

Video-Based Content Recognition of Bank Cards with Mobile Devices

H. Chen^{a,*}, S. Ye^{a,**}, A. Kurilovich^{b,***}, R. Bohush^{b,****}, and S. Ablameyko^{c,d,*****}

^a Zhejiang Shuren University, Shuren St. 8, Hangzhou, 310015 China

^b Polotsk State University, ul. Blokhina 29, Novopolotsk, 211440 Belarus

^c Belarusian State University, pr. Nezavisimosti 4, Minsk, 220030 Belarus

^d United Institute of Informatics Problems, National Academy of Sciences of Belarus,
ul. Surganova 6, Minsk, 220012 Belarus

*e-mail: eric.hf.chen@hotmail.com

**e-mail: zjsruysp@163.com

***e-mail: cfif921@yandex.ru

****e-mail: bogushr@mail.ru

*****e-mail: ablameyko@bsu.by

Received October 8, 2019; revised November 5, 2019; accepted December 5, 2019

Abstract—In this paper, we propose an algorithm for detection and recognition of all information fields at the bank card front side based on video sequences. The algorithm is intended for use on mobile devices. This algorithm consists of the following basic steps: detection of the card boundaries in a frame, segmenting the information fields, improving the quality of segments, localizing the boundaries of symbols, and recognizing blocks of symbols. We also conduct a series of experiments. Experimental results show that our algorithm can achieve higher detection rates of 88% for all information fields and 92.5% for the bank card number and expiration date. The processing time per frame at different resolutions for each step by using iPhone 7 is presented. The experimental results confirm the efficiency of the proposed approach.

DOI: 10.1134/S036176882006002X

1. INTRODUCTION

Progress in banking technologies has resulted in the widespread introduction of Internet banking, i.e., remote banking service via the Internet, and mobile banking, which allows the bank client to manage his or her bank accounts by using a tablet or smartphone. Nowadays, one client can have two or more bank cards, which he or she can use for payment. Filling in all data fields when paying for services or making purchases with mobile banking requires time and attention. Thus, the efficiency of mobile banking services can be improved by developing (algorithms and) software to automate the entry of bank card information into the system based on the analysis of bank card images obtained using mobile devices.

The design of a bank card is an important carrier of the bank's brand, which can have various color and luminance characteristics. Thus, the image on a bank card can be heterogeneous and quite complex. In addition, glossy plastic used in bank cards is highly reflective, which can cause glares and reflections on card images in bright light, especially when using a flash under low light conditions. In the presence of shadows

or insufficient illumination, the card image may have rather poor quality.

A bank card is a typical example of a document of flexible form. Hence, for its processing, we can employ the same detection and recognition algorithms that are used for flexible forms of other types [1]. In [2], an approach for segmentation and recognition of characters on business cards on a plain background was proposed. The algorithm consisted of the following basic steps: edge detection using the Sobel operator, line thinning, projective transformation, adaptive binarization, and segmentation of words and characters with their subsequent recognition. The disadvantages of this approach are its applicability only to simple-background images and its rather high computational requirements, which impose a significant constraint on its implementation in mobile devices. In [3], an algorithm for business card processing was proposed: at the first step of the algorithm, rough background removal is carried out by block analysis of an input image and, at the second step, the connected components extracted are classified to determine text regions and eliminate logo images. Then, the result undergoes adaptive binarization to separate the text from the

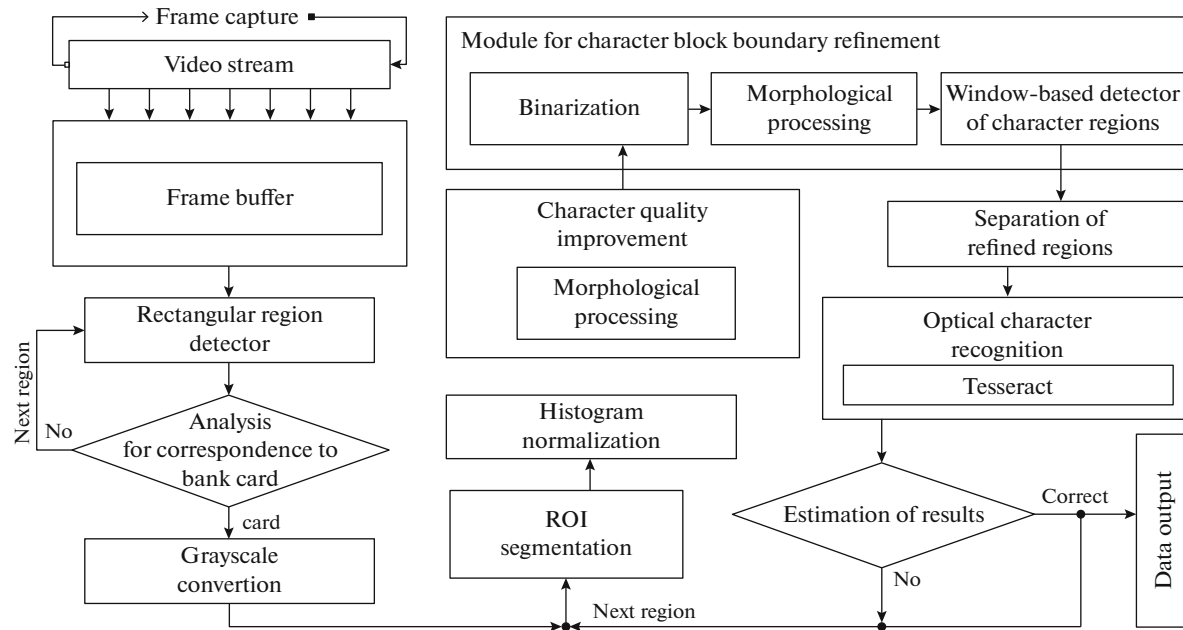


Fig. 1. Flowchart of the algorithm.

background for its further recognition. This algorithm can be employed only for card images with a monotone background. In addition, it does not use projective transformation, which imposes additional requirements on the orientation of an input card image with respect to the smartphone camera.

In [4, 5], methods for business card recognition on the Android platform were presented. In [4], the Mat-Lab package was used for character preprocessing and segmentation, while the Tesseract library was employed for character recognition. However, as noted in [4], the preprocessing procedure used computationally intensive operations for input image refinement and character segmentation, which is why a simplified version of this algorithm without certain preprocessing steps implemented on mobile devices had significantly lower performance. At the first step of the method proposed in [5], the size of an input image is reduced, with the preprocessing operation using OpenCV and the character recognition operation using Tesseract. However, the experiments in those works were carried out on business card images with a single-color background, the characters on which are clearly visible.

In [6], an algorithm for credit card expiration date recognition was proposed and its performance was evaluated using iPhone 4S. This algorithm does not support automated data input from a card image and, therefore, requires further development. The algorithms for card number recognition on Android devices considered in [7, 8] also do not recognize the first and last names of the cardholder. In [7], recognition accuracy reached 80%; in [8], the preliminary processing of card images and the use of a recurrent neural network provided

digit recognition accuracy up to 90%. It should also be noted that these algorithms were designed to process embossed cards. In [9], a method that uses reliability functions for embossed character recognition was proposed and its efficiency in improving the probability of correct recognition was demonstrated. Presently, non-embossed cards, which can be issued to the client on the day he or she contacts the bank, have become quite popular. However, due to the fact that the alphanumeric information on such cards is not embossed, its localization and recognition proves much more difficult and requires efficient preprocessing to localize characters and ensure their correct recognition.

As input data, existing approaches use static images, which sometimes contain noise, glares, significant luminance differences, insufficient sharpness, etc. This may require obtaining a new image, which, in turn, can also be unsuitable for correct detection of the card and information fields on it, as well as their further recognition. The efficiency of data processing can be improved by using video, which is easy to obtain using a mobile device. However, a sequence of images (rather than a single image) requires different approaches to its processing. In [10], it was proposed to use video sequences for document edge detection with mobile devices. In that case, the method includes image segmentation in the Lab color space by using morphological processing, linking individual segment boundaries based on the Hough transform, and boundary detection of the whole document. However, to reduce computational cost, the frame size of an input image was reduced from 1280×720 to 180×100 pixels. This reduction is not adequate for localizing and recognizing bank card information because the probability of

character recognition decreases with the decrease in the character size.

In this paper, we propose an algorithm for detecting and recognizing bank card details by video sequences recorded with a mobile device. The algorithm enables almost real-time recognition of all information fields on the front of a bank card, including its number and expiration date, as well as the first and second names of the cardholder in Latin and Cyrillic, for embossed and non-embossed cards.

2. ALGORITHM FOR RECOGNIZING BANK CARD DETAILS IN VIDEO SEQUENCES

Figure 1 shows the flowchart of our algorithm. A sequence of frames recorded using a mobile camera is input to a rectangular region detector, which detects all rectangles in a frame and returns only one that meets the bank card parameters.

This region is converted to a grayscale image, which is input to a segmentation block that selects and indexes the regions on the card image that correspond to the information fields containing the number of the card, its expiration date, and the first and last names of the cardholder. The next block carries out certain transformations that improve the contrast qualities of the input images and reduce the noise. Then, adaptive binarization and morphological processing are carried out. The next step refines the boundaries of the character regions by using the sliding window method. After that, the segmented regions are passed to an OCR block for digit and character recognition. Finally, a data estimation block processes the received information to display or reject the result. If the card number and expiration date extracted are valid, then the result for them is considered correct. The recognition of the cardholder's name stops once the first result is obtained. If all three information fields in the current frame are not recognized, then, in the next frame, only the missing fields are processed.

2.1. Card Detection

When developing the card detection algorithm, we took into account that the frame size of an input video sequence recorded with modern mobile devices can vary, reaching 3840×2160 pixels. These devices support automatic focusing, luminance adjustment, and white balance. Thus, the resulting image generally has acceptable characteristics (sharpness, luminance, and resolution) for automatic extraction of bank card data.

At the first step, all rectangular regions on the image are detected and localized. Taking into account possibly high frame resolution (up to 4 K), as well as limited computing power of mobile devices, we use a detection method based on the fast Viola–Jones [11] and OverFeat [12] algorithms, which is also characterized by high detection capabilities. Then, to determine the contour that bounds the bank card, the aspect

ratios of the extracted objects are analyzed. With the height m_o and width n_o of the card being standardized and its aspect ratio being constant, it is reasonable to use it as a criterion for finding the contour of the card among all rectangular regions detected. Thus, the $m_i \times n_i$ rectangular region r_i can be classified as a card image in the frame if the following condition is met:

$$\frac{n_o}{m_o} - e < \frac{n_i}{m_i} < \frac{n_o}{m_o} + e, \quad (2.1)$$

where e is a coefficient that characterizes the permissible deviation of the card's aspect ratio.

In this case, the result is a rectangle with the longest side m_{\max} among all rectangles detected. The size of the rectangular region found at this step and its position on the image $p_i(x_i, y_i)$ are used to extract the card region from the input frame.

2.2. Region of Interest (ROI) Segmentation

On the resulting $m_I \times n_I$ image I of the card, it is required to find the regions that contain the information on its number (I_C), expiration date (I_D), and name of its holder (I_E). The size and location of these regions (determined by ISO/IEC 7811-5.4:2018, which can be used for segmentation of these fields) are denoted as follows:

$C(x_C, y_C, m_C, n_C)$ is the card number;

$D(x_D, y_D, m_D, n_D)$ is the card expiration date;

$E(x_E, y_E, m_E, n_E)$ is the cardholder's name.

Since the values of m_I and n_I can vary in the process of videorecording, normalization is required to correctly determine C , D , and E . For this purpose, we use the following width scaling (c_{ws}) and height scaling (c_{hs}) factors, which are multiplied by the parameters of these regions:

$$c_{ws} = \frac{n_I}{n_o}, \quad c_{hs} = \frac{m_I}{m_o}. \quad (2.2)$$

The card number has 16 digits, divided into 4 equal groups of 4 digits each. Thus, the image I_C (extracted at the previous step) with the card number region C_I is divided into four equal subregions C_{I1} , C_{I2} , C_{I3} , and C_{I4} :

$$C_{Ij} \left(x_{CI} + \left(\frac{n_{CI}}{4} \times (j-1) \right), y_{CI}, \frac{n_{CI}}{4}, m_{CI} \right), \quad (2.3)$$

$$j = 1, 2, 3, 4.$$

Having extracted the computed regions from I_C , we obtain the segmented images I_{C1} , I_{C2} , I_{C3} , and I_{C4} of the card number groups.

2.3. Refinement of the Segmented Images

To reduce the amount of information to be processed, the resulting images (I_D , I_E , and $I_{C1}-I_{C4}$) are converted to grayscale with a further increase in their

contrast by the histogram normalization method, which stretches only the most informative region of intensity variation (rather than the entire intensity variation range), thus increasing the contrast effect due to the elimination of noise regions with rare intensities.

With this approach, the resulting gray level g for a pixel is determined as follows:

$$g_{x,y} = \frac{255(Y_{x,y} - Y_{c_min})}{Y_{c_max} - Y_{c_min}}, \quad (2.4)$$

where $Y_{x,y}$ is the luminance of the input pixel with the coordinates x and y , while Y_{c_min} and Y_{c_max} are the specified minimum and maximum luminance levels of the input image pixels, respectively.

To improve the processing result, Y_{c_min} and Y_{c_max} differ from the minimum and maximum luminance levels of the input image, which allows us to neglect a small number of pixels at the edges of the histogram.

Thus, Y_{c_min} and Y_{c_max} are defined as

$$Y_{c_min}(i) = \begin{cases} Y_{c_min}(i+1), & \text{if } \sum_{j=0}^i f_i(Y_j) < T \\ Y_i, & \text{otherwise;} \end{cases} \quad (2.5)$$

$$Y_{c_max}(i) = \begin{cases} Y_{c_max}(i-1), & \text{if } \sum_{j=255}^i f_i(Y_j) < T, \\ Y_i, & \text{otherwise,} \end{cases}$$

where f_i is the number of pixels on the image with the luminance Y_j , $T = mne$, and e is the permissible deviation (in %) determined experimentally.

Then, using mathematical morphology operators, the boundaries of the characters on the grayscale image obtained at the previous step are emphasized. To determine the colors of the background and characters, the mean luminance is evaluated. If the result exceeds 127, then the background color is white, while the character color is black, and vice versa.

If the color of the characters is determined as white, then the WhiteTopHat morphological transformation, which subtracts a morphologically open image from the original one to emphasize the contours of the white characters, is applied to the image. The BlackTopHat transformation is used to emphasize the boundaries of the black characters by subtracting the original image from a morphologically close image. The structuring element b for the kernels of both filters has a rectangular shape; its size $n_b \times m_b$, taking into account the experimentally determined scaling factors, is computed as follows:

$$n_b = 0.06n_I, \quad m_b = 3n_b, \quad (2.6)$$

where n_I is the width of the image to be transformed.

2.4. Boundary Localization

Since the luminance and contrast characteristics of the fragments in the bank card images, despite their small size, can significantly differ, the luminance levels of the pixels that form the background of the characters can also vary. Taking this into account, to preserve the boundaries of the characters, we use the image binarization method with adaptive threshold based on local region analysis:

$$bin_{x,y} = \begin{cases} 255, & \text{if } Y_{x,y} > T(x,y) \\ 0, & \text{otherwise,} \end{cases} \quad (2.7)$$

where $T(x,y)$ is the adaptive binarization threshold computed for each pixel as $T(x,y) = corr(I, G) - c$; $corr(I, G)$ is the cross correlation of a $k \times k$ local image fragment with the Gaussian window; and c is a constant. The $k \times k$ coefficient matrix of the Gaussian window is defined as follows [13]:

$$G_i = \alpha e^{-\frac{(i - \frac{k-1}{2})^2}{2\sigma^2}}, \quad (2.8)$$

$$\sigma = 0.3((k-1)0.5 - 1) + 0.8,$$

where $i = 0 \dots k-1$ and α is the scaling factor selected in such a way that $\sum_i G_i = 1$.

To reduce the amount of noise, remove non-informative details from the image, and roughly separate the characters from the background, the morphological closure and erosion operators are sequentially applied.

Taking into account the specific characteristics of the font characters (in accordance with ISO/IEC 7811-7.1:2018), for the kernels of both filters, their structuring element

b of the size $n_b \times n_b$ ($n_b = \frac{w_{symb}}{3}$) is shaped as an ellipse.


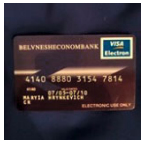

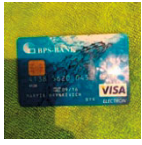
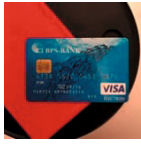
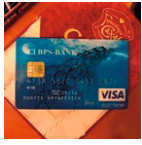
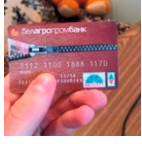

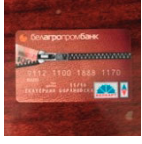
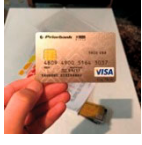
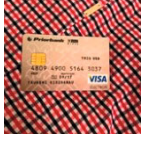
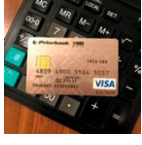
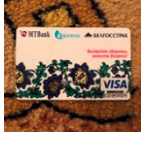
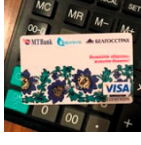
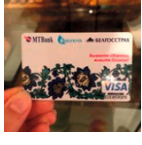

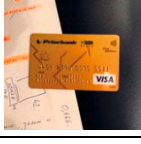

The thickness of a character w_{symb} does not depend on the type of a card image fragment to be processed and is set for OCR-B. The projection of the font width (in mm) onto the font size (in pixels) with respect to the size of the bank card region is a constant computed based on its overall width:

$$w_{symb} = 0.004884n_0. \quad (2.9)$$

Next, the edges of the character region on the processed image are refined by the vertical sliding window method. The height of the window depends on the font size of the characters; H_{cn} is the font height of the card number (4.0 mm), H_{ed} is the font height of the expiration date (2.85 mm), and H_{chn} is the font height of the cardholder's name (2.65 mm). Then, their height projections (in pixels) with respect to the size of the entire card region are h_{cn} , h_{ed} , and h_{chn} . The general formula is as follows:

$$h = \frac{Hm_0}{54.0}, \quad (2.10)$$

Table 1. Algorithm testing results

Frame from video sequence				Card is detected (number of frames)	Number of frames with correct recognition of all card details/total number of frames	Total processing time (ms)
1	2	3	No.			
			1	22	9/90	1217
			2	—	—/90	—
			3	26	10/90	1221
			1	23	10/90	1284
			2	26	11/90	1149
			3	31	13/90	1265
			1	24	—/90	—
			2	21	7/90	1344
			3	—	—/90	—
			1	30	18/120	852
			2	37	21/120	924
			3	36	26/120	892
			1	39	—/135	—
			2	44	—/135	—
			3	41	—/135	—
			1	34	24/90	1324
			2	35	28/90	1238
			3	32	25/90	1287

where 54.0 mm is the height of the card (ISO/IEC 7811-6.3:2018) and m_o is the height of the card region detected (px).

The character region extracted at the previous step, black and white lists of characters, and language identifier are sequentially transferred to the Tesseract recognition system.

A white list that consists of a set of numbers from 0 to 9 and a language identifier “eng” are passed as parameters of the card number image. For the expiration date image, the numbers from 0 to 9 and symbol “/” constitute a white list with “eng” being a language identifier. For the cardholder’s name, “rus” and “eng” are language identifiers and a black list comprises a set of characters that include punctuation marks, special symbols, and numbers. This approach speeds up character recognition by the Tesseract library, as well as the entire image processing.




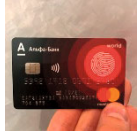
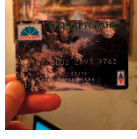
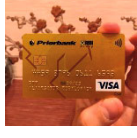
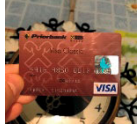
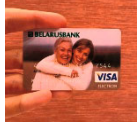
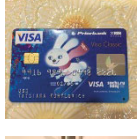

3. SOFTWARE IMPLEMENTATION AND EXPERIMENTAL RESULTS

3.1. Software Implementation

The algorithm is implemented in Objective-C by using the OpenCV low-level image processing library, Tesseract optical character recognition system, and iPhone SDK frameworks: CoreMedia and AVFoundation for media data management, UIKit for working with application interfaces, and CoreGraphics for low-level 2D image processing based on Quartz.

AVFoundation is used to capture the video stream from the main camera of an iOS device in real time at 30 fps. The frames are queued in separate threads by using NSOperationQueue class objects. The first thread receives frames and places them in a sequential processing queue if it is not full; otherwise, frames are ignored. The second thread sequentially receives frames from

Table 2. Comparison of the approaches for content recognition of bank cards

Initial data		CardIO			Our algorithm			
Card image	Embossing	Processing time for successful recognition	Result of information field recognition*		Processing time for successful recognition	Result of information field recognition*		
			card number	expiration date		card number	expiration date	cardholder
	Yes	0.343	1	1	1.018	1	1	1
	No	—	0	0	0.896	1	1	1
	Yes	0.314	1	1	1.059	1	1	0
	Yes	0.916	1	0	0	0	0	0
	No	—	0	0	1.01	1	1	1
	Yes	0.864	1	1	0.971	1	1	1
	Yes	0.48	1	1	0.861	1	1	0
	No	—	0	0	—	0	0	0
	Yes	0.478	1	0	0.957	1	1	1
	No	—	0	0	0.986	1	1	1

*1 corresponds to successful information field recognition and 0 indicates misrecognition.



Fig. 2. Interface of the mobile application that implements the proposed algorithm: (1) viewing area, (2) mark of successful card detection, and (3) output area for recognized data.

the capture queue and processes them until the results are obtained. Once processing is complete, the processed frame is deleted, the queue is emptied, and the processing iteration is initiated for the next frame.

The Tesseract library carries out asynchronous processing in the background. Once the algorithm terminates, the recognition result is passed (using a call-back function) to the user for displaying in the main thread. The interface of the mobile application with the recognition result is shown in Fig. 2.

3.2. Experimental Results

The performance of the proposed algorithm was tested using iPhone 7 on a set of 180 bank cards (see Fig. 3). These cards were placed against complex backgrounds; the information fields on the cards were clearly visible (not blocked by any foreign objects).

The values of the coefficients used in the computations were experimentally selected using ROC analysis [14]. The permissible deviation of the card's aspect ratio (e) was evaluated taking into account its exact size (in accordance with ISO/IEC 7811-5.4:2018) on a set of 214 images, 178 of which represented cards and 36 contained no cards or their boundaries were blurred or distorted. As a result of the experiment, for each coefficient e (which varied from 0 to 0.02 in increments of 0.0002), true positive rate (TPR) and false positive rate (FPR) were computed. For $e = 0.011$, the maximum TPR was 0.974194 and the minimum FPR was 0.111111.

For binarization, the size of the local region was determined taking into account the scaling factor, as

this parameter depends on the input size of the card region detected (its width W_l in pixels):

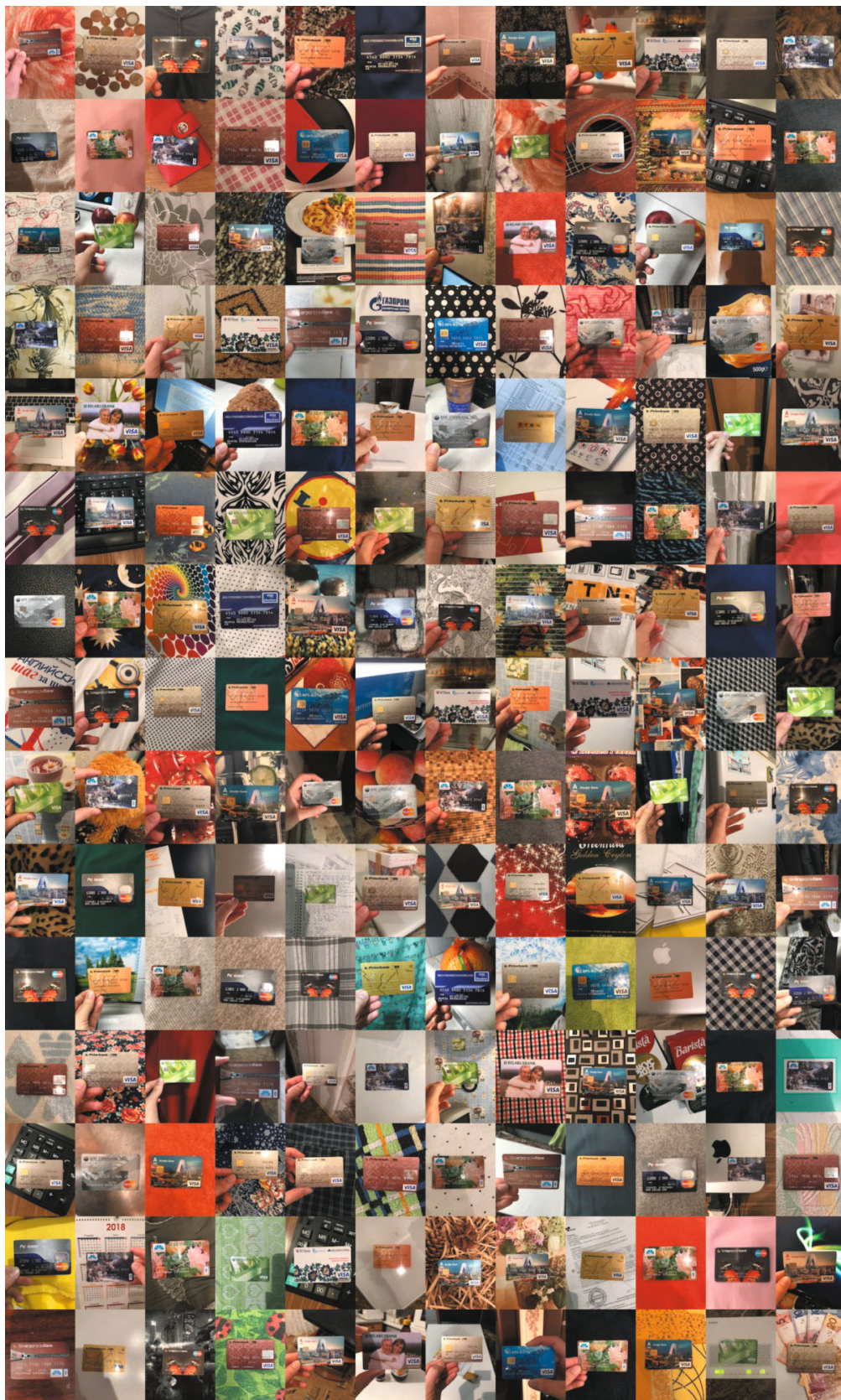
$$k = W_l k'. \quad (3.1)$$

The values of k' were varied from 0.002 to 0.03 in increments of 0.001. At this step, the maximum TPR was 0.83447 and the maximum FPR was 0.140864 for $k' = 0.016$. The constant $c = 13$ was determined similarly.

Table 1 shows some of the experimental results that demonstrate the efficiency of card data recognition on video sequences obtained under various conditions by using the proposed approach.

Upon setting the optimal coefficients and thresholds in the implementation of the algorithm, the ROC analysis shows TPR = 0.88 and FPR = 0.14. Figures 4–6 illustrate the detection of bank cards on a set of video sequences with indication of key frames and detection diagrams. On the diagram, 1 represents the successful detection of a card and 0 represents the absence of a card. The video was recorded at a constant rate of 30 fps.

The frame analysis of the video sequence shown in Fig. 4a suggests that the card image has strong luminance differences, shadows, and glares, which is why its stable detection (starting from the 71st frame, see Fig. 5a) is achieved when the user carries the card over to an area with more stable illumination. In another video sequence (see Fig. 4b), the card is not detected since its boundary does not entirely fall within the frame. Then, it is successfully detected (see Fig. 5b) but has a blurry contour and fuzzy characters; therefore, data recognition is not possible.



PROGRAMMING AND COMPUTER SOFTWARE Vol. 46 No. 6 2020

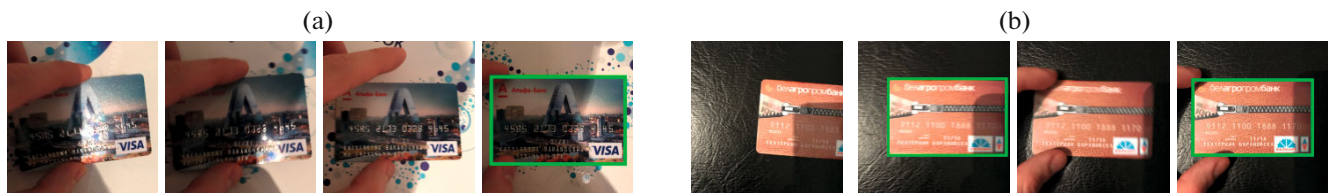


Fig. 4. Example of video sequence frames.

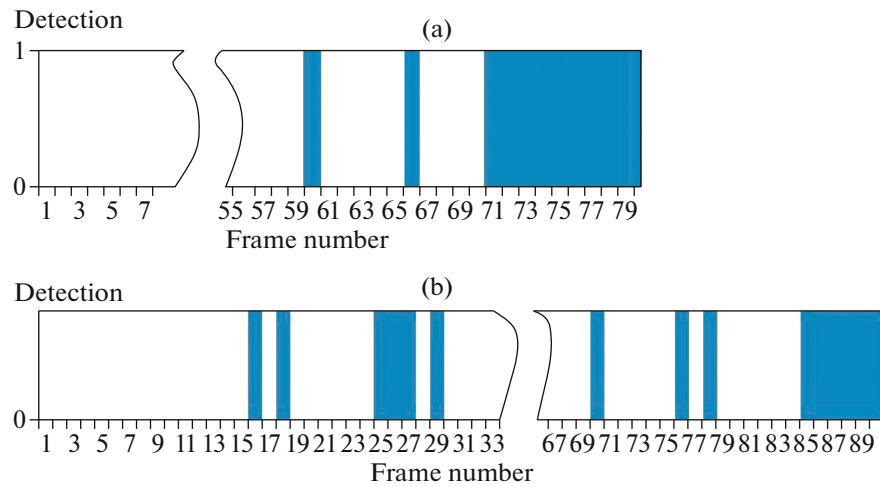


Fig. 5. Diagram of card detection on a video sequence: (a) for Fig. 4a and (b) for Fig. 4b.

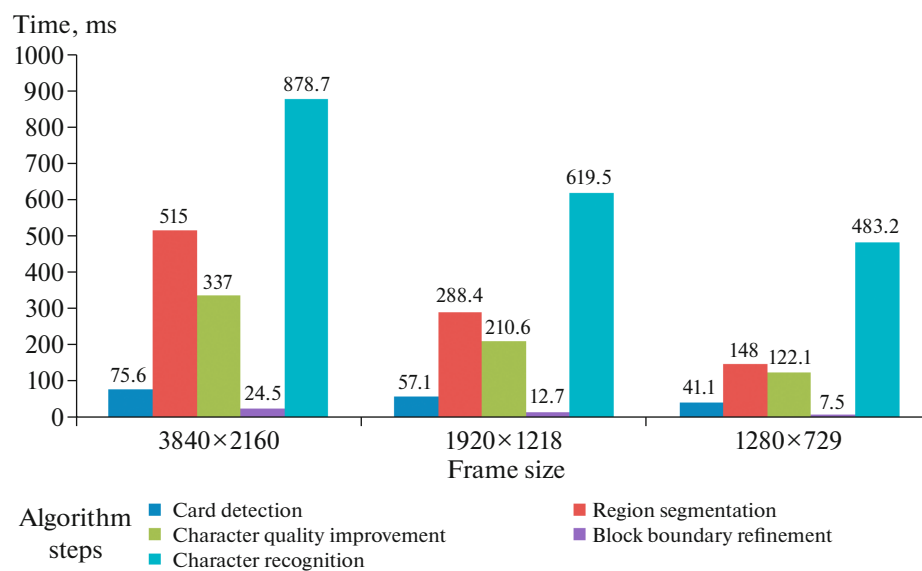


Fig. 6. Time cost distribution for the algorithm.

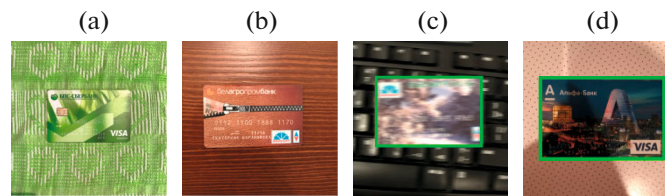


Fig. 7. Images of the cards that cannot be detected or the data on which cannot be recognized.

The time it takes to process one video frame at different resolutions by using iPhone 7 is shown in Fig. 6 for each step of the algorithm.

It can be seen (Fig. 6) that the time required for character recognition (in contrast to the other procedures) depends nonlinearly on the size of a frame to be processed.

3.3. Comparison and Discussion

The algorithms proposed in [7] and [8] have digit recognition accuracies of 80% and 90%, respectively. However, it should be noted that these results were obtained on different sets of images. To compare the proposed algorithm with some well-known approaches under the same conditions on the same image set, we used the CardIO SDK module [15].

The experiments were carried out under various video recording conditions for 180 cards, 104 of which were embossed ones. Table 2 shows the recognition results on 10 card images for the CardIO algorithm and our algorithm.

The analysis of the recognition results for all cards suggests that our approach significantly improves recognition on non-embossed cards as compared to CardIO, which does not support recognition of the cardholder's name. In general, our algorithm has TPR = 0.88 when recognizing all three information fields and TPR = 0.925 when recognizing the card number and expiration date. The CardIO module achieves TPR = 0.68 in numeric field classification. However, on the embossed cards only, CardIO has TPR = 0.935.

If the luminance and color characteristics of a bank card image input to the proposed algorithm are very similar to those of the background (see Figs. 7a and 7b), input image is blurry (see Fig. 7c), or its significant part is shadowed in such a way that the font color com-

pletely coincides with the background (see Fig. 10d), then the card cannot be detected (Figs. 7a and 7b) or the data on it cannot be recognized (Figs. 7c and 7d). In addition, the blurred, damaged, or defective contour of the characters, which creates no contrast between the card image and the background, makes data recognition impossible (see Fig. 8).

The analysis of the experimental results suggests that detection and recognition errors occur due to a number of factors: placement of a card against a complex background similar to the card itself, presence of glares or significant luminance differences, large distance between the card and the camera, and blurred card contours due to destabilization of the camera or card in the process of video recording. In the future, we intend to improve the proposed algorithm by improving the character recognition procedure.

4. CONCLUSIONS

Recognition of information fields on bank cards by using mobile devices is an important problem. In this paper, we have proposed an algorithm for content recognition of bank cards by using mobile devices. The efficiency of card detection and data recognition has been improved as compared to other approaches due to analyzing a sequence of images extracted from a video sequence. The proposed approach involves the following basic steps: detection of card boundaries in a frame based on the Viola–Jones algorithm and OverFeat method, segmentation of information fields taking into account their location on a card image, refinement of segmented images by histogram normalization and morphological processing, boundary detection of character blocks through adaptive binarization and morphological processing, their refinement using the vertical sliding window, and character recognition based on the Tesseract library. The software implementation of the proposed algorithm has been carried out using iPhone SDK, OpenCV, and Tesseract. The algorithm has been tested using iPhone 7 with the true positive rate being 0.88 for all three data fields and 0.925 for the card number and expiration date.

FUNDING

This work was supported by the Public Welfare Technology Applied Research Program of the Zhejiang Province

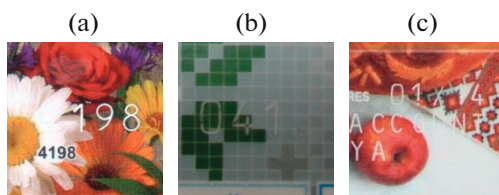


Fig. 8. Fragments of the cards with damaged characters.

(LGF19F020016, LGJ18F020001, and LGJ19F020002) and the National High-End Foreign Experts Program (GDW20183300463).

REFERENCES

1. Sheshkus, A., Nikolaev, D., Ingacheva, A., and Skoryukina, N., Approach to the recognition of flexible forms on the example of the credit card date recognition, *Proc. 8th Int. Conf. Machine Vision (ICMV)*, Barcelona, 2015, pp. 83–88.
2. Christian, T. and Gustavsson, D., Content recognition of business cards, IT Univ. Copenhagen, 2007.
3. Mollah, A., Basu, S., Das, N., Sarkar, R., Nasipuri, M., and Kundu, M., Text region extraction from business card images for mobile devices, *Proc. Int. Conf. Information Technology and Business Intelligence (ITBI)*, Nagpur, India, 2009, pp. 227–235.
4. Bhaskar, S., Lavassar, N., and Green, S., Implementing on the Android operating system for business cards, Stanford Univ., 2010, pp. 1–5.
5. Sharma, P. and Fujii, K., Automatic contact importer from business cards for Android, Stanford Univ., 2013.
6. Hua, G., Liu, Z., Zhang, Z., and Wu, Y., Automatic business card scanning with a camera, *Proc. Int. Conf. Image Processing*, Atlanta, 2006, pp. 373–376.
7. Cai, S., Wen, J., Xu, H., Chen, S., and Ming, Z., Bank card and ID card number recognition in Android financial APP, *Lect. Notes Comput. Sci.*, 2017, vol. 10135, pp. 205–213.
8. Liu, L., Huang, L., and Xue, J., Identification of serial number on bank card using recurrent neural network, *Proc. 9th Int. Conf. Graphic and Image Processing (IC-GIP)*, Qingdao, China, 2017, pp. 319–325.
9. Arlazarov, V.V., Bulatov, K.B., and Karpenko, S.M., Method for determining the reliability of embossed character recognition, *Tr. Inst. Sist. Anal. Ross. Akad. Nauk* (Proc. Inst. Syst. Anal. Russ. Acad. Sci.), 2013, vol. 63, no. 3, pp. 117–122.
10. Puybureau, E. and Geraud, T., Real-time document detection in smartphone videos, *Proc. 24th IEEE Int. Conf. Image Processing (ICIP)*, France, 2018, pp. 1498–1502.
11. Viola, P. and Jones, M., Rapid object detection using a boosted cascade of simple features, *Proc. IEEE Comput. Soc. Conf. Computer Vision and Pattern Recognition*, Kauai, USA, 2001, pp. 511–519.
12. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y., OverFeat: Integrated recognition, localization and detection using convolutional networks, *Proc. Int. Conf. Learning Representations (ICLR)*, Banff, Canada, 2014, pp. 1055–1061.
13. OpenCV documentation, Image filtering. <https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html#getgaussiankernel>.
14. Mason, S.J. and Graham, N.E., Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation, *Q. J. R. Meteorol. Soc.*, 2002, vol. 128, pp. 2145–2166.
15. CardIO official website. <https://www.card.io>.

Translated by Yu. Kornienko