Orientation Correction

Veer Singh

July 31, 2021

1 Introduction

This algorithm corrects the orientation of scanned documents. Works on text documents with best results between -45 to 45 degree of rotation.

2 Applying Preprocessing on the Image

Gaussian Blur and Otsu Threshold is first applied to the input image.

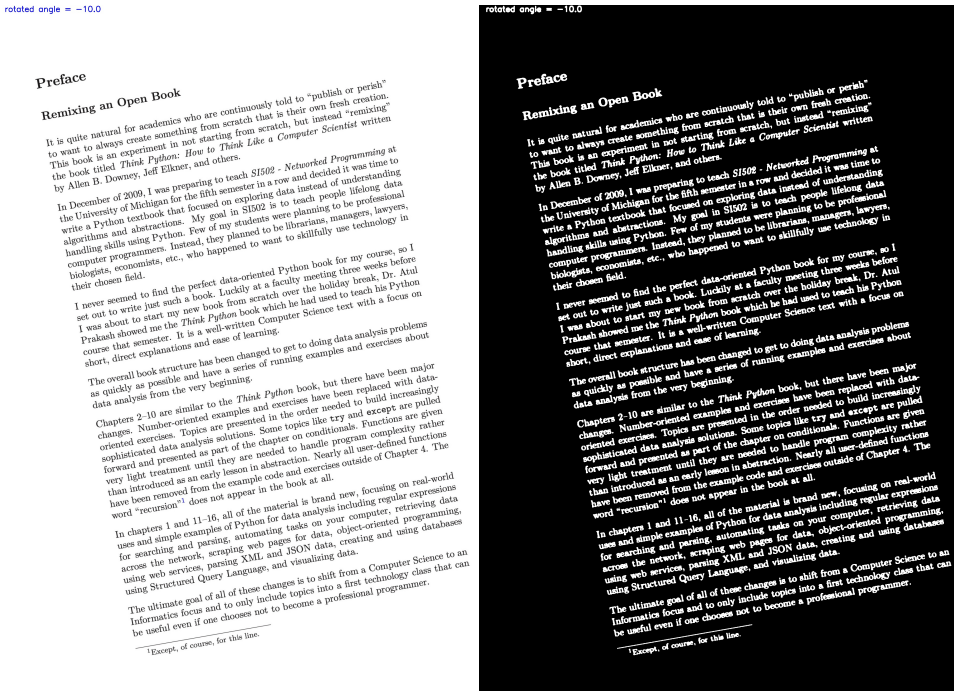


Figure 1: (a) Original Image (b) Image with Gaussian Blur and Thresholding

3 Apply Dilation and Find the Largest Contour

Next we apply dilation to get the paragraphs and then find the largest contour. This is done so that we can apply the cv2.minAreaRect method which gives us the angle at which we need to rotate the text.

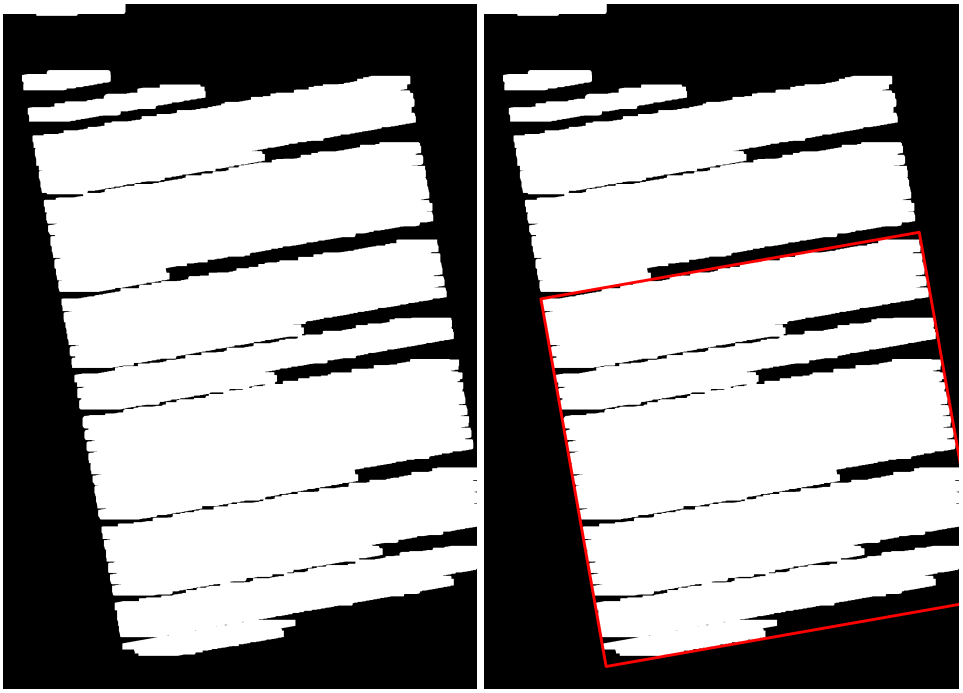


Figure 2: (a) Dilation Applied (b) Largest Contour

4 Getting angle of rotation

I applied cv2.minAreaRect() on the largest contour to get the minimum area rotated rectangle which gives the angle of rotation as the third element. The following logic is applied to this angle to get the angle the image is rotated at:

```
angle = minAreaRect[2]

# Calculate the skew angle
if angle > 45:
    angle = 90 - angle
else:
    angle = -angle
skew_angle =  round((-1.0 * angle), 2)
```

5 Creating Dataset to Check Performance

I used two different types of scanned documents to assess the performance. One was a page from a digital pdf which basically represents the best case scenario when scanning a document. The second image was a scan of an old document with a bit of noise.

Preface

Remixing an Open Book

It is quite natural for academics who are continuously told to "publish or perish" to want to always create something from scratch that is their own fresh creation. This book is an experiment in not starting from scratch, but instead "remixing" the book titled *Think Python: How to Think Like a Computer Scientist* written by Allen B. Downey, Jeff Elkner, and others.

In December of 2009, I was preparing to teach *SI502 - Networked Programming* at the University of Michigan for the fifth semester in a row and decided it was time to write a Python textbook that focused on exploring data instead of understanding algorithms and abstractions. My goal in SI502 is to teach people lifelong data handling skills using Python. Few of my students were planning to be professional computer programmers. Instead, they planned to be librarians, managers, lawyers, biologists, economists, etc., who happened to want to skillfully use technology in their chosen field.

I never seemed to find the perfect data-oriented Python book for my course, so I set out to write just such a book. Luckily at a faculty meeting three weeks before I was about to start my new book from scratch over the holiday break, Dr. Atul Prakash showed me the *Think Python* book which he had used to teach his Python course that semester. It is a well-written Computer Science text with a focus on short, direct explanations and ease of learning.

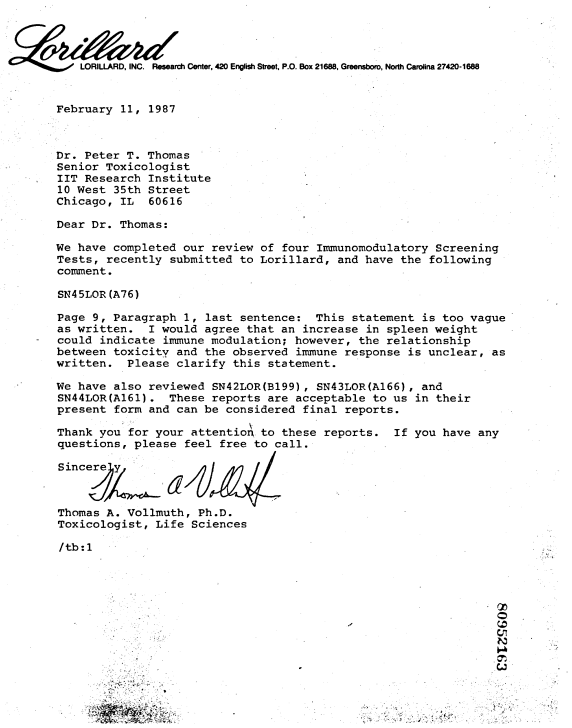
The overall book structure has been changed to get to doing data analysis problems as quickly as possible and have a series of running examples and exercises about data analysis from the very beginning.

Chapters 2–10 are similar to the *Think Python* book, but there have been major changes. Number-oriented examples and exercises have been replaced with data-oriented exercises. Topics are presented in the order needed to build increasingly sophisticated data analysis solutions. Some topics like `try` and `except` are pulled forward and presented as part of the chapter on conditionals. Functions are given very light treatment until they are needed to handle program complexity rather than introduced as an early lesson in abstraction. Nearly all user-defined functions have been removed from the example code and exercises outside of Chapter 4. The word "recursion"[1] does not appear in the book at all.

In chapters 1 and 11–16, all of the material is brand new, focusing on real-world uses and simple examples of Python for data analysis including regular expressions for searching and parsing, automating tasks on your computer, retrieving data across the network, scraping web pages for data, object-oriented programming, using web services, parsing XML and JSON data, creating and using databases using Structured Query Language, and visualizing data.

The ultimate goal of all of these changes is to shift from a Computer Science to an Informatics focus and to only include topics into a first technology class that can be useful even if one chooses not to become a professional programmer.

[1]Except, of course, for this line.



Lorillard
LORILLARD, INC. Research Center, 420 English Street, P.O. Box 21688, Greensboro, North Carolina 27420-1688

February 11, 1987

Dr. Peter T. Thomas
Senior Toxicologist
IIT Research Institute
10 West 35th Street
Chicago, IL 60616

Dear Dr. Thomas:

We have completed our review of four Immunomodulatory Screening Tests, recently submitted to Lorillard, and have the following comment.

SN45LOR(A76)

Page 9, Paragraph 1, last sentence: This statement is too vague as written. I would agree that an increase in spleen weight could indicate immune modulation; however, the relationship between toxicity and the observed immune response is unclear, as written. Please clarify this statement.

We have also reviewed SN42LOR(B199), SN43LOR(A166), and SN44LOR(A161). These reports are acceptable to us in their present form and can be considered final reports.

Thank you for your attention to these reports. If you have any questions, please feel free to call.

Sincerely,

Thomas A. Vollmuth, Ph.D.
Toxicologist, Life Sciences

/tb:1

80952163

Figure 3: (a) Test Image 1 (b) Test Image 2

Then I created a dataset of 1400 images of each image rotated at angles from -70 degrees to 70 degrees with 0.1 degree increments (For example: 10.0 , 10.1, 10.2, 10.3, ...).

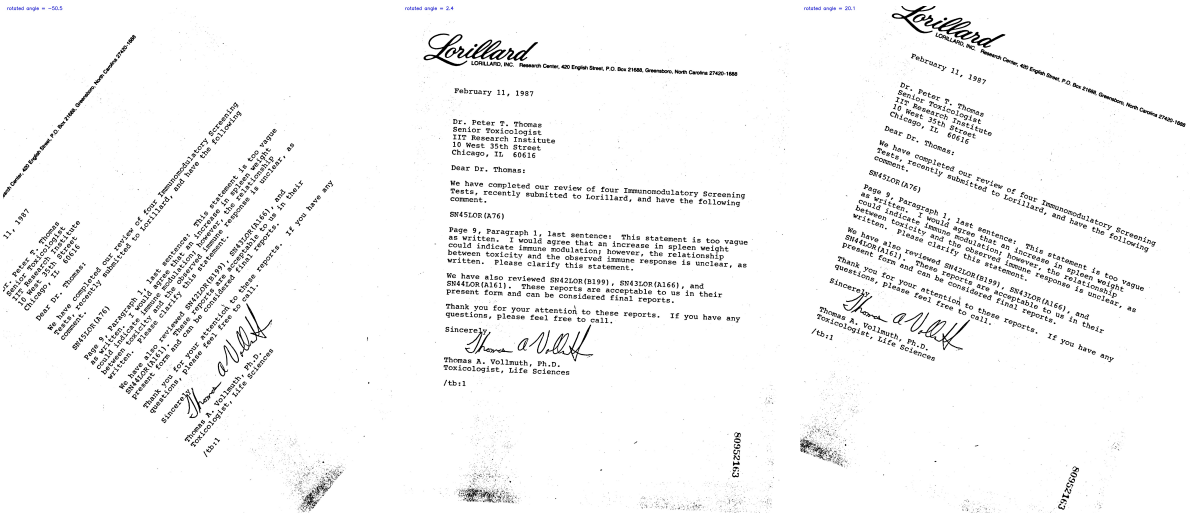


Figure 4: Example Images from Created Dataset

I also created a json file where the ground truth angle values of the images were saved. Example of the ground truth values:

```
{
  "1.jpg": -70.0,
  "2.jpg": -69.9,
  "3.jpg": -69.8,
  .
  .
  .
  "1398.jpg": 69.7,
  "1399.jpg": 69.8,
  "1400.jpg": 69.9
```

}

6 Testing Accuracy

I iterated through all the images and applied the algorithm to find the angle of rotation. This value was compared to the ground truth values.
Percentage difference was calculated = $\|(CalculatedAngle - GroundTruthAngle)/GroundTruthAngle\|$*100

I then plotted a scatter plot of 1400 angle values and the percentage difference between true value and calculated value.

```
Image #1 |real angle = -70.0 |calculated angle = 22.19 |difference = 131.7%
Image #2 |real angle = -69.9 |calculated angle = 22.3 |difference = 131.9%
Image #3 |real angle = -69.8 |calculated angle = 22.38 |difference = 132.06%
.
.
.
Image #1398 |real angle = 69.7 |calculated angle = -20.43 |difference = 129.31%
Image #1399 |real angle = 69.8 |calculated angle = -20.31 |difference = 129.1%
Image #1400 |real angle = 69.9 |calculated angle = -20.22 |difference = 128.93%
```

7 Results

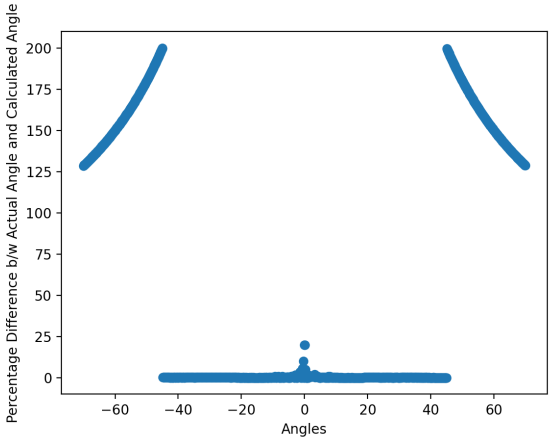


Figure 5: Percentage Difference at different angles for Test Image 1

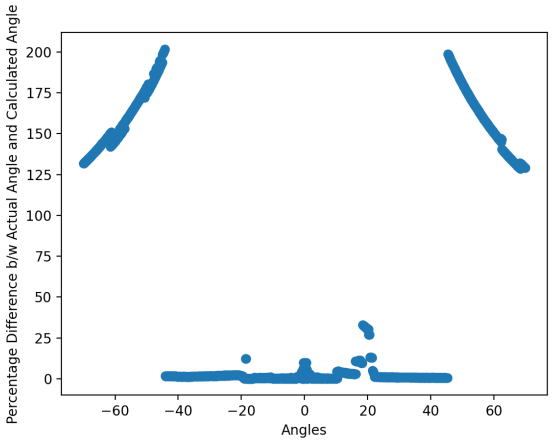


Figure 6: Percentage Difference at different angles for Test Image 2

As the results show this method works great between angles -45 and 45 where it has almost 0% difference between the true value and the calculated value. But this method completely breaks down at angles less than -45 and greater than 45 degrees.