AN OPEN-SOURCE APPROACH TO PREPARE SCANNED IMAGES FOR OCR

Tiberiu BUJOR

Abstract

The progress made by the big companies in the artificial intelligence domain is often shared with the public as open source or by allowing free use of software programs over the internet. Google drive OCR removes the burden of having to obtain data needed to feed a classifier by using its huge dataset. The greatest results are always obtained when the correct information is provided and, using Google's OCR is no different because image repair is a resource costly process and should be done separately if one is to obtain the best results. This paper describes methods and algorithms developed for document image enhancement and retrieval that could be used before the optical character recognition process.

1. INTRODUCTION

Document images usually suffer from several defects mainly due to natural ageing, environmental conditions, usage, poor storage conditions of the original, as well as human manipulations during the scanning process. Since document image analysis and recognition methods usually assume a smooth background and a good quality of printing or writing, it is imperative to have an efficient image enhancement process in order to restore the good quality of document images. Moreover, document image enhancement process enhances the readability of text areas and permits less image storage space.

Google Drive’s Optical Character Recognition (OCR) lets you convert PDF files with images with text into text documents with support for over 200 different languages. The major disadvantage is that it’s limited when it comes to resolving problems mentioned previously, that are obtained from low contrast and uneven background illumination, bleed-through, shining, or shadow-through effects, damaged characters or noisy background, borders or parts of neighboring page, etc.

This paper is organized as follows: section 2 provides the general techniques for document image enhancement and normalization, section 3 will review the most popular and current algorithms used in document image enhancement and normalization applications, and section 4 concludes the paper.

2. ENHANCEMENT TECHNIQUES

Image enhancement can be easily described as transforming an image f into image g using T. (where T represents the transformation). The main categories in which these techniques fall are either of the Spatial domain (pixel oriented) or based on a Frequency domain (Fourier transform).

One major issue in image processing is *contrast enhancement*, to solve the low contrast and uneven background illumination of an image, one must focus on enhancing the luminosity between background and text areas, smoothing of background texture, as well as the elimination of noisy areas.

Some popular methods in this manner are:

- Histogram equalization (most simple, and effective)
- Otsu's thresholding [1]
- Kapur's entropy [2]
- Kittler and Illingworth minimal error [3]
- Leung generalized fuzzy operator preprocessed with histogram equalization and partially overlapped sub-block histogram equalization (achieves good performance when working with extremely low contrast and low illuminated document images) [4]
- Nomura method in which morphological operations are used to remove undesirable shapes called critical shadows on the background of document images before proceeding to binarization [5]

One other major issue consists of the *Bleed-Through, Shining, and Shadow-Through Effects* that can be categorized into either non-registration (or blind) or registration (or non-blind) methods, depending on whether the verso image of a page is available and precisely registered to the recto image.

a) When the two sides of a page are treated independently and processed separately (*non-registration-based methods*). Those methods attempt to clean the front side of a document without referring to the reverse side. Most of these methods treat bleed-through interference as a kind of background artifacts or noise and remove it using threshold-like techniques.

b) When the verso image of a page is available and precisely registered to the recto image (*registration-based methods*). In order to register the verso with the recto image, several techniques have been proposed based on:

(i) intensity characteristics, intensity patterns are compared using correlation metrics;

(ii) feature characteristics, correspondence between features such as points, lines, and contours is searched;
(iii) transformation models;
(iv) spatial and frequency domain.

When character areas are not visible or there is noise in the background (*noise reduction method*) common operations are applied in order to enhance the quality of a binary document image; these include masks, connected component analysis, morphological operations, as well as shrink and swell operations.

Enhancement of grayscale or color images has proved to be efficient by application of Ajayi's Partial Diffusion Equation (PDE) for enhancing text in degraded document images. His algorithm is based on a lookup table classification algorithm that learns the corrections of patterns of text degradation in document images.[11]

When borders or parts of adjacent pages are visible in the current image, methods that can be applied are Connected Component Analysis-Based Methods, Projection Profile-Based Methods and "Flood-Fill" Algorithms.

When document digitization is done with flatbed scanners or camera-based systems, the resulting images can often suffer from skew, warping, and perspective distortions. The scanned text image may happen to be rotated by half or even be upside down. A document image normalization step is imperative in order to restore text areas horizontally aligned without any distortions as well as in a straight angle.

Portrait/landscape orientation detection is mainly accomplished by using projection histograms as well as by counting black-to-white transitions.

When the skew angle ranges between -10° to 10° we can apply deskew and deslant methods which fall into the following categories:

a) Projection Profile-Based Skew: works by calculating a series of horizontal projection profiles as we rotate the document page at a range of angles. The optimization of an objective function for a given skew angle leads to the actual document skew.
b) *Hough Transform-Based Skew:* Each black pixel is mapped to the Hough space (*p*, *0*) and the skew is estimated as the angle in the parameter space that gives the maximum sum of squares of the gradient along the *p* component.
c) *Nearest-Neighbor Clustering-Based***:** spatial relationships and mutual distances of connected components are used to estimate the page skew. The direction vector of all nearest-neighbor pairs of connected components is accumulated in a histogram and the peak in the histogram gives the dominant skew.
d) *Cross-Correlation:* are based on measuring vertical deviations among foreground pixels along the document image in order to detect the page skew.

Using a flatbed scanner or a digital camera also results in several unavoidable image distortions due to the form of printed material, the camera setup, or environmental conditions (page erosion).

Many different techniques for dewarping can be classified into two main categories based on 3-D document shape reconstruction and 2-D document image processing.

a) Extraction of the 3-D information of the document and then models the page surface by curved developable surfaces to estimate the 3-D shape of the page using texture flow fields.

b) Detection of distorted text lines at the original document image which is a well known hard task. Some of these techniques propose a method to straighten distorted text lines by fitting a model to each text line.

3. ENHANCEMENT ALGORITHMS

3.1 Detect and fix skew in images containing text [6]

Methods used: *Canny Edge Detection, Hough Transform*

How it works:

- Converts the image to grayscale
- Performs Canny Edge Detection on the Image
- Calculates the Hough Transform values
- Determines the peaks
- Determines the deviation of each peaks from 45 degree angle
- Segregates the detected peaks into bins
- Chooses the probable skew angle using the value in the bins

Installation:

1. # mkdir deskew-test
2. # cd deskew test
3. # mkvirtualenv .
4. # pip install alyn
5. # source bin/activate

Technical specifications:

- Year: 2017
- Programming Language: Python
- License: MIT
- Author: Kakul Chandra

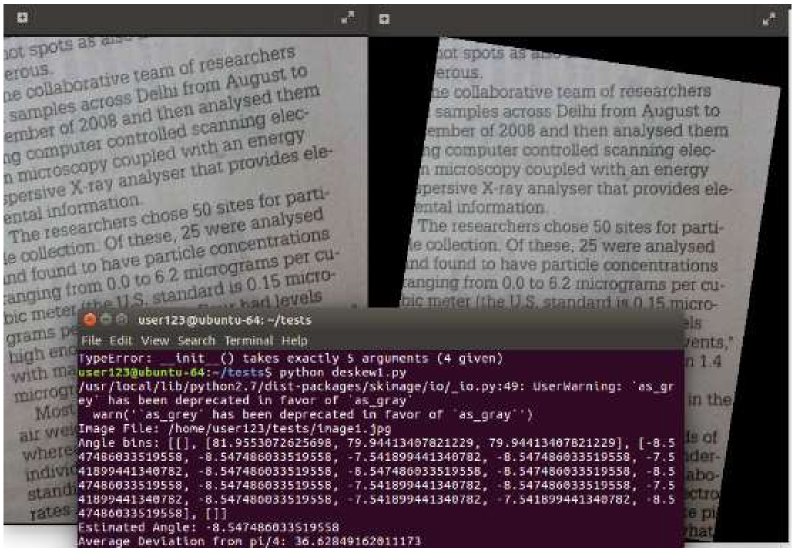
Requires:

- numpy
- matplotlib
- scipy
- scikit-image

Usage:

```
from alyn import Deskew
d = Deskew(
input_file='path_to_file', display_image='preview the image on screen',
output_file='path_for_deskewed_image',
r_angle='offest_angle_in_degrees_to_control_orientation')
d.run()
```

```
./skew_detect.py -i image.jpg
```


user123@ubuntu-64: ~/tests
File Edit View Search Terminal Help

```
TypeError: __init__() takes exactly 5 arguments (4 given)
user123@ubuntu-64:~/tests$ python deskew1.py
/usr/local/lib/python2.7/dist-packages/skimage/io/_io.py:49: UserWarning: `as_gr
ey` has been deprecated in favor of `as_gray`
  warn('`as_grey` has been deprecated in favor of `as_gray`')
Image File: /home/user123/tests/image1.jpg
Angle bins: [[], [81.9553072625698, 79.94413407821229, 79.94413407821229], [-8.5
47486033519558, -8.547486033519558, -7.541899441340782, -8.547486033519558, -7.5
41899441340782, -8.547486033519558, -8.547486033519558, -8.547486033519558, -8.5
47486033519558, -8.547486033519558, -7.541899441340782, -8.547486033519558, -7.5
41899441340782, -8.547486033519558, -7.541899441340782, -7.541899441340782, -8.5
47486033519558], []]
Estimated Angle: -8.547486033519558
Average Deviation from pi/4: 36.62849162011173
```

Before (left) and after (right) of test run with angle < 5 degrees [Fig. 1]

3.2 Contrast Enhancement for low-light images [7]

Methods used: *Exposure Fusion Network*

How it works:

- design the weight matrix for image fusion using illumination estimation techniques
- use camera response model to synthesize multi-exposure images
- find the best exposure ratio so that the synthetic image is well-exposed in the regions where the original image was under-exposed
- input image and the synthetic image are fused according to the weight matrix to obtain the enhancement result

Technical specifications:

○ Year:	Sep 24, 2018
○ Programming Language:	Python
○ License:	Open
○ Author:	Andy Huang

Installation:

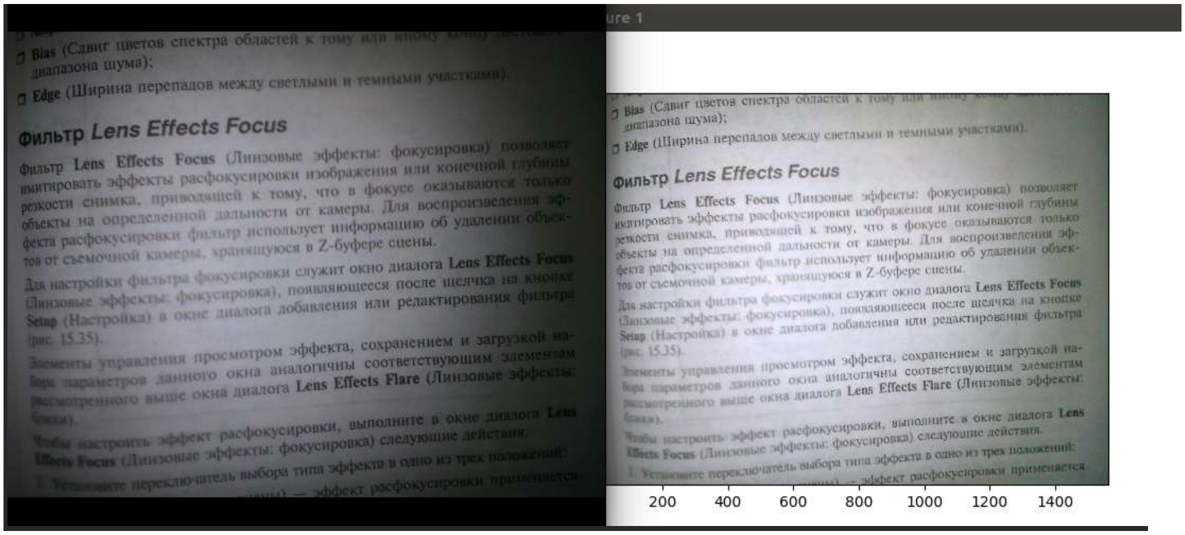
```
# mkdir enhancement-test
# cd enhancement-test
# git clone https://github.com/AndyHuang1995/Image-Contrast-Enhancement
```

Requires:

- ○ numpy
- ○ scipy
- ○ scikit-image
- ○ cv2
- ○ imageio
- ○ matplotlib
- ○ git
- ○ skimage

Usage:

```
# python ying.py <input image>
```



□ **Bias** (Сдвиг цветов спектра областей к тому или иному концу цветового диапазона шума);

□ **Edge** (Ширина перепадов между светлыми и темными участками).

Фильтр *Lens Effects Focus*

Фильтр **Lens Effects Focus** (Линзовые эффекты: фокусировка) позволяет имитировать эффекты расфокусировки изображения или конечной глубины резкости снимка, приводящей к тому, что в фокусе оказываются только объекты на определенной дальности от камеры. Для воспроизведения эффекта расфокусировки фильтр использует информацию об удалении объектов от съемочной камеры, хранящуюся в Z-буфере сцены.

Для настройки фильтра фокусировки служит окно диалога **Lens Effects Focus** (Линзовые эффекты: фокусировка), появляющееся после щелчка на кнопке **Setup** (Настройка) в окне диалога добавления или редактирования фильтра (рис. 15.35).

Элементы управления просмотром эффекта, сохранением и загрузкой набора параметров данного окна аналогичны соответствующим элементам рассмотренного выше окна диалога **Lens Effects Flare** (Линзовые эффекты: блики).

Чтобы настроить эффект расфокусировки, выполните в окне диалога **Lens Effects Focus** (Линзовые эффекты: фокусировка) следующие действия.

1. Установите переключатель выбора типа эффекта в одно из трех положений:

— эффект расфокусировки применяется

Before (left) and after (right) of contrast enhancement algorithm with no input parameters [Fig. 2]

3.3 Image Shadow Detection and Removal [8]

Methods used: *Processing in HSV color space, histogram matching*

How it works:

- take the derivatives of the original image
- apply a mean shift segmentation
- setting shadow edges derivatives to 0 and reintegrate the image by:
  - solving a Poisson equation, 2-dimensional integration
  - generating random Hamiltonian paths, 3-dimensional integration

Technical specifications:

- Year: Nov 21, 2017
- Programming Language: Matlab
- License: Open
- Author: Jiarui Gao

Requires:

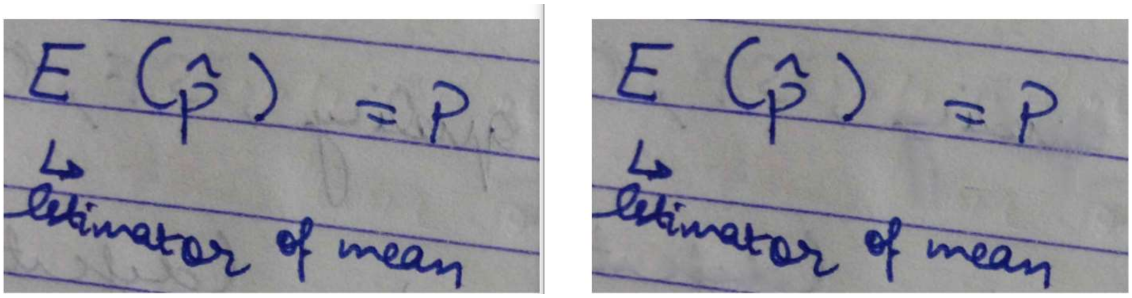
- Git
- Matlab

Installation:

```
# install matlab & plugins

# mkdir shadow_remove-test

# cd shadow_removetest

# git clone https://github.com/kittenish/Image-Shadow-Detection-and-Removal
```

Usage:

```
# open main.m in matlab and select the image
```



Before (left) and after (right) of the shadow remove algorithm on a selected image. [Fig. 3]

3.4 Denoising [9]

Methods used: *Despeckle and Enhance*

How it works:

- consider each pixel in the image
- sort the neighboring pixels into order based upon their intensities
- replace the original value of the pixel with the median value from the list

Technical specifications:

- Year: December 15, 2018
- Programming Language: Bash
- License: For personal use only
- Author: Fred Weinhaus

Requires:

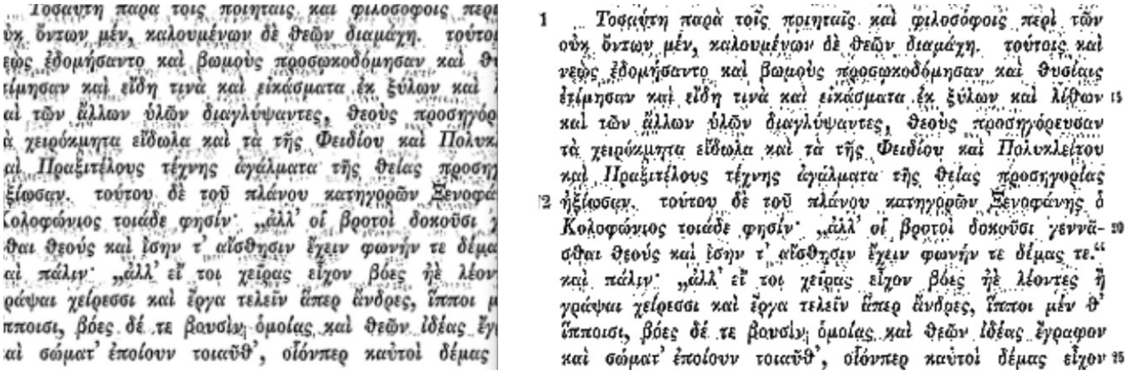
- Git
- Matlab

Installation:

```
# mkdir noisecleaner-test
# cd noisecleaner-test
#wget
http://www.fmwconcepts.com/imagemagick/downloadcounter.php?sc
riptname=noisecleaner&
dirname=noisecleaner
# chmod +x *
```

Usage:

```
# ./noisecleaner-m 2 -n 10 -f all image.png image2.png
```



… Τοσαυτη παρα τοις ποιηταις και φιλοσοφοις περι
ὐκ ὄντων μέν, καλουμένων δὲ θεῶν διαμάχη. τούτοι
εὼς ἐδομήσαντο καὶ βωμοὺς προσῳκοδόμησαν καὶ θ
ιίμησαν καὶ εἴδη τινὰ καὶ εἰκάσματα ἐκ ξύλων καὶ
αὶ τῶν ἄλλων ὑλῶν διαγλύψαντες, θεοὺς προσηγόρ
ὰ χειρόκμητα εἴδωλα καὶ τὰ τῆς Φειδίου καὶ Πολυκ
αὶ Πραξιτέλους τέχνης ἀγάλματα τῆς θείας προση
ξίωσαν. τούτου δὲ τοῦ πλάνου κατηγορῶν Ξενοφά
Κολοφώνιος τοιάδε φησίν· „ἀλλ' οἱ βροτοὶ δοκοῦσι γ
θαι θεούς καὶ ἴσην τ' αἴσθησιν ἔχειν φωνήν τε δέμα
αὶ πάλιν· „ἀλλ' εἴ τοι χεῖρας εἶχον βόες ἠὲ λέον
ράψαι χείρεσσι καὶ ἔργα τελεῖν ἅπερ ἄνδρες, ἵπποι μ
ιπποισι, βόες δέ τε βουσίν ὁμοίας καὶ θεῶν ἰδέας ἔγ
αὶ σώματ' ἐποίουν τοιαῦθ', οἷόνπερ καὐτοὶ δέμας

1 … Τοσαύτη παρὰ τοῖς ποιηταῖς καὶ φιλοσόφοις περὶ τῶν
οὐκ ὄντων μέν, καλουμένων δὲ θεῶν διαμάχη. τούτοις καὶ
νεὼς ἐδομήσαντο καὶ βωμοὺς προσῳκοδόμησαν καὶ θυσίαις
ἐτίμησαν καὶ εἴδη τινὰ καὶ εἰκάσματα ἐκ ξύλων καὶ λίθων
καὶ τῶν ἄλλων ὑλῶν διαγλύψαντες, θεοὺς προσηγόρευσαν
τὰ χειρόκμητα εἴδωλα καὶ τὰ τῆς Φειδίου καὶ Πολυκλείτου
καὶ Πραξιτέλους τέχνης ἀγάλματα τῆς θείας προσηγορίας
2 ἠξίωσαν. τούτου δὲ τοῦ πλάνου κατηγορῶν Ξενοφάνης ὁ
Κολοφώνιος τοιάδε φησίν· „ἀλλ' οἱ βροτοὶ δοκοῦσι γεννᾶ-
σθαι θεούς καὶ ἴσην τ' αἴσθησιν ἔχειν φωνήν τε δέμας τε."
καὶ πάλιν· „ἀλλ' εἴ τοι χεῖρας εἶχον βόες ἠὲ λέοντες ἢ
γράψαι χείρεσσι καὶ ἔργα τελεῖν ἅπερ ἄνδρες, ἵπποι μέν θ'
ἵπποισι, βόες δέ τε βουσὶν ὁμοίας καὶ θεῶν ἰδέας ἔγραφον
καὶ σώματ' ἐποίουν τοιαῦθ', οἷόνπερ καὐτοὶ δέμας εἶχον

Before (left) and after (right) of the noise removal algorithm on a text image. [Fig. 4]

3.5 Dewarping [10]

Methods used: *Split the text into lines and find a warp or coordinate transformation that makes the lines parallel and horizontal*

How it works:

- Obtain page boundaries
- Detect text contours
- Assemble text into spans
- Sample spans
- Create naive parameter estimate
- Remap image and threshold

Technical specifications:

- Year: Oct 2, 2016
- Programming Language: Python
- License: MIT
- Author: Matt Zucker

Requires:

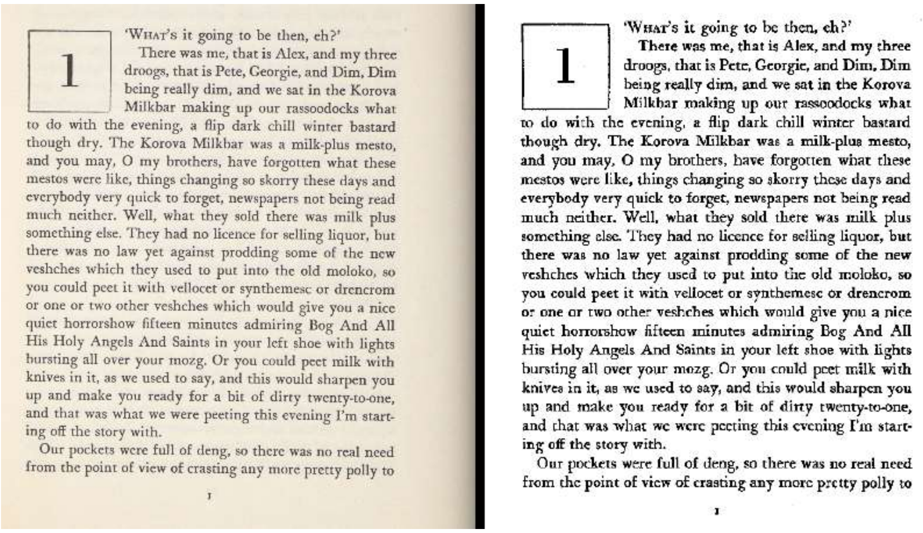
- scipy
- opencv3
- pillow
- git

Installation:

```
# mkdir dewarp-test
# cd dewarp-test
# git clone https://github.com/mzucker/page_dewarp
```

Usage:

```
# python python page_dewarp.py image1.jpg
```



1

'WHAT's it going to be then, eh?'

There was me, that is Alex, and my three droogs, that is Pete, Georgie, and Dim, Dim being really dim, and we sat in the Korova Milkbar making up our rassoodocks what to do with the evening, a flip dark chill winter bastard though dry. The Korova Milkbar was a milk-plus mesto, and you may, O my brothers, have forgotten what these mestos were like, things changing so skorry these days and everybody very quick to forget, newspapers not being read much neither. Well, what they sold there was milk plus something else. They had no licence for selling liquor, but there was no law yet against prodding some of the new veshches which they used to put into the old moloko, so you could peet it with vellocet or synthemesc or drencrom or one or two other veshches which would give you a nice quiet horrorshow fifteen minutes admiring Bog And All His Holy Angels And Saints in your left shoe with lights bursting all over your mozg. Or you could peet milk with knives in it, as we used to say, and this would sharpen you up and make you ready for a bit of dirty twenty-to-one, and that was what we were peeting this evening I'm starting off the story with.

Our pockets were full of deng, so there was no real need from the point of view of crasting any more pretty polly to

1

1

'WHAT's it going to be then, eh?'

There was me, that is Alex, and my three droogs, that is Pete, Georgie, and Dim, Dim being really dim, and we sat in the Korova Milkbar making up our rassoodocks what to do with the evening, a flip dark chill winter bastard though dry. The Korova Milkbar was a milk-plus mesto, and you may, O my brothers, have forgotten what these mestos were like, things changing so skorry these days and everybody very quick to forget, newspapers not being read much neither. Well, what they sold there was milk plus something else. They had no licence for selling liquor, but there was no law yet against prodding some of the new veshches which they used to put into the old moloko, so you could peet it with vellocet or synthemesc or drencrom or one or two other veshches which would give you a nice quiet horrorshow fifteen minutes admiring Bog And All His Holy Angels And Saints in your left shoe with lights bursting all over your mozg. Or you could peet milk with knives in it, as we used to say, and this would sharpen you up and make you ready for a bit of dirty twenty-to-one, and that was what we were peeting this evening I'm starting off the story with.

Our pockets were full of deng, so there was no real need from the point of view of crasting any more pretty polly to

1

Before (left) and after (right) of the dewarp algorithm on a book page image. [Fig. 5]

4. CONCLUSION & FUTURE IMPROVEMENTS

The discussed techniques are more than likely to have an open source implementation that is relatively easy to find and can be used with little to no configuration required. The highlighted algorithms performed better than expected and could be integrated in a automated tool (preparator) that would further serve them to the OCR. Another approach would be to look at the source code of a bigger tool that has all these features already integrated (GIMP) and use the code in a specific implementation targeted to only fix scanned text images.

Installing the necessary libraries for running these algorithms is a very straight forward process that should not require a lot of time for an experienced user. I used an Ubuntu 16.04 VM to test each algorithm and wrote a few lines of Python to fix some incompatibility issues.

If one individual confronts with the problem of extracting text from an image where the OCR software is not capable of detecting characters, a simple revision of the presented algorithms can come in very handy and for further approach methods he can study the standard presented techniques and search for a common implementation.

Acknowledgement: This work was supervised by Associate Professor *Elena Nechita*, from *University "Vasile Alecsandri" of Bacau, Romania.*

5. REFERENCES

[1] Otsu N (1979) A threshold selection method from Gray-level histograms – https://ieeexplore.ieee.org/document/4310076

[2] J.N. Kapur, P.K. Sahoo, A.K.C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram", Computer Vision, Graphics, and Image Processing, Vol. 29 – https://www.sciencedirect.com/science/article/pii/0734189X85901252

[3] J. Kittler, J. Illingworth, "Minimum error thresholding", Pattern Recognition, Vol. 19, No. 1, 1986 – https://www.sciencedirect.com/science/article/abs/pii/0031320386900300

[4] Leung CC, Chan KS, Chan HM, Tsui WK (2005) A new approach for image enhancement applied to low-contrast–low-illumination IC and document images. – http://hub.hku.hk/handle/10722/73568

[5] Nomura S, Yamanaka K, Shiose T, Kawakami H, Katai O (2009) Morphological preprocessing method to thresholding degraded word images. – https://www.sciencedirect.com/science/article/abs/pii/S016786550900049X

[6] *Deskew algorithm* – https://github.com/kakul/Alyn

[7] Contrast Enhancement algorithm – https://github.com/baidut/OpenCE

[8] Shadow Removal algorithm – https://github.com/kittenish/Image-Shadow-Detection-and-Removal

[9] Noise cleaner algorithm – http://www.fmwconcepts.com/imagemagick/noisecleaner/index.php

[10] Dewarp algorithm – https://github.com/mzucker/page_dewarp

[11] Obafemi-Ajayi T, Agam G, Frieder O (2010) Historical document enhancement using LUT classification. – https://link.springer.com/article/10.1007/s10032-009-0099-3

TIBERIU Bujor
"Vasile Alecsandri" University of Bacău
Faculty of Sciences
Bacău ,Calea Mărășești 157
ROMANIA
E-mail: bujor.tiberiu@gmail.com