \emptyset$ **then**
    $l_i \coloneqq Length(R_i \cap Q_k)$;
    …
    AddToHistogram(H_{Q_k}, h_i);
  end if
end while

SLBP is our custom descriptor that is used in classifier ensembling. Typically, Local Binary Patterns (LBPs) are utilized for encoding local pixel information. There are several works considering document as a spatial pyramid that try to encode grayscale level distribution, but results shown in these works are relatively poor comparing to more advanced methods. In this paper, we propose several extensions to classic LBP that improve performance dramatically. Nevertheless, our experiments show that SLBP performance is slightly wore than GRLH. That is, we choose GRLH as the main descriptor as it can be utilized for both document and text classification and use SLBP to improve ensembling only.

Fig. 4. An example of recursive document subdivision

SLBP extraction is performed in the following manner: first, we convert input image to grayscale and recursively divide it into plurality sub-images, similar to GRLH. Obtained sub-images are downscaled to 100x100 pixels and each pixel is turned into a local binary pattern, according to equation 9:

$$LBP(g_0) = \sum_{i=1}^{8} s(g_0, g_i) \cdot 2^{i-1}, \quad (9)$$

where g_i is an intensity of i-th neighbor pixel and $s(g_0, g_i)$ is a function 10, which compares pixel intensities and returns binary code:

$$s(g_0, g_i) = \begin{cases} 1, g_0 \geq g_i, \\ 0, g_0 < g_i. \end{cases} \quad (10)$$

We convert binary pattern into a numeric binary sequence by extraction of pattern elements starting from the top-left pixel in the counter clockwise direction with 3-pixel margin between central pixel and its neighbors. Extracted 1's and 0's form 8-bit pattern, which is interpreted as unsigned byte. Resulting bytes are combined into 8-bin histogram and normalized in [0; 1] range. Final descriptor has the following length: 21 sub-images x 8 bin histogram = 168.

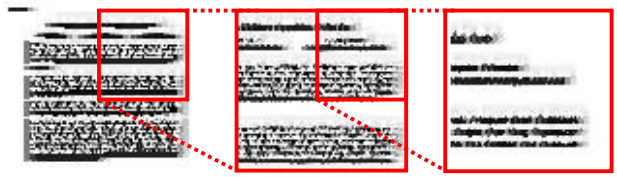

Fig. 5. Image scaling by SLBP descriptor

Each sub-image has size 100x100 pixels, thus histogram values can be converted from integer to real numbers in [0;1] interval by dividing them by 10^4. Image scaling is a critical part of our algorithm (see fig. 5). According to our experiments, 100x100 size leads to the best performance. Another advantage of SLBP is an implicit histogram normalization: it was shown in [16] that choosing particular normalization strategy can significantly affect classifier performance. Scaling each image to the same size resolves this ambiguity.

The last descriptor we use is well-known Fisher Vectors (FV) [24, 25] that are proven to be great choice for document image classification. Typically, FV rely on Gaussian Mixture Model (GMM) which is applied for clusterization of SIFT descriptors. The major disadvantage of SIFT is high calculation complexity. Our classification framework is focused on processing time reduction; hence, we use BRISK descriptors instead. The former are binary descriptors, which means that they extract features that are distributed according to Bernoulli distribution, and not Gaussian. That is, we use Bernoulli Mixture Model (BMM) instead of GMM.

FV with BMM (BMMFV) are obtained as following: first, local BRISK descriptors $X = \{x_1, \dots, x_t, \dots, x_T\}$ are extracted from input image and projected onto T/2 dimensional space using PCA (Principal Component Analysis). Then BMM $\lambda = \{w_i, \mu_{id}, i = 1..N, d = 1..D\}$ is trained, where N is a number of components in BMM and D is a number of bits in each descriptor. Using this model, Fisher scores for each local descriptor are computed:

$$G^X_{\mu_{id}} = \frac{1}{T} \sum_{t=1}^{T} \gamma_t(i) \frac{(-1)^{1-x_{td}}}{\mu_{id}^{x_{td}} (1-\mu_{id})^{1-x_{td}}} \quad (11)$$

Here T is a number of binary features, extracted from an image, and $\gamma_t(i) = p(i|x_s, \lambda)$. Fisher matrix is obtained as following:

$$F_{\mu_{id}} = T w_i \left(\frac{\sum_{j=1}^{N} w_j \mu_{jd}}{\mu_{id}^2} + \frac{\sum_{j=1}^{N} w_j (1-\mu_{jd})}{(1-\mu_{id})^2} \right). \quad (12)$$

Finally, Fisher vector is G_λ^X is obtained by concatenation of normalized Fisher scores $F_{\mu_{id}}^{-\frac{1}{2}} G_{\mu_{id}}^X (i = 1..N, d = 1..D)$. The Fisher vector is further normalized with power normalization and L_2 normalization. Given a Fisher vector $z = G_\lambda^X$, the power-normalized vector *f(z)* is calculated as $f(z) = sign(z)|z|^\alpha$, where $\alpha = 0.5$. We extract Fisher vectors for 21 sub-images, similar to SLBP and GRLH.

3.4. Classification framework

Proposed classification framework is shown in figure 6. We introduce two-step processing: on the first step we classify document image with GRLH based classifier and estimate prediction likehood. If the likehood is high enough, we take the prediction as the result. Otherwise, we classify the document with SLBP and BMMFV based classifiers and use meta-classifier to ensemble their predictions. We estimate a likehood of the final prediction and mark document as "unclassified" if it is too low.

Let us describe the procedure more formally. If $\mathrm{I} \in [0;1]^2$ is a document image then $\mathrm{C} = \{\mathrm{C}_1, \dots, \mathrm{C}_\mathrm{N}\}$ is a set of classifiers, such as:

$$C_n = \{m_n, f_n\}: I \to \{p_n(i|I)\}, \tag{13}$$

where $i \in \mathbb{Z}$ is *i*-th class, $m_n: I \to X$ is descriptor extractor, $X = \{x_1, \dots, x_D\} \in [0;1]^D$ extracted descriptor, and $f_n: X \to \{p_n(i|I)\}$ is classification algorithm.

We choose classifier C_τ according to the following criteria:

$$C_\tau \in C: \tau = \arg\min_T T[C_n(I)], \tag{14}$$

where $T[\cdot]$ is an average image classification time.

Classifier C_τ is used to find probabilities of each class *i* for the image *I*:

$$C_\tau(I) \to \{p_\tau(i|I)\}. \tag{15}$$

Assuming that the class with the highest probability is the predicted class, we estimate the prediction likehood using Bayes formula:

$$p(M_\tau|\Delta P_\tau) = \frac{p(\Delta P_\tau|M_\tau)\, p(M_\tau)}{p(\Delta P_\tau)}, \tag{16}$$

where $M_\tau \coloneqq \arg\max_i p_\tau(i|I)$ and ΔP_τ is obtained as follows:

$$\Delta \mathrm{P}_\tau = \frac{\max_\mathrm{i} \mathrm{p}_\tau(i|I)}{\sum_{\mathrm{i}=\{1..\mathrm{K}\}\setminus \mathrm{M}_\tau} \mathrm{p}_\tau(i|I)}. \tag{17}$$

$\mathrm{p}(\Delta \mathrm{P}_\tau|\mathrm{M}_\tau), \mathrm{p}(\mathrm{M}_\tau)$ and $\mathrm{p}(\Delta \mathrm{P}_\tau)$ can be found from training set by estimating corresponding likehood functions with EM algorithm.

If the likehood $p(M_\tau|\Delta P_\tau) > P_{\tau\,min}$, where $P_{\tau\,min} = 0.9$ we consider C_τ prediction as true and bind *I* to the class with the highest probability M_τ. Otherwise, we classify the image with the all the rest classifiers and merge the predicted probabilities $\{p_n(i|I)\}$ into a single feature vector:

$$P_V = [p_1(1|I), p_1(2|I), \dots, p_n(1|I), p_n(2|I), \dots, p_N(T|I)], \tag{18}$$

where $i = 1..T$ are all possible image classes.

Obtained vector P_V is used as an input for the meta-classifier C_0 that makes the final decision:

$$C_0: P_V \to \{p_0(i|I)\}, \tag{19}$$

If $M_0 \coloneqq \arg\max_i p_0(i|I)$ and ΔP_0 is a ratio of maximal meta-classifier probability to the other probabilities calculated in the same manner as (17) then we can determine the image class as:

$$c(I) = \begin{cases} \arg\max_i p_\tau(i|I), if\ p(M_\tau|\Delta P_\tau) > P_{\tau\,min}, \\ \arg\max_i p_0(i|I), if\ p(M_0|\Delta P_0) > P_{0\,min}, \\ unclassified, \quad otherwise \end{cases} \tag{20}$$

Note, that all classifiers are the pairs $C_n = \{m_n, f_n\}$ of descriptor extractor m_n and classification algorithm f_n. In this paper we use GRLH, SLBP and BMMFV as extractors and SVM as classification algorithm. GRLH and SLBP SVMs use intersection kernel:

$$K(x,y) = \sum_{i=1}^{N} min(x_i, y_i), \tag{21}$$

where $x = \{x_1, \dots, x_D\}$, $y = \{y_1, \dots, y_D\}$ и $x, y \in [0;1]^D$ are descriptors, and BMMFV is classified with SVM with linear kernel. Meta classifier is an SVM also. It uses the same χ^2 kernel (8) as text classifier, but with different parameter $\gamma = 0.05$.

The training is performed in the following manner: first, we train $C_1 \dots C_N$ classifiers using Platt's framework [26]. Then we train meta-classifier C_0 on the same data using probabilities $\{p_1(i|I) \dots p_N(i|I)\}$ obtained with classifiers $C_1 \dots C_N$. Finally, we estimate $p(\Delta P_\tau|M_\tau), p(M_\tau), p(\Delta P_\tau)$ and $p(\Delta P_0|M_0), p(M_0), p(\Delta P_0)$ with EM algorithm.

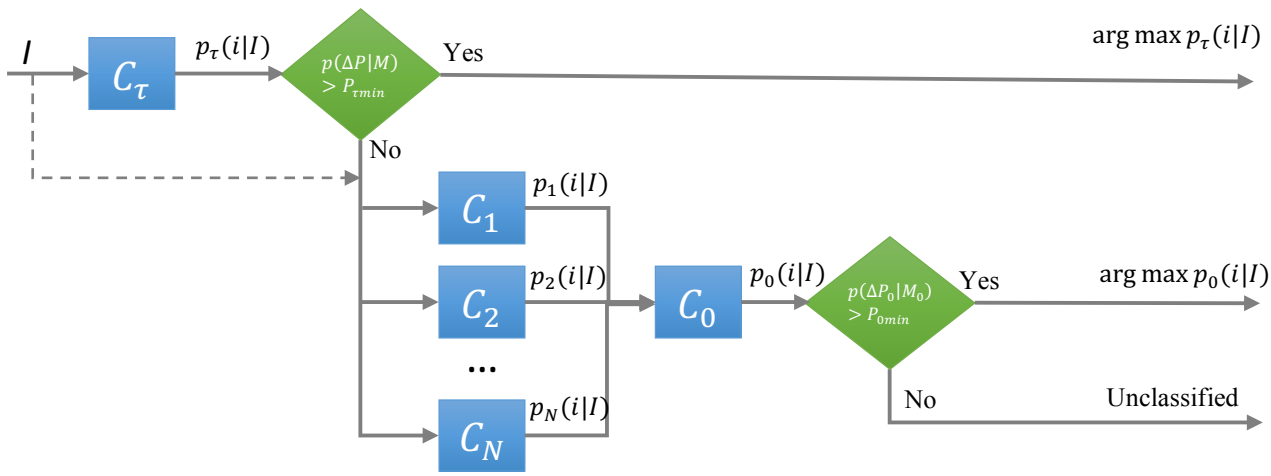

Fig. 6. Proposed classification framework

4. Results and Discussion

4.1. Document datasets

There are several well-known document datasets that are utilized for document processing, but only few of them are suitable for classification. That is, we were forced to generate new datasets by ourselves. Our approach to document generation is template-based: first, we specify several different templates and then automatically fill them with random "lorem ipsum" text. If the number of words in each template is fixed, it is considered as fixed template and if the number of words can vary, it is flexible.

Fixed templates correspond to fill-in forms, while flexible are similar to typical business documents, such as letters, bills, invoices and so forth. Documents with fixed templates are represented by NIST dataset [27]. Documents with flexible templates can be found in MARG dataset [28]. While classification of NIST is rather simple, MARG is more challenging due to high inter-class variability. In addition to these datasets, we generated four new datasets by ourselves. To make task more challenging we gathered two large in-house datasets also (see table 1 and figure 7). Let us describe all the datasets in detail.

Table 1. Description of evaluation datasets

Dataset	Template	Artificial	Scanned	# of classes	# of test images
NIST	Fixed	+/-	+	20	824
MARG	Flexible	-	+	8	1135
FlexScan	Flexible	+	+	15	780
FlexBack	Flexible	+	+/-	15	780
FlexDist	Flexible	+	+	15	60
FlexRot	Flexible	+	-	15	840
Fixed	Fixed	+/-	-	43	1720
Joint	Flexible	+/-	-	78	3324

NIST [27] is a document collection comprised of NIST Special Database 2 and NIST Special Database 6. It contains 20 types of filled forms with fixed layout. Total number of images is 11185. We took a random subset of 824 images for testing.

MARG [28] consists of 9 classes with complex layouts. This dataset is very challenging. In our evaluation we use 1135 testing images from 8 classes out of 9. We removed class "othertype" because it has no particular layout.

FlexScan is an automatically generated dataset comprised of 15 classes. It contents documents with flexible layouts. We physically printed and scanned all the images from this dataset with 300 DPI resolution in order to simulate real documents. 780 images were taken for testing.

FlexBack is the same dataset as FlexScan with random backgrounds added. We use it to evaluate performance of proposed classification pipeline in the case of highly distorted documents.

FlexDist contains training images from FlexScan dataset and 60 testing images that are heavily distorted with drawings, handwritten notes, coffee spots and so forth. In theory, each classification algorithm trained on FlexScan should be able to perfectly classify this dataset.

FlexRot is comprised of images from FlexScan and FlexDist datasets rotated by $\alpha = \alpha_1 + \alpha_2$, where $\alpha_1 \in \{0°, 90°, 180°, 270°\}$ and $\alpha_2 \in [-3°, +3°]$ are random variables. We also change document image brightness and contrast by applying the following rule:

$$I(x) = \alpha I(x) + \beta, \qquad (22)$$

where $I(x) \in [0; 255]$ are the pixels of the initial image and $\alpha \in [0.85; 1.25], \beta \in [-5; 5]$ are random variables.

Finally, **Fixed** is an in-house dataset comprised of 1720 training images of filled forms assembled into 43 classes.

Joint is an extreme case, comprised from NIST, FlexScan and Fixed, which is used for performance evaluation on large datasets. It includes 78 classes and 3324 testing images in total.

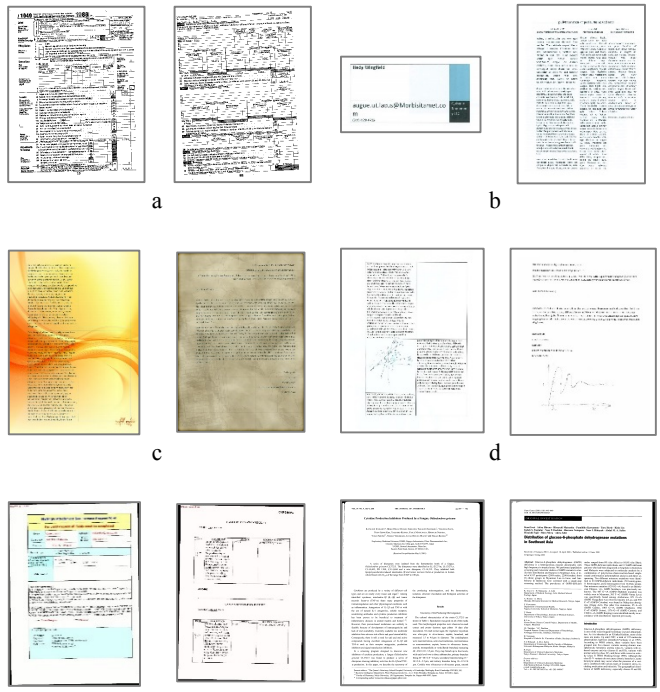

Fig. 7. Evaluation datasets examples: a) NIST, b) FlexScan, c) FlexBack, d) FlexDist, e) Fixed, f) MARG

4.2. Experimental results

In this section, we will test proposed document image classification pipeline using aforementioned datasets. One can note, that we use text extraction algorithm as a pre-processing step. Therefore, let us compare our MSER-SI+GRLH algorithm to existing MSER+RLH algorithm first.

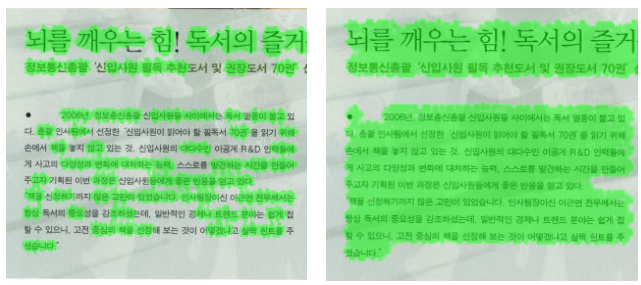

Fig. 8. Text candidates (green areas) detected by: a) MSER and b) MSER-SI

Comparison was performed using document collection [29] extended with a large number of magazine articles. We manually prepared groundtruth data for each image. All the images were downscaled to 6×10^5 pixels. Evaluation was performed using two

criteria. First, we measured the total number of found text candidates. Similar to typical segmentation task, we considered region as found if it has more than 70% area overlapping with groundtruth text region. On the second stage, we measured the number of text regions found by the algorithms.

Evaluation was performed by the means of precision, recall and F1 metrics. Table 2 contains comparison of the total number of found text candidate regions. One can see that MSER finds significantly less regions than our MSER-SI. Table 3 contains comparison of the number of text regions found by the algorithms. In the second case we used MSER-SI as detector; hence this comparison demonstrates the difference between RLH and our GRLH descriptor. Easy to see that RLH is slightly better in the terms of precision, but GRLH demonstrates much better recall.

Table 2. Comparison of the amount of found text regions, %

Method	Precision	Recall	F1
MSER	82.7	91.7	85.0
MSER-SI	**96.3**	**94.0**	**95.1**

Practically, it means that GRLH is able to find more text regions than RLH with slightly higher error rate. In addition, MSER-SI finds significantly larger number of text candidates than MSER, which is the major goal for text extraction in the case of document classification by the means of text layout. An example of text candidates found with MSER and MSER-SI is given in figure 8.

Table 3. Comparison of the amount of real text regions, %

Method	Precision	Recall	F1
MSER-SI + RLH	**81.3**	80.6	80.1
MSER-SI + GRLH	80.4	**85.0**	**83.7**

Let us now compare proposed SLBP descriptor with existing LBP. We performed comparison using FlexRot dataset. The results are given for the maximal accuracy; hence, the number of training images for each descriptor was different. The classifier for both descriptors is SVM with the same settings. Table 4 contains the results. As it was expected, naive LBP demonstrates poor accuracy. One can see that SLBP without scaling sub-images to 100x100 pixels demonstrates poor results too. Therefore, both spatial pyramid utilization and scaling are essential for achieving good results with our descriptor.

Table 4. Classification accuracy of SLBP and LBP

Descriptor	Accuracy, %
LBP	62.9
SLBP w/o scaling	76.4
SLBP	**89.4**

Finally, we compared our classification pipeline with several existing algorithms: RLH with SVM [16] and Fisher vectors based on Gaussian Mixture Model with SVM (GMMFV) [20]. We also added majority voting algorithm (MV) to comparison in order to prove that our approach to classifier ensembling gives better results than trivial voting. MV uses the same GRLH, SLBP and BMMFV based classifiers as in the proposed pipeline.

Our approach is demonstrated in two modifications: the first one, CE (Classifier Ensembling), always uses all three classifiers, which means that classification with GRLH alone is omitted. The second modification, CE+, uses exactly the same pipeline that is shown in figure 6. We include two modifications in comparison in order to show that fast two-stage algorithm CE+ demonstrates similar results to the robust, but slow CE.

In the most papers, algorithms are trained using a large number of training images, but real-life datasets are relatively small as it is hard to obtain a large number of documents. Therefore, we tried to determine an optimal number of training images first. We took FlexRot dataset and compared algorithms accuracy using different numbers of training images per class. The results are shown in figure 9.

One can see that RLH demonstrates the worst results. GMMFV has low accuracy if the number of training images is small, but its results significantly improve when it is trained using more than 30 images per class. MV demonstrates better results than individual classifiers, but proposed CE has the best accuracy for all possible training image numbers, except the largest ones that lead to overfitting

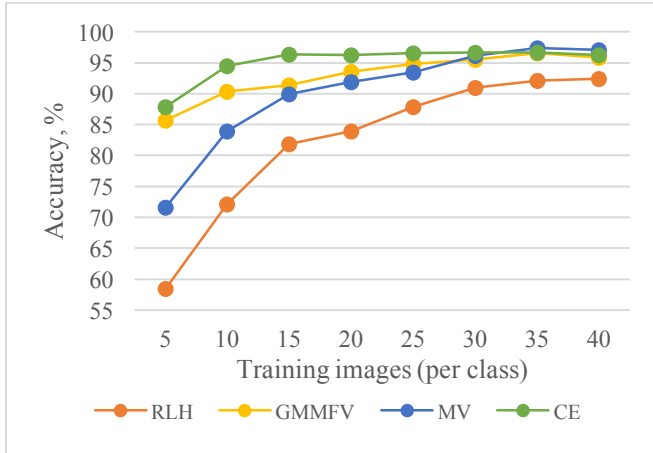

Fig. 9. Classification accuracy depending on the number of training images per class

According to obtained results, 10 images per class is an optimal number of training images. Lower numbers lead to significant accuracy decrease, while higher numbers of documents are hard to collect. Nevertheless, we test the algorithms using both 10 training images per class and "unlimited" number of the images. The algorithms were trained with different numbers of images until overfitting. The best result was taken as a maximal accuracy.

The results for 10 training images are shown in table 5. The maximal accuracy could be found in table 6. In the most cases our algorithms CE and CE+ demonstrate the best or at least the second result.

Table 5. Classification accuracy for 10 images per class, %

Dataset	Classifier				
	RLH	GMMFV	MV	CE	CE+
NIST	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
FlexScan	97.6	98.7	97.6	**99.5**	**99.5**
FlexDist	95.0	91.7	96.7	**98.3**	**98.3**
FlexRot	82.1	90.4	84.8	**96.4**	94.5
Fixed	99.4	**100.0**	99.8	99.9	99.9
Joint	99.0	99.6	99.0	**99.7**	**99.7**
MARG	56.7	50.4	56.5	54.8	**57.1**

Table 6. Maximal classification accuracy, %

Dataset	Classifier				
	RLH	GMMFV	MV	CE	CE+
NIST	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
FlexScan	99.5	**99.7**	99.6	**99.7**	99.6
FlexDist	96.7	93.3	**100.0**	98.3	98.3
FlexRot	92.4	96.6	**97.4**	96.7	96.3
Fixed	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Joint	**99.9**	**99.9**	**99.9**	**99.9**	**99.9**
MARG	68.9	54.6	68.7	70.3	**70.7**

Let us now compare processing time. We used FlexScan dataset for comparison and 10 training images per class. The results are shown in figure 10. Obviously, the fastest classifier is RLH-based as the RLH is the simplest algorithm and CE classifier is the slowest. Nevertheless, CE+ proposed in this paper demonstrates much better results than GMMFV and MV. That is, our classifier CE+ has significant performance improvement in comparison to complex classification algorithms.

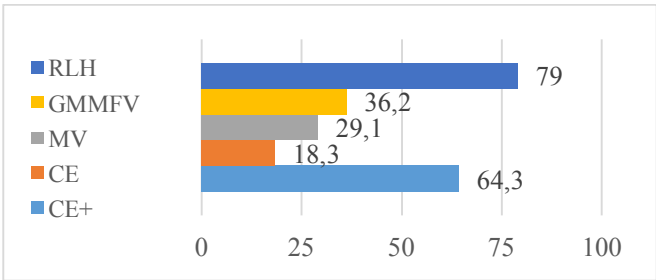

Fig. 10. Classification performance, img/sec

Finally, we compared all algorithms using FlexBack dataset with our text extraction (TE) algorithm enabled and without it. The results are shown in table 7. Obviously, text extraction significantly improves classification accuracy of all algorithms. Regardless the TE presence, our algorithms demonstrate significantly better results in comparison to the others. Note, that GMMFV result is poor if TE is disabled. The reason is that the local descriptors encode to much noise as they cannot separate background from the text.

Table 7. Classification accuracy for documents with complex backgrounds

Dataset	Classifier				
	RLH	GMMFV	MV	CE	CE+
FlexBack	81.7	67.2	80.5	**90.8**	**90.8**
FlexBack+TE	86.4	89.7	90.6	**96.8**	**96.8**

5. Conclusion

In this paper, we proposed a novel document image classification framework based on layout information. Our framework provides the best accuracy for the most of real-file classification scenarios. According to our experiments, proposed framework is able to classify up to 63 documents per second, while the best high-accuracy algorithms are capable to handle only 36 images in the same time. Developed descriptors demonstrate high robustness to background presence and proposed text extraction algorithm significantly improves classification accuracy.

References

[1] N. Chen, D. Blostein. A survey of document image classification: problem statement, classifier architecture and performance evaluation. Int. Journal of Doc. Anal. and Recogn., 10(1), 1-16 (2007)

[2] D. Pintsov, Method and system for commercial document image classification, U.S. Patent No. 8,831,361 (2014)

[3] Mao, Song, Azriel Rosenfeld, and Tapas Kanungo. "Document structure analysis algorithms: a literature survey." Electronic Imaging 2003. International Society for Optics and Photonics, 2003.

[4] Cesarini, Francesca, et al. "Encoding of modified XY trees for document classification." Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on. IEEE, 2001.

[5] Cesarini F. et al. Structured document segmentation and representation by the modified XY tree //Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on. – IEEE, 1999. – C. 563-566.

[6] Soda, Stefano Baldi Simone Marinai Giovanni. "Using tree-grammars for training set expansion in page classification." (2003).

[7] Marinai, Simone, et al. "A general system for the retrieval of document images from digital libraries." Document Image Analysis for Libraries, 2004. Proceedings. First International Workshop on. IEEE, 2004.

[8] Marinai, Simone, Emanuele Marino, and Giovanni Soda. "Tree clustering for layout-based document image retrieval." Second International Conference on Document Image Analysis for Libraries (DIAL'06). IEEE, 2006.

[9] Marinai, Simone, Marco Gori, and Giovanni Soda. "Artificial neural networks for document analysis and recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 27.1 (2005): 23-35.

[10] Nattee C., Numao M. Geometric method for document understanding and classification using online machine learning //Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on. – IEEE, 2001. – C. 602-606.

[11] Shin, Christian, David Doermann, and Azriel Rosenfeld. "Classification of document pages using structure-based features." International Journal on Document Analysis and Recognition 3.4 (2001): 232-247.

[12] Byun, Yungcheol, and Yillbyung Lee. "Form classification using DP matching." Proceedings of the 2000 ACM symposium on Applied computing-Volume 1. ACM, 2000.

[13] Shimotsuji, Shigeyoshi, and Mieko Asano. "Form identification based on cell structure." Pattern Recognition, 1996., Proceedings of the 13th International Conference on. Vol. 3. IEEE, 1996.

[14] Ting, Antoine, and Maylor KH Leung. "Business form classification using strings." Pattern Recognition, 1996., Proceedings of the 13th International Conference on. Vol. 2. IEEE, 1996.

[15] Usilin, Sergey, et al. "Visual appearance based document image classification." 2010 IEEE International Conference on Image Processing. IEEE, 2010.

[16] Gordo, Albert, Florent Perronnin, and Ernest Valveny. "Large-scale document image retrieval and classification with runlength histograms and binary embeddings." Pattern Recognition 46.7 (2013): 1898-1905.

[17] Gao, Hongxing, et al. "Key-Region Detection for Document Images--Application to Administrative Document Retrieval." 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013.

[18] Chen, Siyuan, et al. "Structured document classification by matching local salient features." Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012.

[19] Kumar, Jayant, Peng Ye, and David Doermann. "Structural similarity for document image classification and retrieval." Pattern Recognition Letters 43 (2014): 119-126.

[20] Gordo, Albert, Florent Perronnin, and Francois Ragnet. "Unstructured document classification." U.S. Patent Application No. 12/632,135.

[21] F. Perronnin, D. Larlus, Fisher vectors meet neural networks: A hybrid classification architecture, in: CVPR, 2015.

[22] Nistér, David, and Henrik Stewénius. "Linear time maximally stable extremal regions." European Conference on Computer Vision. Springer Berlin Heidelberg, 2008.

[23] Yin, Xu-Cheng, et al. "Robust text detection in natural scene images." IEEE transactions on pattern analysis and machine intelligence 36.5 (2014): 970-983.

[24] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: CVPR, 2007.

[25] F. Perronnin, J. Sa´nchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: ECCV, 2010.

[26] Platt, John. "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods." Advances in large margin classifiers 10.3 (1999): 61-74.

[27] D. Dimmick, et.al, Structured forms database, Tech. Report Spec. Database 2. SFRS, National Institute of Standards and Technology (2001)

[28] Thoma, G. F. G. "Ground truth data for document image analysis." Symposium on document image understanding and technology (SDIUT). 2003.

[29] Antonacopoulos, Apostolos, et al. "ICDAR 2013 competition on historical book recognition (HBR 2013)." 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013.

Author Biography

Sergey *S.* Zavalishin *received his MS degree in Computer Science from Moscow Engineering Physics Institute/University (MEPhI), Russia in 2012. Currently he is a post graduate student of Ryazan State Radio Electronics University. His research interests include machine learning, computer vision and image processing.*

Andrey V. Bout *received his MS degree in Applied Mathematics and Computer Science from Southern Federal University (SFedU), Rostov-on-Don, Russia in 2012. He has finished his postgraduate course from SFedU in 2015 (doctoral work not finished yet). His research interests focus in machine learning, deep learning ,computer vision and high performance computing.*

Ilya V. Kurilin *received his MS degree in radio engineering from Novosibirsk State Technical University (NSTU), Russia in 1999 and his PhD degree in theoretical bases of informatics from NSTU in 2006. Since 2007, Dr. I. Kurilin has join Image Processing Group, Samsung RnD Institute Russia where he is engaged in photo and document image processing projects.*

Michael N. Rychagov *received his MS degree in physics from Department of Physics at Lomonosov Moscow State University, Russia in 1986, PhD degree and Dr. Sc. degree from the same university in 1989 and 2000 correspondingly. Since 1989 he has been involved in R&D and teaching on Faculty of Electronic and Computer Engineering at Moscow Institute of Electronic Technology (Technical University). Since 2004, he joined Samsung R&D Institute Rus, Moscow, Russia, where he is currently Director of Algorithm Laboratory. He is member of IS&T and IEEE Societies.*