Machine Perception DS/NC/ESD 863

Denoising Dirty Documents

Project Proposal

Rishabh Manoj (IMT2013035) Simran Dokania (IMT2013044)
Udbhav Vats (IMT2013055) Sriveda Reddy (IMT2013047)

March 10, 2017

1 Goal Statement Description

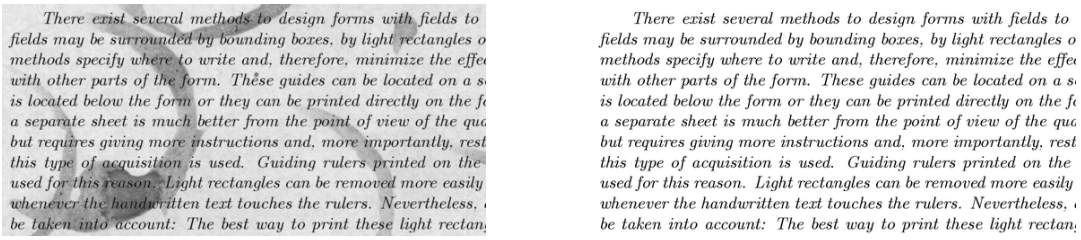
Optical Character Recognition (OCR) is the process of getting typed or handwritten documents into a digitized format. The motivation of converting to a digitized format is to ensure security, accessibility, edit-ability and ease of searching and sharing. Also, digital documents don't get dirty and cannot be ruined by coffee stains. [2]

Unfortunately, a lot of documents eager for digitization are being held back. Coffee stains, faded sun spots, dog-eared pages, and lot of wrinkles are keeping some printed documents offline and in the past.

Given a dataset of images of scanned text (synthetic images) that are "noisy" with stains and wrinkles, we propose to clean up the noise and help with the digitization process.

2 Dataset

Kaggle provided a dataset which consists of two sets of images - train and test. These images contain various styles of text, to which synthetic noise has been added to simulate real-world, messy documents. The dirty images contain stains as well as creased paper. The training set also includes the cleaned up images of those found in the test file (train_cleaned) [2]. By clean, we mean black letters on a white background. Below are two sample images:



(a) Original Image (b) Cleaned Image

Figure 1: Sample Images

3 Proposed Plan of Execution

A gray-scale image as shown above can be thought of as a three dimensional surface. The x and y co-ordinates gives us the location of a particular pixel in the image and the z co-ordinate gives us the brightness of the image at that location. The greater the brightness, the whiter the image at that location. Therefore, our task from a mathematical standpoint is to convert one three dimensional surface to other three dimensional surface such that we are just left with a good contrast writing with no stains or paper creases. We propose few methods/approaches which are as follows:

1. **Least Square Regression** - We assume that there is a linear relationship between the brightness of the dirty images and the cleaned images and therefore try to fit a linear model. However, there will be some noise which should be cleaned like a clump of pixels which are neither white nor black. So our first model would be a linear transformation model.

2. **Image Thresholding** - Just to fine tune the above approach, we will try image thresholding. In this approach, we will use the fact that writing is darker than the background. Thus if we choose a good threshold value, then we can say that everything above that particular value will be black and everything below that particular value will be white. Implementing this method might help us get rid of some noise.

3. **Adaptive Thresholding** - In fixed thresholding, we use the fact the the writing on the document is significantly darker than the background that contains noise in the form of creases or dog eared pages. Therefore, if we choose a good enough threshold value we can get rid of the creases and dog ears. However, intuitively this would not work with noise like coffee stains which are also pretty dark. The threshold chosen may remove parts of the writing too in the attempt to remove the stains. [3]

We can use adapative thresholding to overcome this problem. We have identified that coffee stains are dark so fixed thresholding can't efficiently distinguish between the dark writing and dark coffee stains. However, a useful observation is that the writing is definitely darker than the stains. Since adaptive thresholding looks at the pixel values in the user defined neighborhood of each pixel to determine the threshold for that pixel, it may work to clean up the coffee stains. Essentially, for each pixel in the coffee stain, the surrounding pixels will also be relatively lighter as opposed to the pixels surrounding darker writing. Exploiting this observation, adaptive thresholding may prove to be a good approach to deal with coffee stains. [1]

4. **Canny Edge Detection and Morphology** - From experience of using adaptive thresholding we anticipate that it may leave a pattern of specks of pixels in the place of the stain, which need to be cleaned up. In order to do so we can try the most standard image processing technique - edge detection. The canny edge detector will recognise the specks but also may recognise the writing as edges.[1] We can use image morphology techniques on the edges by identifying certain distinguishing properties of the edges around the stains and those around the writing to remove only the ones around the stains while retaining those around the writing. We can use dilation and erosion techniques.

5. **Using Median Filter** - One other way is to use median filter to get rid of the noise. Median filter replaces the value of each pixel with the median of the surrounding pixels. On median filtering the image, the small irrelevant details gets blurred out and only the significant and visible details are left [1]. Here, we are assuming that median filtering will blur out the text and only the background details (which may include coffee stains, paper creases and other noises) will remain. We may get the denoised image by subtracting this from the origninal image.

6. **Character Detection & Segmentation** - The dataset deals with text images. We propose a method to clean the images using OCR and segmentation. Optical Character Detection is used to detect the characters. With the location of the characters we segment the image into two categories *foreground*, which contains the character and *background*, which contains the noise(wrinkles & coffee stains). We then superimpose the foreground image on a white image (cleaned background image).

4 Main Challenges

1. **Least Square Regression** - This model depends on the assumption that there is a linear relationship between the brightness of pixels in the cleaned and dirty images.

2. **Image Thresholding** - Enormous amount of hit and trial will be required to come up with an appropriate threshold value.

3. **Adaptive Thresholding** - We need to do some amount of trial and error to come up with a suitable constant value to compute our threshold function. Further, we anticipate that although adaptive thresholding may clean up coffee stains to a good degree, it may leave a residual speckled pattern where the stain originally was. We need to come up with a way to remove those patterns without affecting the writing in the document.

4. **Canny Edge Detection and Morphology** - We have to identify distinguishing properties of the edges around the stains and edges around the writing so that we can exclusively eliminate those around the stains. It will require some amount of trial and error and observation on the canny edge detector output to be able to come up with an efficient way to do this for all images.

5. **Using Median Filtering** - The main challenge here lies in finding the correct size of kernel used for median filtering and also to find the correct way of subtracting the background image from the original such that the text is retained.

6. **Character Detection & Segmentation** - Existing character detection might not detect the characters properly if the intensity of the stains match the text. We speculate that the OCR gives an approximate location of the characters, which might not work effectively when there are noises so close to the characters.

5 Learning Objective

The objective of this project is to clean noisy text images that can later be digitalized. *Image Processing* techniques are vital to ensure the completion of the project. We expect, at the very least, to have a good understanding of common image processing techniques like thresholding, segmentation, morphological operation, edge detection, filtering, character detection.

Machine Learning algorithm will supplement the existing techniques to increase accuracies. We expect to learn how to implement these algorithms for images.

As this is a team project, we will use Git to collaborate our work. Git helps in sharing code, correcting bugs and solving problems across a team.

References

[1] Colin blog. http://tinyurl.com/gnptby6. Accessed: 2017-03-10.

[2] Kaggle - denoising dirty documents. http://tinyurl.com/z4ukatx. Accessed: 2017-03-10.

[3] Kaggle blog. http://tinyurl.com/gnedxjq. Accessed: 2017-03-10.