esCam: A Mobile Application to Capture and Enhance Text Images

J. Pastor-Pellicer, Maria Jose Castro-Bleda(✉), and J.L. Adelantado-Torres

Departamento de Sistemas Informáticos y Computación,
Universitat Politècnica de València, Valencia, Spain
mcastro@dsic.upv.es

Abstract. Taking high resolution photos with mobile devices anytime anywhere is becoming increasingly common. Therefore, images of all kinds of text documents are recorded. This work presents esCam, an application for Android platform, whose goal is to preprocess the images of those text documents, in particular, perspective correction and image cleaning and enhancing. What truly differentiates our application is that esCam focuses on treatment of text that may appear in the image, using neural networks. These preprocessing steps are needed to make easier the digitalization and also to benefit subsequent steps such as document analysis and text recognition.

1 Introduction

Since the recent emergence of mobile platforms (like iOS, Android, Windows Phone, etc.), mobile terminals have experienced a huge increase in computational power and other features like screen and camera resolution. Furthermore, these devices include many tools that can be used anytime, anywhere. Therefore, we can take pictures of any type of documents: printed or manuscripts, ancient documents, or the most recent ones. All this without the need for specific devices such as scanners or high resolution cameras. However, taking snapshots with mobile devices, smartphones or even tablets, has disadvantages in comparison with other methods:

- Resolution: even though resolution of camera mobile phones is constantly increasing, these cameras still have lower resolutions than other specific image capture devices.
- Quality: brightness, shadows, noise and others, are examples of issues of snapshots taken by mobile devices, because they are not taken under ideal conditions.
- Perspective: when mobile devices are used to take snapshots, often we use our hands to hold the device, so the image is commonly taken from a bad perspective or unwanted areas appear within the snapshot.

In particular, when dealing with handwritten or printed documents, it is very important to be able to detect and extract the area of text with relevant information for the user, and to store it in a digital format accessible from any

I. Rojas et al. (Eds.): IWANN 2015, Part II, LNCS 9095, pp. 601–604, 2015.
DOI: 10.1007/978-3-319-19222-2_50

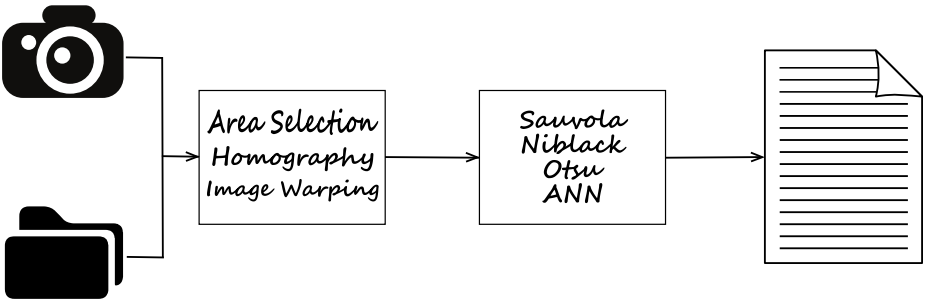

Fig. 1. esCam application workflow

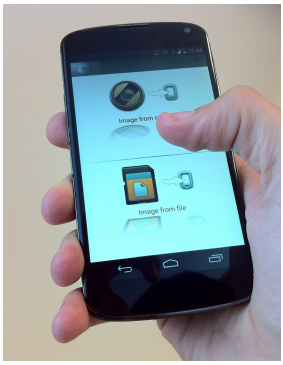

Fig. 2. esCam application menu

device [7]. It is also likely, specially when dealing with ancient documents, that artifacts such as noise, gloss, or perspective distortion appear [3]. These effects must be corrected before applying a character recognition system [2,6].

Despite the existence of similar applications in the market, none of these offers text enhancement filters like ours. The goal of this work is to design an application to use classical binarization algorithms like Otsu, Niblack, or Sauvola [8,10], as well as other more specific techniques like convolutional neural networks [4,9]. This demostration presents "esCam", the developed Android application to capture and enhance images of text documents in mobile devices, freely available at [1].

2 Application Description

Figure 1 shows a workflow diagram of the application and Figure 2 shows its initial menu. Figure 3 illustrates actual screenshots of the main phases of the application. It can be observed that, regardless of the image loading and saving steps, the image processing is divided in two main stages: perspective correction, and image cleaning and binarization.

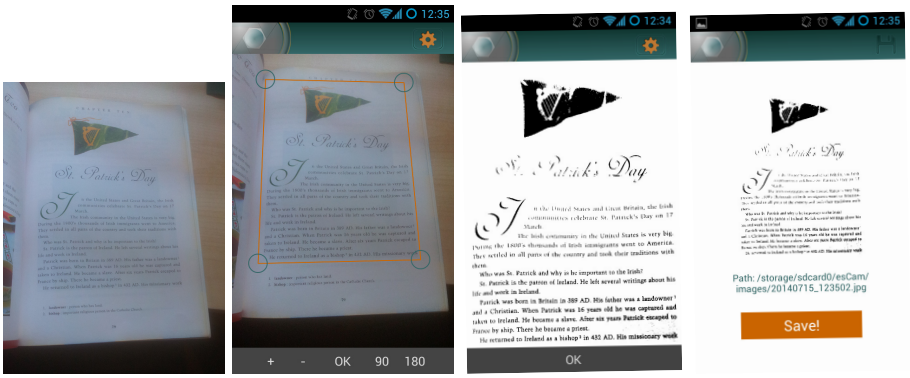

Fig. 3. Screenshots of esCam app (from left to right): Original image; Area selection; Normalized perspective image; Cleaned and enhanced image

Load and Save Images. Like other similar applications, the image can be taken by a camera or have been previously stored in memory. This is useful for the user, who can take a snapshot using the device camera for later image processing.

Perspective Correction. The perspective correction is a key step, especially when the images are taken with mobile cameras. Four coordinates suffice to correct perspective, that is, the four corners of the region that the user wants to focus on. In our application, the user can adjust these four points. Once the points are fixed, image warping correction is applied by means of homography. In addition, the user can correct the orientation of the image before image warping using proper rotations.

Filtering. The following preprocessing steps are image cleaning and binarization. When dealing with documents, particularly with text, it is desirable to have a high ratio between foreground text and background. Applying binarization filters have several advantages: first, it helps human reading and benefits the recognition for most OCR engines, and, secondly, it reduces the storage size of the image. Our application includes the following binarization filters:

- Niblack/Sauvola filtering [10] calculates a threshold using a sliding window in a greyscaled version of the image. Threshold calculation is based on the local mean and standard deviation of that window.
- Otsu method [8] is based in the idea of separate background and object. It uses statistical methods as the variance among grey tones in the image.
- A Neural Networks technique presented in [2] is also included: a multilayer perceptron returns a clean and enhanced pixel, given an input of this pixel plus a context. The neural network has been trained with the corpus Noisy-Office [11], freely available at UCI Machine Learning Repository [5], http://archive.ics.uci.edu/ml/datasets/NoisyOffice.

3 Conclusions

The esCam application is presented in this demostration.[1] In addition to the Android software developing kit intended for Android apps, we used OpenCV as a library for efficient image processing. The convolutional neuronal networks are trained using the APRIL-ANN toolkit [12]. The application is devoted to the preprocessing of text images, especially images taken with mobile devices, adding tools for perspective correction and image filtering. The preprocessed images achieve a better information representation for further steps such as line extraction or text recognition.

Acknowledgments. This work has been partially supported by the Spanish Government project TIN2010-18958.

References

1. http://www.dsic.upv.es/mcastro/esCam/index_en.htm
2. España Boquera, S., Castro-Bleda, M.J., Gorbe-Moya, J., Zamora-Martinez, F.: Improving offline handwritten text recognition with hybrid HMM/ANN models. IEEE Trans. PAMI 33(4), 767–779 (2011)
3. Gatos, B., Pratikakis, I., Perantonis, S.J.: Adaptive degraded document image binarization. Pattern Recognition **39**(3), 317–327 (2006)
4. Hidalgo, J.L., España, S., Castro, M.J., Pérez, J.A.: Enhancement and Cleaning of Handwritten Data by Using Neural Networks. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3522, pp. 376–383. Springer, Heidelberg (2005)
5. Lichman, M.: UCI ML repository (2013). http://archive.ics.uci.edu/ml
6. Mori, S., Suen, C.Y., Yamamoto, K.: Historical review of OCR research and development. Proceedings of the IEEE **80**(7), 1029–1058 (1992)
7. Nagy, G.: Twenty Years of Document Image Analysis in PAMI. IEEE Trans. PAMI **22**(1), 38–62 (2000)
8. Otsu, N.: A threshold selection method from gray-level histograms. Automatica **11**(285–296), 23–27 (1975)
9. Pastor-Pellicer, J., Zamora-Martínez, F., España-Boquera, S., Castro-Bleda, M.J.: F-Measure as the Error Function to Train Neural Networks. In: Rojas, I., Joya, G., Gabestany, J. (eds.) IWANN 2013, Part I. LNCS, vol. 7902, pp. 376–384. Springer, Heidelberg (2013)
10. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recognition **33**(2), 225–236 (2000)
11. Zamora-Martínez, F., España-Boquera, S., Castro-Bleda, M.J.: Behaviour-Based Clustering of Neural Networks Applied to Document Enhancement. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 144–151. Springer, Heidelberg (2007)
12. Zamora-Martínez, F., España-Boquera, S., Gorbe-Moya, J., Pastor-Pellicer, J., Palacios, A.: APRIL-ANN toolkit, A Pattern Recognizer In Lua with Artificial Neural Networks (2013). https://github.com/pakozm/april-ann

[1] Freely available at http://www.dsic.upv.es/~mcastro/esCam/index_en.html, along with a tutorial and a demo video.