February 2021

UNWARPING PAGES USING SEMANTIC SEGMENTATION INFORMATION AND DENSE IMAGE WARPING MATRIX

HP INC

Title

Unwarping pages using semantic segmentation information and dense image warping matrix

Abstract

Scanning pages from books or magazines usually result in a digital image with warped regions, which impacts the text quality and decreases OCR precision. To address this problem, this work proposes a method using machine learning to estimate an unwarping matrix that can fix the warping regions of the scanned page. Furthermore, this method allows the user interaction with the final document, making it possible to adjust the level of unwarping adjustment.

Introduction

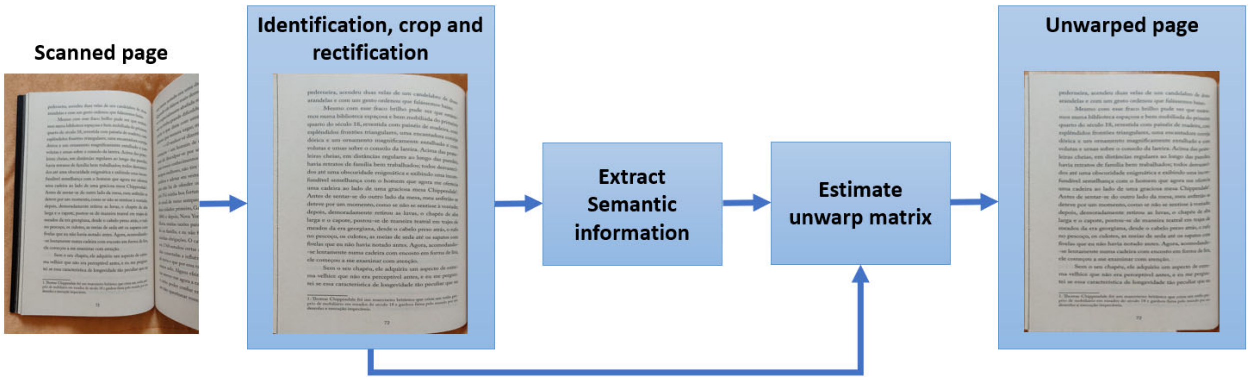
Scanning pages from books or magazines usually result in a picture with part of the text being warped. This is caused due the pages attachments in the center of the book or magazine resulting in this effect even on pages scanned using some dedicated hardware, such as a flatbed scanner. However, while hardware scanner users can overcome part of this problem by applying pressure to the page positioned on the scanner tray while scanning, users of mobile scanner applications cannot rely on the same solution. For this last scenario, only some post-processing algorithm could adjust the warped regions of the returned page image.

Hence, unwarping a page means re-establishing the original content alignment of the scanned page. This is crucial for OCR applications because most of its engines expects some pre-defined text alignment. Besides that, the regions near to the book or magazine center, that usually suffers with more intense warping, result in a bad resolution text for the user itself.

Proposed solution

A high-level overview of the proposed solution is illustrated in Figure 1. Below is the description for each of the steps taken by the solution.

Figure 1. Proposed workflow for unwarping scanned pages.



1. Identify and crop the page from the scene. In the first step of the solution it is necessary to identify the page in the captured scene. This is done by identifying the four extreme points (most left, most right, most top and most bottom) of the page. With these four extreme points it is possible to extract a box that encloses the page.

2. Rectify the cropped page. To have a better view of the cropped page, the four corner points are identified and used to rectify the page image. This process corrects the page perspective, adjusting, for example, possible rotations of the page regarding the box that was used to extract it.

3. Extract semantic segmentation information. This step is intended so the next algorithm can better understand the displacement of the page in the scene, because, even with the crop and page rectification, images with warped pages still contain some background information. Then, it is necessary to extract features that retain some semantic information to allow the next algorithm to discriminate between page content and background content.

4. Estimate the pixel-wise unwarping matrix. For this step, both the cropped-rectified image and the semantic information are used to estimate a pixel-wise matrix that is used to unwarp the page. This matrix is used as a nonlinear function to remap the pixels of the input image.

5. Apply the estimated pixel-wise unwarping matrix. Given an input image **I** with shape (H, W, C), where H is the image height, W is its width and C is the number of channels (usually 3), the output unwarp matrix **M** will have shape (H, W, 2) and the expression that remaps each pixel in **I** to the corresponding unwarped image **D** is presented in Equation 1.

Equation 1. $$\mathbf{D}(j,i,c) = \mathbf{I}(j - \mathbf{M}(j,i,0), i - \mathbf{M}(j,i,1), c)$$

Notice that the locations specified by this formula do not necessarily map to an integer index. Therefore, the pixel values need to be interpolated using some of the nearest pixels [2]. Such examples of interpolation could be nearest neighbors or bilinear.

At this step, it is also possible to multiply the matrix **M** by some value **a** so the user could have some control on the intensity of the unwarping. For instance, if the estimated matrix **M** has very low values, usually the final unwarping will be very smooth. By multiplying this estimated matrix by some value **a**>0 the user would be capable of controlling the effects of the unwarping in order to improve the final result.

Proof of Concept

Herein it is described the methodology used to design the neural network capable of estimating the unwarping matrix, since the steps **1** and **2** are already well described in the current literature [3,4,5] and can be achieved in many ways.

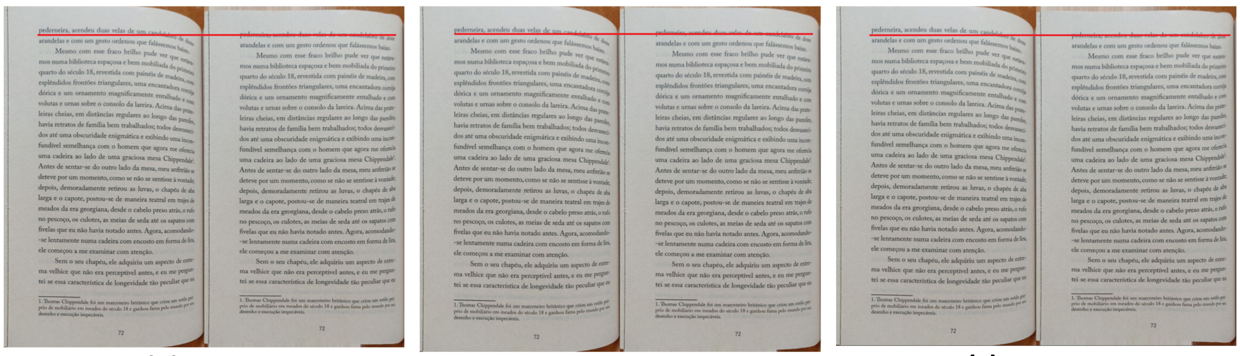
The used neural network architecture has its first branch (described in step **3**) inspired by the work of Zhang, Song-Hai, et al on the PortraitNet architecture [6]. This model was proposed to extract segmentation masks from portrait pictures and to be used on real time edge applications. The PortraitNet receives as input a RGB image and outputs a segmentation mask and a contour mask with same size as the input image. Before those output layers, it is possible to modify this model to return a feature map with the same size as the input image.

So, for this solution, it was removed the contour output; and the returned segmentation mask was used only during the training of the neural network. Then, the final extracted feature map was concatenated to the rectified image page to be fed into the next step of the complete model.

The following branch of the neural networks, regarding step **4**, used an architecture inspired by the work of Tan, Mingxing, and Quoc V. Le on the EfficientNets [7]. It could be used any of the proposed architectures (B0 to B7), but for the purpose of deploying it on the edge, it was used the B0 - the smallest one. It was also made a modification on this architecture to return the 6th block of the neural network (downsizing the original model size). By doing this, the estimated unwarp matrix has size 32 times smaller than the input image. This was done so the returned unwarp matrix had less possible variations on its values and to help network training convergence. After that, it is used the bilinear interpolation so the unwarp matrix and the input image have the same size.

Equation 1 presented in step **5** can now be applied directly on the rectified image using the estimated (and resized) unwarp matrix. However, before that, it is also possible to adjust the variable **a** that multiplies the unwarp matrix **M** and, hence, controlling the unwarping intensity. Figure 2 presents some results of this proposed solution varying the **a** value.

Figure 2. Results from the POC by varying the **a** value. From left to right: the input rectified and the unwarped page. The red line was placed to compare the text alignment.



(a) a = 1 **(b) a** = 0,64 **(c) a** = 1,3

References

[1] Fan, J. (2007). “Enhancement of camera-captured document images with watershed segmentation”. CBDAR07, 87-93.

[2] Tensorflow. “Dense image warp”. Available at: https://www.tensorflow.org/addons/api_docs/python/tfa/image/dense_image_warp

[3] Ying Xiong (2016). “Fast and Accurate Document Detection for Scanning”. Available at: https://dropbox.tech/machine-learning/fast-and-accurate-document-detection-for-scanning

[4] Shakleen Ishfar (2020). “Document Detection in Python”. Available at: https://medium.com/intelligentmachines/document-detection-in-python-2f9ffd26bf65

[5] Adrian Rosebrock (2014). “How to Build a Kick-Ass Mobile Document Scanner in Just 5 Minutes”. Available at: https://www.pyimagesearch.com/2014/09/01/build-kick-ass-mobile-document-scanner-just-5-minutes/

[6] Zhang, Song-Hai, et al. "PortraitNet: Real-time portrait segmentation network for mobile device." Computers & Graphics 80 (2019): 104-113.

[7] Tan, Mingxing, and Quoc V. Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." arXiv preprint arXiv:1905.11946 (2019).

- *Disclosed by Lucas Nedel Kirsten, HP Inc.*