

## Research Article

# Stamps Detection and Classification Using Simple Features Ensemble

**Paweł Forczmański and Andrzej Markiewicz**

*Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin, Żołnierska Street 52, 71-210 Szczecin, Poland*

Correspondence should be addressed to Paweł Forczmański; [pforczmanski@wi.zut.edu.pl](mailto:pforczmanski@wi.zut.edu.pl)

Received 1 August 2014; Accepted 23 September 2014

Academic Editor: Erik Cuevas

Copyright © 2015 P. Forczmański and A. Markiewicz. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The paper addresses a problem of detection and classification of rubber stamp instances in scanned documents. A variety of methods from the field of image processing, pattern recognition, and some heuristic are utilized. Presented method works on typical stamps of different colors and shapes. For color images, color space transformation is applied in order to find potential color stamps. Monochrome stamps are detected through shape specific algorithms. Following feature extraction stage, identified candidates are subjected to classification task using a set of shape descriptors. Selected elementary properties form an ensemble of features which is rotation, scale, and translation invariant; hence this approach is document size and orientation independent. We perform two-tier classification in order to discriminate between stamps and no-stamps and then classify stamps in terms of their shape. The experiments carried out on a considerable set of real documents gathered from the Internet showed high potential of the proposed method.

## 1. Introduction

Nowadays, computer analysis of digitized documents is one of key areas of digital image processing and pattern recognition. Computer-based processing of documents joins many fields of research, especially computer vision, image analysis, pattern recognition, and artificial intelligence. The detection and recognition of rubber stamps (seals) on digital images are still very important problems. As we face a significant change in the technology, namely, a conversion from paper documents into digital ones, the need for solutions for automatic segmentation and extraction of important document's elements is very high. The problem dates back to the '80s last century, when an issue of stamp authenticity has been addressed [1]. However, despite considerable progress in this field, this problem still remains open [2–4].

The literature survey shows that algorithms of stamp detection and recognition employ different features of digital images. Most of the methods can be classified as ones using shape information [4, 5] or ones based on color features [3, 6]. There are only a few algorithms that join features from those domains (e.g., [7]).

While stamp detection can be realized by means of both shape and color features, the problem of stamp classification is purely shape-based. It has been proved many times that shape (silhouette, contour) is one of the most important low level image features since it is a very important attribute to human perception [8]. Humans tend to perceive complex scenes as being composed of elementary objects, which can be identified by their shapes. Shape is a particularly useful, especially when we want to identify specific objects in image, for example, in medicine or biology (e.g., cell shapes in microscopic images) [9, 10] or surveillance (e.g., optical character recognition of license plates [11]) and many other systems, where objects easy to segment may be determined. While general shape recognition and classification problems have a significant place in the scientific literature, often with reported accuracies of over 90% (e.g., [12, 13]), the specific topic of stamps retrieval by means of low-dimensional feature sets is much less represented (with significantly lower accuracies).

A recent study shows that few different methods oriented at automatic stamp detection and classification have been proposed. They work either on color stamps [3] or detect

objects of particular shape [4, 14]. Unfortunately, there is no general approach aimed at detection and classification of all the diversity of stamps. There is also a large group of algorithms oriented at logo retrieval, which is a similar problem [15, 16]. The newest ones include application of modern shape descriptors, for example, SIFT/SURF/FAST [17–19] or ART [20]. It should be, however, noted that such approaches aimed at logo detection can not be directly employed to stamp detection because of different properties of both objects' classes. Such geometrical features that are common for both object classes [15, 21, 22] are heavily influenced by the process of imprinting (noise, inconsistency, gaps, stains, etc.) in the stamp representation. Because of this fact and the diversity of stamps, even within a single class (in terms of shape), a straightforward application of such descriptors is not possible.

In this paper we present a new solution for detecting stamps of different shapes and colors. The experiments showed that they can be extracted properly even if they are overlapped with signature or text. Because of the high variance in the shape domain we focus on rather standardized stamps (with well-defined shape) of any particular color. In opposition to [3], we do not rely on color properties only; hence the detection of black stamps is also possible. Moreover, shapes being detected are not limited to ovals and squares, like in [4]. The experiments showed also some good results of detecting stamps in documents containing other similar objects such as logos and texts.

*1.1. Existing Approaches.* As it has been noted, the algorithms of stamp detection and recognition can employ different features of digital images. These features are based, in most cases, on color or edge information. The algorithms proposed in [3, 6] use  $YC_bC_r$  color representation. Candidate areas are extracted using the projection of  $C_b$  and  $C_r$  matrices onto both axes. As an effect, a binary mask of bounding boxes (instead of the original areas) is generated. Unfortunately, it negatively influences further processing aimed at determining their shapes. Although, the reported detection accuracy of color stamps is rather high (over 83%), such a solution can not be employed in case of grayscale images or black-colored stamps.

The method proposed in [23] allows detecting stamps even in highly degraded documents. It does not take into consideration the color information, since the main principle of the proposed method is so called connected edges that creates contour of the stamp. In the first step, an input image is subjected to scaling and edge detection. Then, candidate regions are evaluated using a set of rules that are characteristic to pixels within stamps. Selected elements are finally verified using Hough Transform. Unfortunately, the detection algorithm can deal with circular- and oval-shaped stamps only. The reported accuracy is higher than 35% and it is not sufficient for practical implementations.

Another method that employs edge information is presented in [5]. It uses a two-stage approach. The first step is based on letters detection. The letters being often a part of a stamp have specific spatial properties, which are employed

in presented method. The second stage involves the usage of a Generalized Hough Transform in order to verify the result obtained in the first step. GHT has a clear advantage over the basic Hough Transform version since it allows finding any particular shapes. The experiment performed by the authors showed that the accuracy is close to 70%.

The solution proposed in [15] addresses the problem of logo detection, which can be considered similar to the tasks of stamp detection and recognition. Proposed approach is divided into three stages: segmentation, in which a document is decomposed into smaller parts (text regions, images, logos, etc.), detection of logotypes based on the features extracted from output segments, and matching logotypes by means of template matching. The main principles of the proposed method are projection of binarized image, analysis of connected components, and a classification stage. Unfortunately, this solution can not be directly used for the detection of stamps. Features used by the authors of [15] do not allow getting such good accuracy of classification in case of stamps, which differ significantly from logotypes.

The whole group of methods is based on recently proposed features, like Scale-Invariant Feature Transform (SIFT) or Speeded Up Robust Features (SURF). In the paper [17] a fast and efficient logo detector is proposed. The solution involves the use of feature vectors obtained by SURF. It is designed to detect a set of local features for each key point in the image. The final features are reduced using Principal Component Analysis. The reference database contains a large collection of samples. Despite a quite high complexity, the accuracy of the proposed algorithm is equal to 67%.

The algorithm proposed in [20] is also a multitier approach. Firstly, it performs a noise reduction (median filtering) on the input image. Then the image is binarized and subjected to horizontal dilation. Extracted candidate regions are described using a set of geometrical features. The final verification is done using a tree. The next step is verification. It is done using a set of rules. After successful detection, logos are classified using  $k$ -Nearest Neighbor classifier. Reported accuracy is equal to 92%; however the method works only on logotypes that are not overlapped by text or any other objects.

It is worth noticing that in the method presented in this paper we joined several elementary approaches in order to create more flexible and more accurate algorithm.

*1.2. Data Characteristics.* The problem with stamp detection and classification comes mainly from the lack of templates or general models. The in-depth analysis shows that this is due to the absence of standard and commonly used stamp's representation. Stamps are complex objects, containing graphical and textual elements that can be located anywhere within documents. Their diversity comes from varying orientation, color, fonts, ornaments, and quality of imprinting. Even two imprints of the same physical stamp can look very different. The same applies to other objects that were described in [21]. Taking into consideration above limitations, object's shape is considered one of the most stable features in terms of classification. Exemplary stamps extracted from real documents divided into shape-specific classes are provided



FIGURE 1: Sample official stamps, often regular and without decorations (first four rows), and unofficial stamps, more complex, with many decorative motives (the last row).

in Figure 1. The last row of Figure 1 shows stamps that can not be categorized in terms of shape. Such stamps are mostly unofficial and less common.

The color information is very important when it comes to document analysis since original documents in most cases include color stamps. The black stamps are less frequently used for official stamping, and the lack of color in many cases suggests a copy (not original document). It is also worth noticing that the stamps on official documents do not cover large area (often not more than 3%–5% of the total image area for A4 page) [24]. Above observation can ease the stage of stamp detection.

## 2. Algorithm Overview

The input for the proposed approach is given as a scanned or photographed (paper) document. In the scanning process the imaging plane is parallel to the document plane; hence we preserve geometrical features of all elements. In order to reduce the artifacts it is also recommended that the input image has quite high spatial resolution and is stored in a file format that provides minimal loss of quality after decompression. The algorithm presented further is developed to work on both color and monochrome images. It is worth noticing also that the input images are not subjected to any brightness alternation (neither histogram stretching nor equalization). Moreover, we do not assume the number of stamps as well as their location and orientation. The output provides information about the number of stamps detected, their shapes, colors, and coordinates.

The algorithm (see Figure 2) consists of the following steps.

- (1) Load an image.
- (2) Detect candidates (in monochrome images or in each color channel in case of color image):

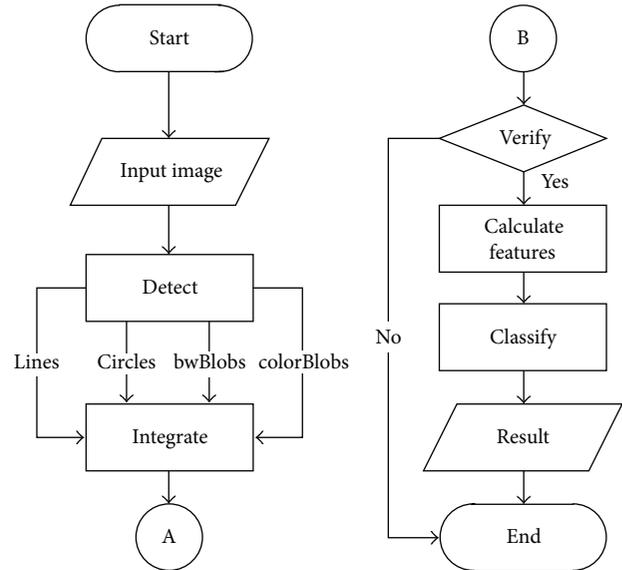


FIGURE 2: Stages of processing.

- (i) detect lines;
- (ii) detect circles;
- (iii) detect other shapes (*bwBlobs* or *colorBlobs*).

- (3) Integrate locations and dimensions.
- (4) Verify stamps/no-stamps.
- (5) Classify and report the results.

Firstly, in case of color images, we perform a color conversion  $RGB \rightarrow YC_bC_r$  and work on  $C_b$  and  $C_r$  planes in an independent manner [6]. Then, in order to detect areas of potential stamps, we look for elementary image structures, like lines, circles, and any other consistent areas. Next, we obtain candidate areas which are later verified using a number of features that characterize specific shapes and classify them into stamps and no-stamps (objects with similar shape, yet with different raster features). The further recognition (in terms of shape) is performed using shape descriptors and dedicated classification methods.

Several experiments on color-based detection have also been described in our previous works [6, 25].

**2.1. Image Preprocessing.** It is known that  $RGB$  color space is not optimal in terms of color image segmentation due to the correlation between channels [6]. Hence we convert an input image into  $YC_bC_r$  representation. Since the color of stamps are mainly in the blue or red range (according to our observations, less than 4% of the stamps are represented by other colors), it is especially usable because of the red/blue separation in that color space. What is more, it is a native color format for JPEG/JFIF files. However, in general case, we assume that an input image of a document is stored in a file with possibly lossless compression, high spatial resolution, and full color range (24-bit  $RGB$ ). Hence, the input image is



FIGURE 3: Sample document containing a stamp (a),  $C_b$  (b) and  $C_r$  (c) components of its  $YC_bC_r$  representation together with the binarized output image (d).

converted into  $YC_bC_r$  color space (ITU-R BT.709 standard) in order to expose the above shown color properties:

$$\begin{aligned} Y &= 16 + (65.481 \cdot R + 128.553 \cdot G + 24.966 \cdot B), \\ C_b &= 128 + (-37.797 \cdot R - 74.203 \cdot G + 112 \cdot B), \\ C_r &= 128 + (112 \cdot R - 93.786 \cdot G - 18.214 \cdot B), \end{aligned} \quad (1)$$

where  $R$ ,  $G$ ,  $B$  and  $Y$ ,  $C_b$ ,  $C_r$  are appropriate color components.

According to the above observations, we examine each channel  $Y$ ,  $C_b$ , and  $C_r$  in an independent manner. (see Figure 3).

As a next preprocessing stage, above matrices are filtered using simple averaging mask in order to reduce noise. The filter uses a mask whose size is equal to 3% of input image's shorter edge. This ensures favorable results in the aspect of computation cost versus quality.

In the next step, each image is binarized and consistent areas are determined. Additionally, before filling holes, each area is subjected to morphological operation, region growing. Binary image is then labeled and each candidate region is passed to the module responsible for stamp detection (verification). Our previous experiments [6] showed also that stamps are objects that are not smaller than 5% of shorter edge of input image and not larger than 40% of its longer edge. Hence, we look for such areas only.

In the case presented below (Figure 3), several potential areas are detected and passed to the verification/classification

stage. Presented binarized image is obtained as a superposition of thresholded  $C_b$  and  $C_r$  channels; hence it contains areas of high blue and red channels intensity.

**2.2. Stamp Detection.** Different stamp shapes are detected by independent modules (see Figure 2). Firstly, the input image with dimensions of  $H_{doc} \times W_{doc}$  pixels is binarized using adaptive thresholding [26], so it contains a number of closed areas.

**2.2.1. Circular Shapes.** Detection of circles is based on Circular Hough Transform (CHT) performed on intensity image. At first, an image is binarized and then edge detection using Canny filtering is performed; then morphological opening in order to eliminate noise is employed. On such image, classical Hough Transform is used. Using a voting strategy, it finds circles of different radii. The details of the algorithm can be found, for example, in [27]. In our experiments we used radii from interval [15, 120]. Exemplary results of circular stamps detection are presented in Figure 4.

**2.2.2. Shapes Consisting of Straight Lines.** For the extraction of shapes containing straight lines (e.g., squares, rectangles, and triangles), we employ a line detector based on Hough Transform. It works on image containing pixel gradients. Firstly, the input image is averaged in order to remove noise. Then, we select pixels of high intensity (above a certain threshold) and for each one we perform the following sequence of operations. Each pixel is represented in the

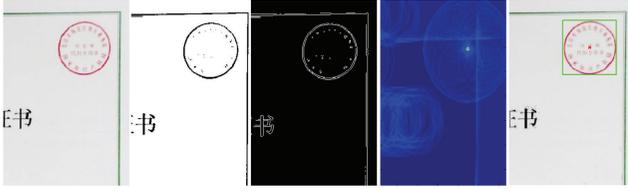


FIGURE 4: Sample document containing circular stamp (left), binarized, after edge detection, accumulator matrix and detection results, respectively.

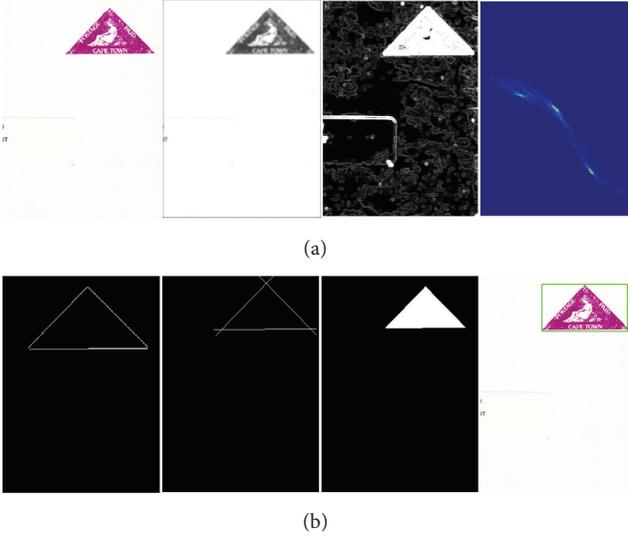


FIGURE 5: Sample document containing triangular stamp, blurred and grayscaled, after gradient detection and accumulator matrix in (a), respectively. Detected straight lines, lines extended to create closed area, filled area and the detection result in (b), respectively.

parametric Hough Space as a sinusoid. After checking all selected pixels we get a bunch of intersecting curves. The values at the intersections give an information about the possible lines in the original image. Since values of both slope and intercept in original Hough approach are unbounded and slope value for vertical lines is huge, application of this original technique is complicated. Hence we use an approach, presented in [28], which is an alternative to the original Hough approach.

Then we extend each detected line using 20 pixels at both of its ends in order to create closed area, which is later filled. Further we eliminate introduced extensions. Exemplary results of stamps containing straight lines in their contours are presented in Figure 5.

**2.2.3. Nonregular Shapes.** In order to extract any other shapes we introduce a heuristic approach presented below. It uses the results of our observations (based on the analysis of our documents database) and very basic algorithms from the image processing area. For each consistent area (blob) found in the image its bounding box is determined. Its area is given as  $A_{\text{box}} = H_{\text{box}} * W_{\text{box}}$ , where  $W_{\text{box}}$  and  $H_{\text{box}}$  are its width

and height, respectively. We assume that the background pixels are zeros; hence, additionally, we calculate the number of pixels (other than zero) in each area  $A_p$ . It gives us an information about complexity of each blob. Candidates that satisfy the following conditions are passed to the further processing:

$$400 < A_p < 5000,$$

$$\min \{H_{\text{box}}, W_{\text{box}}\} > 0.05 * \min \{H_{\text{doc}}, W_{\text{doc}}\}, \quad (2)$$

$$\max \{H_{\text{box}}, W_{\text{box}}\} < 0.40 * \max \{H_{\text{doc}}, W_{\text{doc}}\}.$$

As it can be seen, each object's area must fall into range (400, 5000) pixels. This constraint is not denominated by any relative value and is used just as a filter since other two conditions could easily select objects smaller or larger than expected.

The alternative way of detecting stamps, not taking into consideration their geometrical properties, was presented in [24]. However, that approach rejects longitudinal blobs and promotes objects of high contrast only.

**2.3. Stamp/No-Stamp Verification.** Extracted objects are verified using a set of 11 object's characteristics captured in the spatial domain. We compute seven direct features: average pixel intensity  $f_{\text{avg}}$ , intensity standard deviation  $f_{\text{std}}$ , median value of intensity, pixel contrast, brightness to contrast ratio, intensity of edges, brightness to edges intensity ratio, and four features calculated from Gray-Level Cooccurrence Matrix (GLCM) [29]: contrast, correlation, energy, and homogeneity.

Firstly, image intensity matrix  $I$  of size  $H_{\text{box}} \times W_{\text{box}}$  containing a single blob is converted into a vector  $f$  of length  $l = H_{\text{box}} * W_{\text{box}}$ . Aforementioned features (aside from those which are self-explanatory) are calculated according to following formulas:

$$f_{\text{avg}} = \frac{1}{l} \sum_{i=1}^l f(i), \quad (3)$$

$$f_{\text{std}} = \sqrt{\frac{1}{l} \sum_{i=1}^l (f(i) - f_{\text{avg}})^2}. \quad (4)$$

Median value is calculated over sorted vector  $f$ , so that  $f(i) \leq f(i+1)$ ,  $i = 1 \dots l-1$ . The result is equal either to  $f((l+1)/2)$  or  $(1/2)(f(l/2) + f(l/2+1))$  depending on  $l$  parity. Calculation of remaining features depends on edge detection operator. In our case Sobel algorithm was applied [30]. Intensity of edges is then computed in accordance with (3).

GLCM matrix of size  $P \times Q$  is created by calculating how often a pixel of intensity  $p$  has a pixel of intensity  $q$  in its closest horizontal neighbourhood. However, this could be changed based on  $\Delta x$ ,  $\Delta y$  offset parameter. For a discrete image,  $p = 1, \dots, P$  and  $q = 1, \dots, Q$ . Its size is determined by number of gray-levels; in this case both  $P$  and  $Q$  are equal to 8. It also means that all values in image matrix  $I$  are scaled

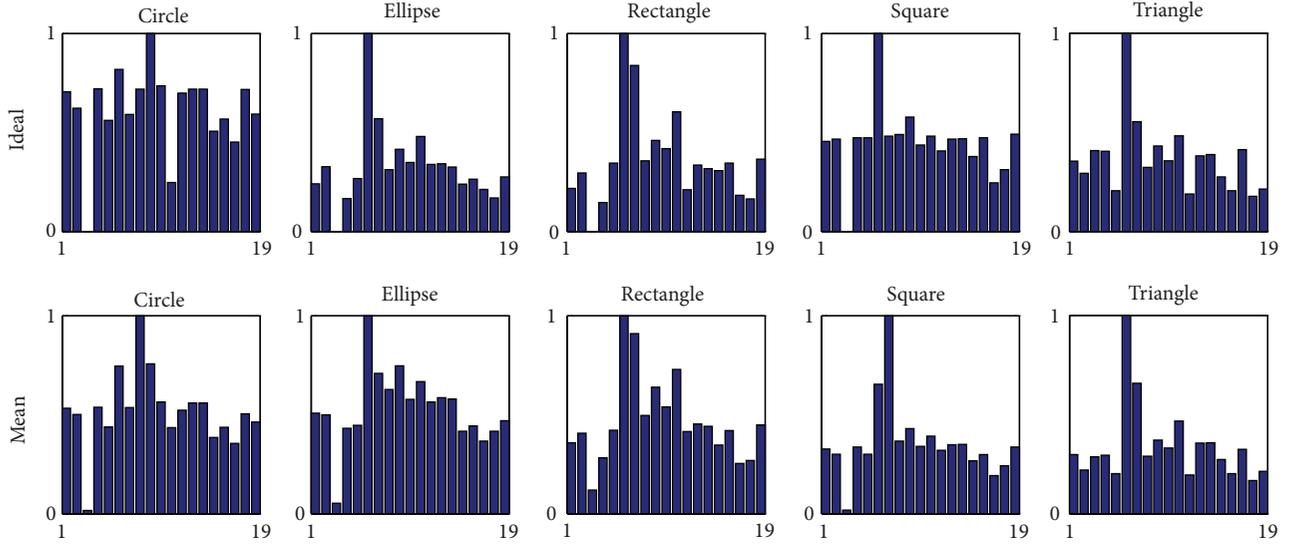


FIGURE 6: Features for each of five distinctive classes.

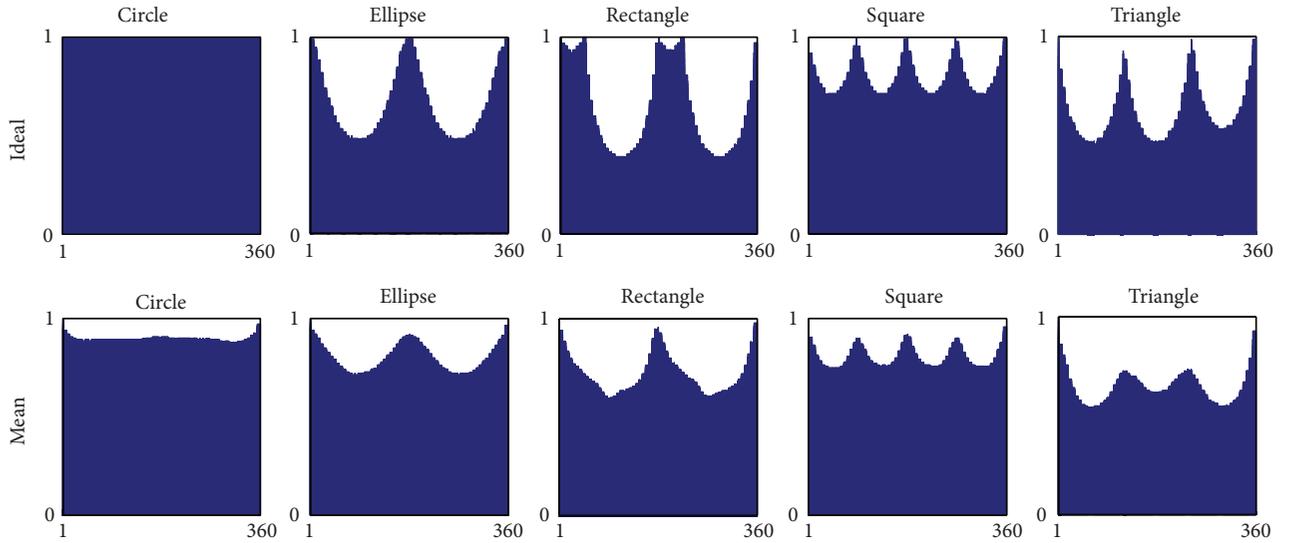


FIGURE 7: Shape signatures of five distinctive classes.

to [1, 8] interval. The GLCM is calculated according to the following formula [31]:

$$\begin{aligned} & \text{GLCM}_{\Delta x, \Delta y}(p, q) \\ &= \sum_{x=1}^M \sum_{y=1}^N \begin{cases} 1, & \text{if } I(x, y) = p, I(x + \Delta x, y + \Delta y) = q \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (5)$$

The contrast of GLCM is given as [31]

$$\text{GLCM}_C = \sum_p \sum_q (p - 1)^2 \text{GLCM}(p, q). \quad (6)$$

The energy is calculated as [31]

$$\text{GLCM}_E = \sum_p \sum_q \text{GLCM}(p, q)^2. \quad (7)$$

The homogeneity is given as [31]

$$\text{GLCM}_H = \sum_p \sum_q \frac{\text{GLCM}(p, q)}{1 + |p - q|}. \quad (8)$$

The correlation of GLCM is calculated as [31]

$$\text{GLCM}_H = \sum_p \sum_q \frac{(p - \mu_p)(q - \mu_q) \text{GLCM}(p, q)}{\sigma_p \sigma_q}, \quad (9)$$

where  $\mu_p$ ,  $\sigma_p$  and  $\mu_q$ ,  $\sigma_q$  are mean and standard deviation calculated in row and column directions, respectively.

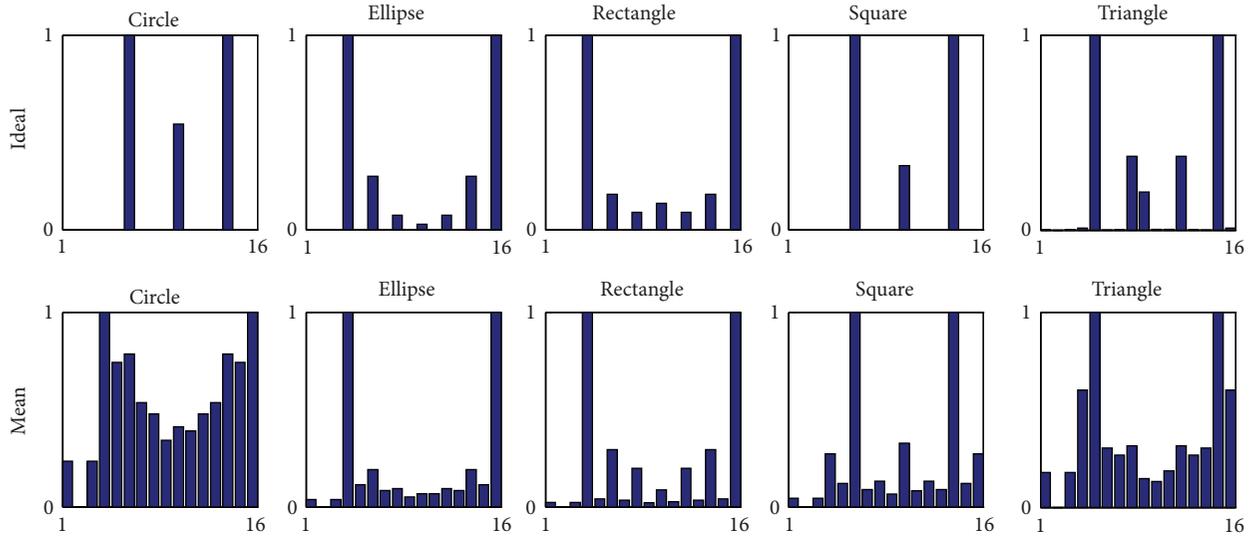


FIGURE 8: Fourier descriptors of five distinctive classes.

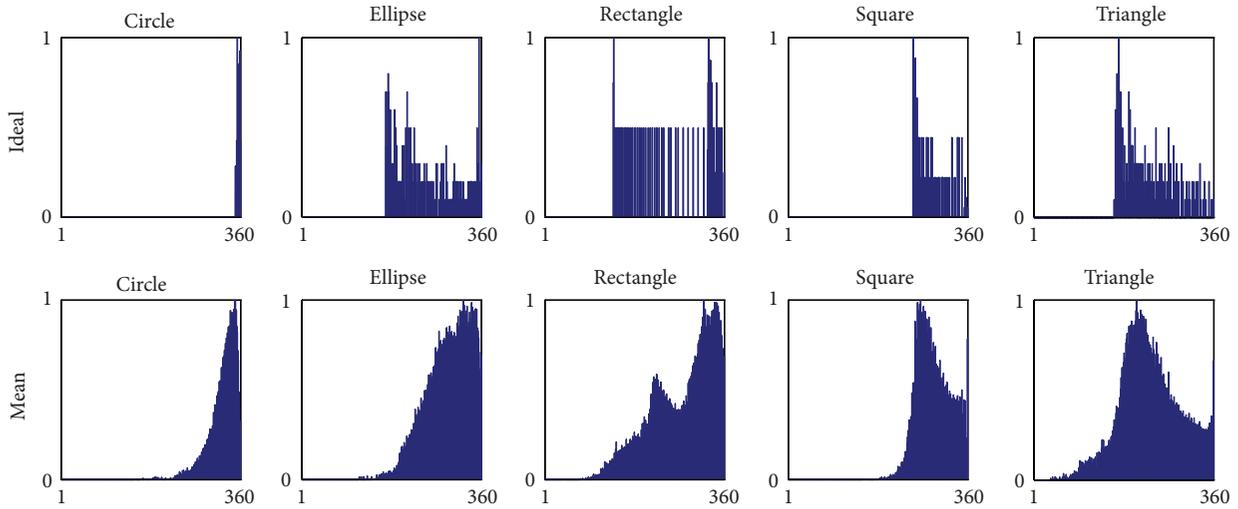


FIGURE 9: Point distance histograms of five distinctive classes.

### 3. Stamps Shape Features

In this paper we focus on the low-level image features employed in classification as it seems to be, according to our opinion, the most challenging problem. Such low-level properties can be easily computed and stored in limited memory. This two criteria are crucial when it comes, for example, for hardware implementation. Moreover, features presented below were selected, since they showed their abilities in other computer vision tasks.

In case of stamps, shape is one of the most valuable properties. Shape of stamp used for recognition can be considered as a binary object stored in a matrix, which can be represented by a specific number of points, including its interior or as a boundary (outer contour). Compact representation of shape is often known as shape descriptor. It is crucial for the recognition or classification to uniquely characterize shape

and stay invariant to as many transformations as possible (i.e., translation, scaling, rotation, presence of noise, and occlusion). These distortions are considered as differences between object under recognition and the reference object belonging to the same class, stored in a database. In practical recognition of stamps one has to take into consideration the following distortions divided into three main categories. The first one includes spatial transformations of an object, mainly translation, rotation in the image plane, and change of scale. The second category includes distortions introduced by imaging system, for example, variable number of captured points, presence of noise, discontinuity of contour, and occlusion. The third category of problems comes from contour representation and contour evaluation. The elements of the second group are the most challenging and difficult to solve.

Shape descriptors can be classified in various ways. The first taxonomy is based on mentioned earlier difference



FIGURE 10: Sample scanned documents used for the performance evaluation.

between object boundary and the whole shape. The second very popular classification (as described in [32]) divides descriptors into global approaches (shape represented as a whole) or structural methods (set of primitives). The third one discriminates spatial and transform domains [13]. Since shape is one of the most important features in problems related to content-based image retrieval, there are many known shape representations and retrieval methods; however, most of those methods neither represent shape in a sufficiently precise manner nor are relatively easy in matching. Among them, methods based on moments, polar coordinate system representation, and histogram of distances from the centroid achieve both good representation and easy normalization.

As it was mentioned earlier, there are two main classes of shape descriptors which capture different features, namely, region-based shape representation and contour-based ones. In region based techniques, all pixels within a shape are taken into consideration to obtain final shape representation. Most of region-based methods employ different variants of moments calculation. Contour-based shape representation

exploits shape boundary information. Such methods can be classified into global shape descriptors, shape signatures, and spectral descriptors. Although global descriptors such as area, circularity, eccentricity, and axis orientation are simple to compute and also robust in representation, they can only discriminate shapes with large dissimilarities and therefore are usually suitable for filtering purpose. Most shape signatures such as complex coordinates, curvature, and angular representations are essentially local representations of shape features; they are sensitive to noise and not robust. In addition, shape representation using shape signatures requires intensive computation during similarity calculation, due to the hard normalization of rotation invariance. As a result, these representations need often further processing. On the other hand, there are plenty of methods (e.g., MPEG-7 descriptors [33]) that are reported to be very efficient in shape retrieval. Unfortunately, their computational complexity is not justified in the task addressed in this paper.

**3.1. Simple Scalar Features.** In the shape analysis module we calculate several elementary measures that are later evaluated

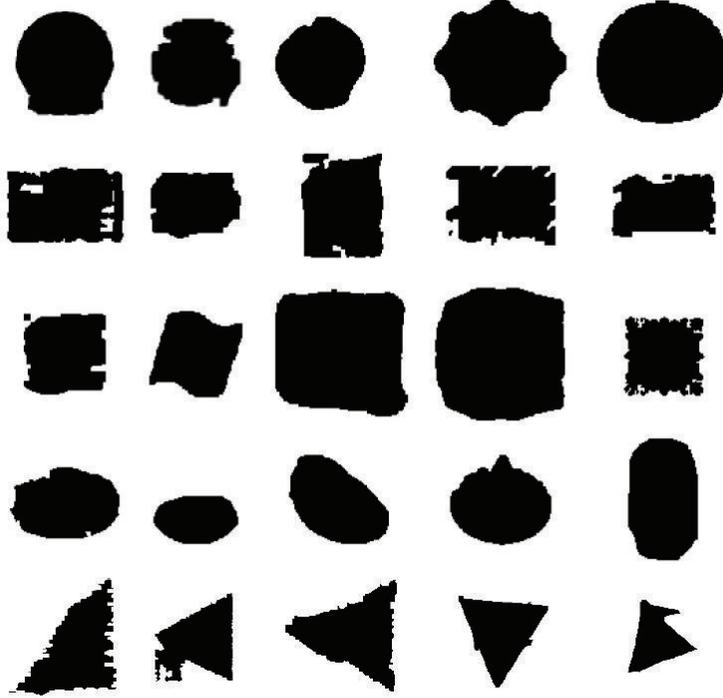


FIGURE 11: Sample shapes used for the performance evaluation: circle, rectangle, square, ellipse, and triangle (in rows, resp.).

at the decision stage. These simple properties are later concatenated in a single vector having 19 elements [34]. All of them use the following measures as common values ( $M_1$ : minor axis length,  $M_2$ : major axis length,  $A_p$ : object area defined by the number of pixels, and  $P$ : object perimeter), which are later employed to build more complex characteristics. Above characteristics were chosen intentionally, since they give a maximal discriminative power in terms of shape classification. All of them are normalized to the interval  $[0, 1]$ . It should be also noted that in case of a bounding box (associated with an analysed blob), that is not rotated,  $M_1$  is equal to  $\min(W_{\text{box}}, H_{\text{box}})$  and  $M_2$  is equal to  $\max(W_{\text{box}}, H_{\text{box}})$ .

**3.1.1. Roundness.** It is computed as an average value of the three following measures  $R_1$ ,  $R_2$ , and  $R_3$ :

$$\begin{aligned} R_1 &= \left| 1 - \frac{|M_2 - M_1|}{\max(M_1, M_2)} \right|, \\ R_2 &= \left| 1 - \frac{|\pi((M_1 + M_2)/2)^2/2 - A_p|}{A_p} \right|, \\ R_3 &= \left| 1 - \frac{|\pi \max(M_1, M_2) - P|}{P} \right|, \end{aligned} \quad (10)$$

where  $R_1$  expresses normalized difference between longest and shortest diameter;  $R_2$  incorporates information about object's area measured in two different ways, as a number of pixels belonging to the area and calculated according to geometrical formula;  $R_3$  is analogous to the  $R_2$  measure but perimeter is used instead of area.

**3.1.2. Squareness.** Squareness, as a measure of similarity between analysed object and a perfect square, is calculated according to the following formula:

$$S = \left( \frac{\min(\sqrt{A_p}, P/4)}{\max(\sqrt{A_p}, P/4)} \right)^2. \quad (11)$$

**3.1.3. Number of Vertices.** It is calculated as a function of object's extreme points (in terms of geometry); that is, if an object contains three extremes, then it is considered to be a triangle. The observations show, however, that binary objects extracted from real documents are often noised or distorted; hence finding extremes is very difficult or the number of detected extremes can differ from the actual one. In order to count the extremes, an input object is binarized, and then we select all potential extreme points (their coordinates) and create a new binary matrix containing zeros. Next, we put ones in the cells that match found coordinates. Further we cluster the points to their groups based on their spatial location. At this stage we perform a dilation procedure in order to fill gaps and create groups of neighboring pixels. Then, based on those groups, the number of extremes is computed.

**3.1.4. Aspect Ratio.** It is a very elementary property that is calculated as the proportion of object's width and height (in the similar manner to the one presented in Section 2.2):

$$R = \frac{M_1}{M_2}. \quad (12)$$



FIGURE 12: Sample stamps correctly recognized at the shape analysis stage.



FIGURE 13: Sample stamps misclassified at the shape analysis stage.

3.1.5. *Extent*. This property represents a ratio of the number of pixels inside a rectangle that is circumscribed around an object to the total number of pixels belonging to the object itself:

$$E_X = \frac{H_{\text{box}} W_{\text{box}}}{A_p} \quad (13)$$

3.1.6. *Moment-Based Properties*. Further measures that are later employed include ellipticity, elliptic variance, circular variance, triangularity coefficient (based on central moments), and minimal bounding figure, a coefficient defining a ratio between the area of a smallest shape that is wrapped around the object and this object's area. They involve the use of moment invariants  $\mu_{pq}$  (of second order), where each contour point  $i \in 1, \dots, N$  is described by its coordinates  $p_i = (x, y)$ . The centroid is calculated as a mean of all points coordinates:  $\mu = (1/N) \sum_i p_i$ , and the mean radius is equal to  $\mu_r = (1/N) \sum_i \|p_i - \mu\|$ .

*Ellipticity* proposed in [35] is calculated using the following formula:

$$I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^A} \quad (14)$$

The moment for a unit radius circle allows computing  $I_1 = 1/16\pi^2$  for a perfect figure. Ellipticity value ranges over  $[0, 1]$  and is computed as follows [35]:

$$E_I = \begin{cases} 16\pi^2 I_1 & \text{if } I_1 \leq \frac{1}{16\pi^2} \\ \frac{1}{16\pi^2 I_1} & \text{otherwise.} \end{cases} \quad (15)$$

*Circular variance*, as a similarity between an analysed object and a circle, was described in [36]. It is calculated using the following formula:

$$C_{\text{var}} = \frac{1}{N\mu_r^2} \sum_i (\|p_i - \mu\| - \mu_r)^2. \quad (16)$$

*Elliptic variance* responsible for describing ellipses employs the following formula [36]:

$$E_{\text{var}} = \frac{1}{N\mu_{rC}} \sum_i \left( \sqrt{(p_i - \mu)^T C^{-1} (p_i - \mu)} - \mu_{rC} \right)^2, \quad (17)$$

where  $C = (1/N) \sum_i (p_i - \mu)(p_i - \mu)^T$  is a covariance matrix, while  $\mu_{rC} = (1/N) \sum_i \sqrt{(p_i - \mu)^T C^{-1} (p_i - \mu)}$ .

*Triangularity* is based on moment invariants [35] and for an ideal right-angled triangle is equal to  $I_1 = 1/108$ ; thus

the triangularity of an analysed object is computed as follows (again, giving values from the interval [0, 1]):

$$T_I = \begin{cases} 108I_1 & \text{if } I_1 \leq \frac{1}{16\pi^2} \\ \frac{1}{108I_1} & \text{otherwise.} \end{cases} \quad (18)$$

*Minimal bounding figure coefficient* is calculated as a ratio of object's actual area to the area of an ideal figure that is circumscribed on it. It is called minimal bounding figure and the method of calculation is similar to [35]. The algorithm checks the similarity between each perfect figure (one of the predefined five classes) and an analysed object.

**3.1.7. Other Properties.** In the shape analysis we employ also a coefficient  $A_p$  related to the object's area understood as the number of pixels belonging to it. The following group of geometrical features is calculated on the basis of object's area, its shortest radius (span)  $R_s$  and its longest radius (span)  $R_l$ . If the difference  $S_{\text{cir}} = R_s - R_l$  is small enough (in our case, lower than 10 pixels), we assume that the object has circular shape. On the other hand, an object is considered to be a square if a value of  $S_{\text{sqr}}$  from the following equation is close to one:

$$S_{\text{sqr}} = \frac{A_p}{4R_s^2}. \quad (19)$$

An ellipse is described as a coefficient:

$$S_{\text{eli}} = \frac{A_p}{\pi R_s R_l}. \quad (20)$$

If it is close to one, then the object is recognized as an ellipse. In the same manner, the coefficients related to other shapes, like rectangle, triangle, and rhombus (which, for classification purposes, is treated as a square) can be calculated, respectively, as

$$\begin{aligned} S_{\text{rct}} &= \frac{A_p}{4R_s \sqrt{R_l^2 - R_s^2}}, \\ S_{\text{tri}} &= \frac{A_p \sqrt{3}}{(R_l + R_s)^2}, \\ S_{\text{rho}} &= \frac{A_p \sqrt{R_l^2 - R_s^2}}{2R_l^2 R_s}. \end{aligned} \quad (21)$$

All of above scalar features are joined into single vector and further used at the classification stage. The feature values for ideal shapes (circle, ellipse, rectangle, square, and triangle) and mean values for shapes from our experimental database are presented in Figure 6. In each plot, the horizontal axis is related to the feature number, while the vertical axis is related to the feature value. The ideal shapes were extracted from images created in painting software, while the mean shapes are averaged objects from our database.

**3.2. Vector Features.** As for the comparison we selected three popular vector shape descriptors: shape signature, Fourier descriptor, and point distance histogram. All of them have been successfully applied in many shape recognition tasks; hence the comparison is justified.

**3.2.1. Shape Signature.** Shape signature (SSig) is one of the most popular representations that belong to the contour-based class of descriptors. There are several variants of SSig which employ different features [32]. Here, we use the so-called Centroid Distance Function (CDF). It is easy to calculate and after some normalization can be invariant to scaling and rotation. However it should be noted that, in addition to the high matching cost, shape signatures are sensitive to noise, and slight changes in the boundary can cause large errors in matching. Hence, shape signatures should be stored in a reduced form. In this paper we calculate SSig according to the following algorithm.

- (1) Calculate the centroid of an object.
- (2) Detect outer contour of object and store its coordinates in a polar system of coordinates  $(r, t)$ .
- (3) Find the maximal distance  $r$  and perform circular shift so the  $t$  related to this maximum occupies the first position in the vector.
- (4) Discard the information about  $t$ ; hence remember only distances  $r$ .
- (5) Normalize the vector to the maximal value of  $r$ .
- (6) Interpolate the vector containing  $r$  to the final length of  $b$  elements.

The shape signature feature vectors with 360 bins for ideal shapes (circle, ellipse, rectangle, square, and triangle) and mean vectors for shapes from our experimental database are presented in (Figure 7). In each plot, the horizontal axis is related to the angle between radius vector and the horizontal direction, while the vertical axis is related to the distance from the origin. The descriptors for ideal shapes were extracted from images created in painting software, while the mean shapes are averaged objects from our database.

**3.2.2. Fourier Descriptor.** There is a whole family of descriptors called Fourier descriptors (FDs). Different shape signatures have been exploited to derive such descriptors. It should be noticed that FDs derived from different signatures can have significantly different effect on the result of retrieval [37]. In this paper we calculate FD according to the following algorithm.

- (1) Calculate the centroid of an object.
- (2) Detect outer contour of object and store its coordinates  $(x, y)$  as complex numbers, where  $x$  coordinate is a real part and  $y$  coordinate is an imaginary part.
- (3) Perform fast Fourier transform (FFT) on these values.
- (4) Normalize FFT spectrum to its maximal magnitude.
- (5) Remember first  $b$  elements (related to low frequency components).

TABLE 1: Classifiers comparison in terms of stamp/no-stamp verification.

Classifier	TP rate	FP rate	Classifier	TP rate	FP rate
NB	91.8%	49.6%	TR	79.0%	21.9%
LLR	84.1%	37.7%	CART	82.9%	23.8%
MLP	86.5%	26.9%	$k$ NN, $k = 1$	83.9%	16.0%
SVM	45.6%	54.6%	$k$ NN, $k = 3$	84.9%	21.3%

The Fourier descriptor feature vectors with 16 bins for ideal shapes and mean vectors for our experimental database are presented in (Figure 8). In each plot, the horizontal axis is related to the spectral component, while the vertical axis is related to its amplitude. The descriptors for ideal shapes were extracted from images created in painting software, while the mean objects were averaged over the whole database.

**3.2.3. Point Distance Histogram.** The point distance histogram (PDH) is a histogram of contour point distances represented in the polar system of coordinates. The algorithm of calculating PDH is as follows [21].

- (1) Calculate the centroid of an object.
- (2) Detect outer contour of object and store its coordinates in a polar system of coordinates  $(r, t)$ .
- (3) Discard the information about  $t$ ; hence remember only distances  $r$ .
- (4) Calculate the histogram  $H$  of distances  $r$  with  $b$  bins.
- (5) Normalize  $H$  to the maximal value.

The PDH feature vectors with 360 bins for ideal shapes and mean vectors for our experimental database are presented in (Figure 9). In each plot, the horizontal axis is related to the distance (or distance interval), while the vertical axis is related to the number of points within such distance. As above, the descriptors for ideal shapes were calculated for images created in painting software, while the mean objects are averaged over the whole our database.

## 4. Experiments

**4.1. Experimental Setup.** The experiments were aimed at the evaluation of the performance of both stages of processing, namely, the stamp detector and stamp's shape classifier. Since there are no benchmark databases oriented at this specific problem, the experiments were performed on our own benchmark database consisting of a number of images collected from the Internet. It contains scanned or photographed documents of different origin and variable quality. The details of this database are as follows: documents with no stamps: 294 (41%), with single stamp: 309 (43%), with multiple stamps: 116 (16%), and with logotypes: 367 (51%). Exemplary documents gathered in the database are presented in Figure 10.

**4.2. Stamp/No-Stamp Verification.** The experiments on stamp verification were performed on 2925 graphical objects extracted from the above presented documents. There are 1589 stamps and 1336 no-stamps (logos, ornaments, and

TABLE 2: Confusion matrix with True Positive and False Positive indicators for INN classifier.

Analysed	Predicted		False Positives	True Positives
	Stamps	No-stamps		
Stamps	1326	263	15.4%	83.4%
No-stamps	206	1130	16.6%	84.6%

large letters) in the set. They were divided into learning and testing subsets according to the 10-fold cross-validation.

In order to select a most suitable classifier we performed several experiments involving several state-of-the-art approaches, namely, Naive Bayes Classifier (NBC), Linear Logistics Regression (LLR), Multilayer Perceptron (MLP), Support Vector Machines (SVM), Classification and Regression Tree (CART), Random Tree (RT), and  $k$ -Nearest Neighbors ( $k$ NN) with  $k = 1$  and  $k = 3$ . The rates of True Positive and False Positive for *Stamp* class are given in Table 1. The analysis of the results showed that the accuracy of SVM is significantly lower than any other classifiers (probably due to high variance). On the other hand, the high True Positive rates for NB and MLP are accompanied by also high False Positive rates. Based on these results at the stage of stamp/no-stamp verification, we selected  $k$ NN with  $k = 1$  as a method with the lowest FP rate. During experiments employing INN, the algorithm was able to recognize 2456 objects, while the rest 469 were misclassified. For such experimental setup, the detailed results are presented in Table 2.

**4.3. Shape Classification.** We performed a shape classification on a database created from stamps extracted from the aforementioned documents. The set consists of 2101 binary objects. Exemplary objects are presented in Figure 11. For each object we calculated a scalar-based feature vector of 19 properties and three vector descriptors shown above. In order to select the most discriminative method we tested several modern classifiers and, as for comparison, simple voting classifier. The later uses simple scalar weights equal to one for all simple features, except the following, which have weights equal to 3:  $R$ ,  $S_{\text{sqr}}$ ,  $S_{\text{rec}}$ ,  $S_{\text{eli}}$ ,  $T_I$ , minimal bounding triangle, minimal bounding circle, and minimal bounding square (from the group of minimal bounding figure). In Table 3 one can observe a comparison of different classifiers versus selected features. The column *Simple* contains the results of simple shape features described in Section 3.1.

As it can be observed, the performance of all classifiers in case of simple scalar features is superior to selected vector features. It is also visible that the classifiers ensembles based

TABLE 3: Classification performance [%].

Classifier	Simple	SSig	FD	PDH
Bayes network (K2 learning rule)	<b>91.9</b>	83.3	87.4	81.7
MLP (single hidden layer)	<b>95.7</b>	94.7	94.1	88.9
SVM (Sequential Minimal Optimization, polynomial kernel)	<b>95.0</b>	84.6	91.7	82.1
INN (1-Nearest Neighbor, Euclidean Distance)	<b>96.6</b>	95.5	95.2	89.9
KStar	<b>97.0</b>	95.1	94.0	90.3
Bagging (Fast Decision Tree Learner)	<b>95.4</b>	91.6	93.5	89.0
RandomCommittee (RandomTree Classifiers)	<b>97.7</b>	95.9	95.4	92.1
RotationForest (C4.5 Decision Tree, Principal Components Analysis)	<b>97.3</b>	96.1	95.6	90.4
Nearest Neighbor with Generalization	<b>96.4</b>	91.8	93.8	89.5
RandomForest (10 trees)	<b>97.4</b>	95.0	95.1	90.4

TABLE 4: Confusion matrix for shape analysis module based on a Committee of RandomTree Classifiers.

Analysed shape	Predicted					Class accuracy
	Circle	Ellipse	Square	Triangle	Rectangle	
Circle	434	9	3	0	0	97%
Ellipse	14	400	0	0	3	96%
Square	2	0	393	0	7	98%
Triangle	0	0	0	321	0	100%
Rectangle	0	4	6	1	504	98%

TABLE 5: The performance of the whole algorithm.

	Detected	Missed	Accepted (TP)	Rejected (FN)
Stamp	417/70.44%	175/29.56%	313/75.06%	104/24.94%
	Detected	Missed	Accepted (FP)	Rejected (TN)
No-stamp	377	—	129/34.22%	248/65.78%

on nearest neighbours can give significantly higher accuracy than any other approach.

Table 4 shows the confusion matrix for shape classification in terms of particular types of stamps. The detailed results show that the highest confusion is observable for shapes pairs, for example, circle versus ellipse and square versus rectangle. It comes from the fact that the boundaries of those pairs of classes may be fuzzy.

**4.4. Overall Performance.** In the final experiment we wanted to evaluate the overall performance of the proposed approach. It included both stages: stamp detection and shape classification. It was performed on a set of 719 scanned or photographed documents. The results are presented in Table 5.

Tables 4 and 5 prove that the proposed set of algorithms is able to detect and classify most of stamps. A considerable number of missed stamps comes from the fact that the database consists of many problematic cases, that is, stamps which are hardly visible, only partially imprinted, or overlap objects such as pictures, post stamps, or printed text (see examples in Figures 12 and 13).

A special case is stamps containing small tables or just texts. It is almost impossible to distinguish them from actual

tables or parts of text. Hence, it leads to the further misclassification. The decrease in classification performance is caused also by the fact that the database contains documents completely without stamps, documents with objects similar to stamps, or actual stamps that do not meet conditions specified in Section 2.2. As it can be seen from the observations, a large part of the original documents has a low contrast. Moreover, many wrongly detected areas (later rejected by the classifier) are simply the artifacts coming from strong compression of the input image.

**4.5. Comparison with State-of-the-Art.** We compared our approach with some of the works presented in the literature. Since it was impossible to repeat all the experiments presented in the papers, we took the results provided by the authors. They are gathered in Table 6. Unfortunately, not all details of compared algorithms could be obtained. Such cases are marked with a “—” sign. The comparison is divided into three areas: stamp detection, stamp verification, and stamp shape classification. In our comparison we included the numbers of input documents, verified objects (stamps/no-stamps), and extracted stamps. Actually, not all elements can be properly compared, since as for now a comprehensive method has not been presented and the methods given in the literature focus on individual actions only.

A full comparison of stamp detection is hard to perform, since several other methods allow detecting only color stamps. The reported accuracy of detection of our method is lower; however our database is the largest and we detect stamps not taking into account their color or overlapping. On the other hand, the actual accuracy of method presented in [3] is in fact equal to 69%.

TABLE 6: Comparison with other approaches, detect refers to detection stage, verify refers to verification stage, and shape refers to shape analysis stage.

	Indicator	Algorithm					
		GSA [6]	PDH & DCT [25]	[3]	GHT & CPG [4]	[5]	Our
Detect	Number of doc.	170	170	400	—	127	719
	Accur. [%]	82	82	83.4 (69)	—	—	70.44
Verify	Number of obj.	—	—	—	—	—	1326
	Accur. [%]	—	—	83.8	—	—	83.4
Shape	Number of stamps	—	140	—	220	—	2101
	Accuar. [%]	93	77	—	98.58	92.03	97.7

As it can be noticed, no method found in the literature performs an appropriate verification step (discrimination between stamps and no-stamps). Our method employs a set of image features to distinguish stamps from objects like logotypes, small tables, and other graphical ornaments.

Finally, our shape recognition algorithm was tested on the largest database among all methods and gave a very competitive results.

## 5. Summary

In the paper we propose a novel approach for detecting and classifying stamp instances in scanned documents. To our best knowledge, this is the first comprehensive algorithm that deals with color and monochrome stamps of any shape. It incorporates several methods of image processing, pattern recognition, and some heuristic. The algorithm is multistage, consisting of object detection, stamp verification, and shape classification. During experiments we selected a set of low-complexity features that are invariant to stamp's color, orientation, scale, and position in the document; hence this approach is independent of the document size and orientation. Such basic low-level features containing elementary shape properties occurred to be superior to several, more complex descriptors. The results of experiments showed also that at the stage of stamp/no-stamp verification the best results gave simple INN classifier and at the stage of shape classification, classifiers ensembles utilizing Random Trees. The experiments performed on a large number of real scanned and photographed documents show that the performance is sufficient for practical implementation. Future works may include the investigations on the additional features of stamps, which can give an increase in the overall robustness, especially at the stage of stamp/no-stamp verification. Above approach could also be employed for recognizing different parts of documents (e.g., texts, tables, signatures, logotypes, etc.). The potential area of application of this algorithm includes office software for processing paper documents, content-based document retrieval systems, and various postal services. It could be also employed in cases, when stamps should be automatically covered in order to preserve privacy.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] K. Ueda and Y. Nakamura, "Automatic verification of seal impression patterns," in *Proceedings of the 7th International Conference on Pattern Recognition.*, pp. 1019–1021.
- [2] S. Dey, J. Mukherjee, S. Sural, and P. Bhowmick, "Colored rubber stamp removal from document images," in *Pattern Recognition and Machine Intelligence*, vol. 8251 of *Lecture Notes in Computer Science*, pp. 545–550, Springer, Berlin, Germany, 2013.
- [3] B. Micenková and J. van Beusekom, "Stamp detection in color document images," in *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR '11)*, pp. 1125–1129, Beijing, China, September 2011.
- [4] P. P. Roy, U. Pal, and J. Lladós, "Document seal detection using GHT and character proximity graphs," *Pattern Recognition*, vol. 44, no. 6, pp. 1282–1295, 2011.
- [5] P. P. Roy, U. Pal, and J. Lladós, "Seal detection and recognition: an approach for document indexing," in *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR '09)*, pp. 101–105, July 2009.
- [6] P. Forczmański and D. Frejlichowski, "Robust stamps detection and classification by means of general shape analysis," in *Computer Vision and Graphics*, vol. 6374 of *Lecture Notes in Computer Science*, pp. 360–367, Springer, Berlin, Germany, 2010.
- [7] L. Cai and L. Mei, "A robust registration and detection method for color seal verification," in *Proceedings of the International Conference on Intelligent Computing (ICIC '05)*, pp. 97–106, August 2005.
- [8] S. Dickinson and Z. Pizlo, *Shape Perception in Human and Computer Vision, An Interdisciplinary Perspective*, Advances in Computer Vision and Pattern Recognition, Springer, London, UK, 2013.
- [9] M. del Pozo-Baños, J. Ticay-Rivas, J. Cabrera-Falcon et al., "Image processing for pollen classification," in *Biodiversity Enrichment in a Diverse World*, G. A. Lameed, Ed., InTech, Rijeka, Croatia, 2012.
- [10] M. Rodríguez-Damián, E. Cernadas, A. Formella, and P. Sá-Otero, "Pollen classification using brightness-based and shape-based descriptors," in *Proceedings of the 17th International*

- Conference on Pattern Recognition (ICPR '04)*, pp. 212–215, August 2004.
- [11] D. Frejlichowski, “An algorithm for the automatic analysis of characters located on car license plates,” in *Image Analysis and Recognition*, M. Kamel and A. Campilho, Eds., vol. 7950 of *Lecture Notes in Computer Science*, pp. 774–781, Springer, Berlin, Germany, 2013.
- [12] G. McNeill and S. Vijayakumar, “2D shape classification and retrieval,” in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI '05)*, pp. 1483–1488, August 2005.
- [13] B. M. Mehtre, M. S. Kankanhalli, and W. F. Lee, “Shape measures for content based image retrieval: a comparison,” *Information Processing and Management*, vol. 33, no. 3, pp. 319–337, 1997.
- [14] J. He and A. C. Downton, “Configurable text stamp identification tool with application of fuzzy logic,” in *Document Analysis Systems VI*, vol. 3163 of *Lecture Notes in Computer Science*, pp. 134–151, Springer, Berlin, Germany, 2004.
- [15] S. Seiden, M. Dillencourt, S. Irani, R. Borrey, and T. Murphy, “Logo detection in document images,” in *Proceedings of the International Conference on Imaging Science, Systems, and Technology*, pp. 446–449, 1997.
- [16] G. Zhu and D. Doermann, “Automatic document logo detection,” in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR '07)*, pp. 864–868, September 2007.
- [17] R. Jain and D. Doermann, “Logo retrieval in document images,” in *Proceedings of the 10th IAPR International Workshop on Document Analysis Systems (DAS '12)*, pp. 135–139, March 2012.
- [18] K. Jeong and H. Moon, “Object detection using FAST corner detector based on smartphone platforms,” in *Proceedings of the 1st ACIS/JNU International Conference on Computers, Networks, Systems, and Industrial Engineering (CNSI '11)*, pp. 111–115, May 2011.
- [19] J. Kleban, X. Xie, and W.-Y. Ma, “Spatial pyramid mining for logo detection in natural scenes,” in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '08)*, pp. 1077–1080, Hannover, Germany, June 2008.
- [20] S. Hassanzadeh and H. Pourghassem, “A novel logo detection and recognition framework for separated part logos in document images,” *Australian Journal of Basic and Applied Sciences*, vol. 5, no. 9, pp. 936–946, 2011.
- [21] D. Frejlichowski, “An experimental comparison of seven shape descriptors in the general shape analysis problem,” in *Image Analysis and Recognition—ICIAR 2010, Part I*, A. Campilho and M. Kamel, Eds., vol. 6111 of *Lecture Notes in Computer Science*, pp. 294–305, Springer, Berlin, Germany, 2010.
- [22] T. D. Pham, “Unconstrained logo detection in document images,” *Pattern Recognition*, vol. 36, no. 12, pp. 3023–3025, 2003.
- [23] G. Zhu, S. Jaeger, and D. Doermann, “A robust stamp detection framework on degraded documents,” in *Proceedings of the International Conference on Document Recognition and Retrieval XIII*, pp. 1–9, 2006.
- [24] D. Frejlichowski and P. Forczmański, “General shape analysis applied to stamps retrieval from scanned documents,” in *Artificial Intelligence: Methodology, Systems, and Applications: 14th International Conference, AIMSA 2010, Varna, Bulgaria, September 8–10*, vol. 6304 of *Lecture Notes in Computer Science*, pp. 251–260, Springer, Berlin, Germany, 2010.
- [25] P. Forczmański and D. Frejlichowski, “Efficient stamps classification by means of point distance histogram and discrete cosine transform,” in *Computer Recognition Systems 4*, vol. 95 of *Advances in Intelligent and Soft Computing*, pp. 327–336, Springer, Berlin, Germany, 2011.
- [26] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [27] J. Illingworth and J. Kittler, “A survey of the hough transform,” *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 1, pp. 87–116, 1988.
- [28] R. O. Duda and P. E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [29] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.
- [30] I. Sobel and G. Feldman, “A  $3 \times 3$  isotropic gradient operator for image processing,” Presentation to Stanford Artificial Intelligence Project 01/1968, 1968.
- [31] F. Albreghsen, “Statistical texture measures computed from gray level cooccurrence matrices,” Image Processing Laboratory, Department of Informatics, University of Oslo, pp. 1–14, 1995.
- [32] D. Zhang and G. Lu, “Review of shape representation and description techniques,” *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004.
- [33] M. Bober, “MPEG-7 visual shape descriptors,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 716–719, 2001.
- [34] P. Forczmański and A. Markiewicz, “Low-Level image features for stamps detection and classification,” *Advances in Intelligent Systems and Computing*, vol. 226, pp. 383–392, 2013.
- [35] P. L. Rosin, “Measuring shape: ellipticity, rectangularity, and tripartularity,” *Machine Vision and Applications*, vol. 14, no. 3, pp. 172–184, 2003.
- [36] M. Peura and J. Iivarinen, “Efficiency of simple shape descriptors,” in *Aspects of Visual Form*, pp. 443–451, World Scientific, 1997.
- [37] D. Zhang and G. Lu, “A Comparative Study on Shape Retrieval using fourier descriptors with different shape signatures,” in *Proceedings of the 5th Asian Conference on Computer Vision (ACCV '02)*, pp. 646–651, 2002.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

