Computer Vision

Laxman Desai

June 15, 2021

Contents

1 Optical Character Recognition

We will try and build a model that can break CAPTCHAs. What's a CAPTCHA? It's basically a the test to determine if the user is a bot or a human. We will first attempt at breaking text based CAPTCHAs.

Stages:

1. Data acquisition
2. Preprocessing
3. Building a model
4. Training the model
5. Testing the model

1.1 Preprocessing

We cannot input an image directly for the OCR system. Some pre-processing has to be done on the image so that it becomes easy for the OCR model to recognize the information in the image.

Preprocessing includes the following:

1. Skew Correction
2. Removing Lines (Horizontal & Vertical)
3. Building a model
4. Testing

1.1.1 Skew Correction

Image obtained from the previous stage may not be correctly oriented, It may be aligned at any angle. So we need to perform skew correction to make sure that the image forwarded to subsequent stages is correctly oriented.

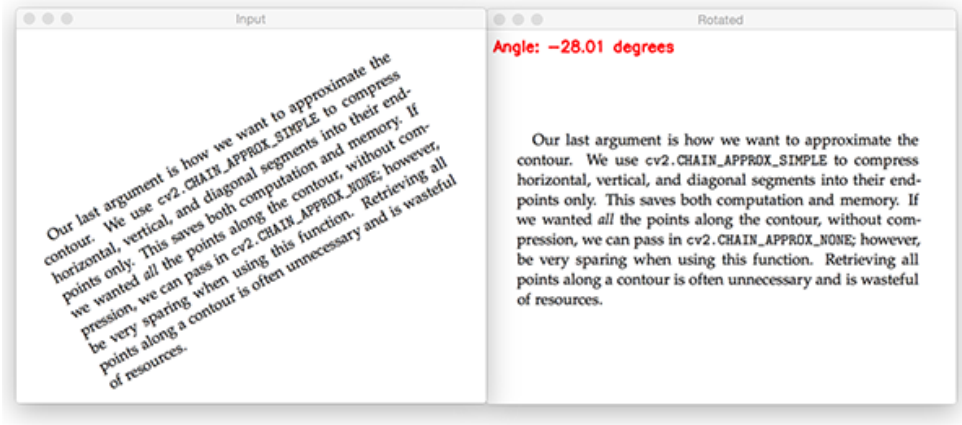

Figure 1: Picture caption

1.1.2 Binarization

That is, converting a coloured image to a black and white binary image. In practice, there exists an intermediate grayscale image.

Coloured image → Grayscale image → Binary image

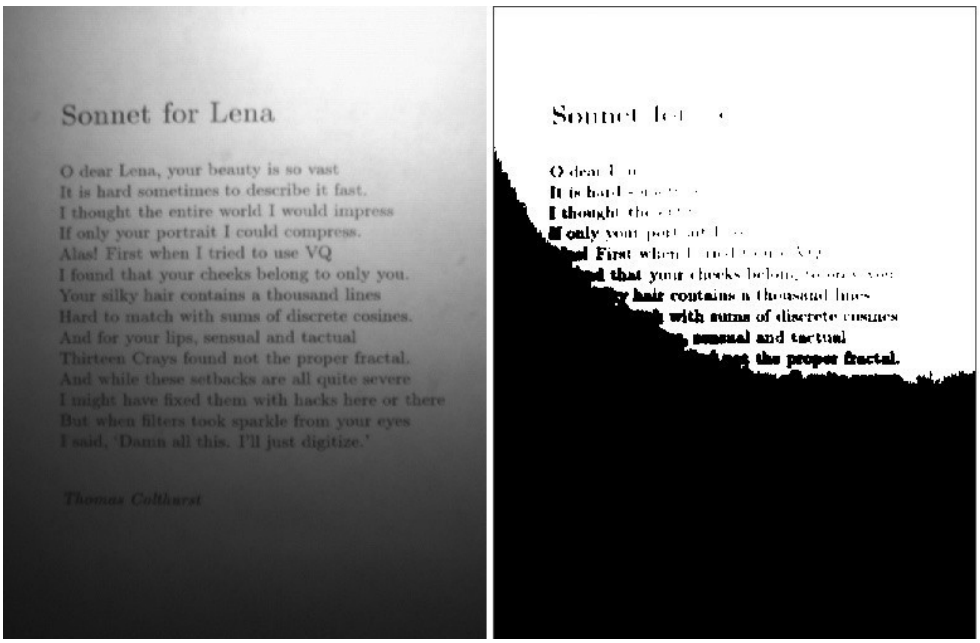

Figure 2: Binarization using a threshold on the image captured under non-uniform lighting.

Converting an RGB image to a grayscale image is easy. To a computer, a black pixel has a value of 0 and a white pixel has a value of 255. Converting a grayscale to B&W requires one to set a threshold value. If the pixel value is greater than the threshold, it is considered as a white pixel, else its as a black pixel. This naive strategy fails when lighting conditions are not uniform in the image. Here we can use Otsu's method or adaptive thresholding.

Otsu's Binarization: Since we are working with bimodal images, Otsu's algorithm tries to find a threshold value (t) which minimizes the weighted within-class variance given by the relation:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

where,

$$q_1(t) = \sum_{i=1}^{t} P(i) \quad \& \quad q_2(t) = \sum_{i=t+1}^{I} P(i)$$

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \quad \& \quad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)}$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \& \quad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Python code:

```
import cv2 as cv

# global thresholding
ret1,th1 = cv.threshold(img,127,255,cv.THRESH_BINARY)

# Otsu's thresholding
ret2,th2 = cv.threshold(img,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)

# Otsu's thresholding after Gaussian filtering
blur = cv.GaussianBlur(img,(5,5),0)
ret3,th3 = cv.threshold(blur,0,255,cv.THRESH_BINARY+cv.THRESH_OTSU)
```

1.1.3 Removing Lines (Horizontal & Vertical)