SCC0251 - Image Processing
Final Project Report

Group 21:
Guilherme de Freitas Perinazzo - 7153362
Gustavo Dutra Santana 8532040

1. Introduction

Often we need to create quick, digital indexes of physical books. Accessing information quickly may be required in multiple situations, and searching on a physical book is an extremely slow task. Finding the page, section, and specific sentence you need may take multiple minutes.

We can solve this issue by storing the book’s content in digital form. A simple approach to this is to store an image of the page with few tags and keywords to allow it to be indexed. Thanks to modern cell phones, acquiring these images becomes an easy task.

This approach, however, has two major pitfalls:

- Inability to search in the inner text of the book, only on the tags.
- Images uses a lot more storage than the text they represent.

Another approach is to use algorithms capable of recognizing the text written in the image, transforming it into a simple text string. This process is complex, and involves multiple steps, many that require image manipulation.

2. Project Goals

This project aims at preparing characters from a picture of a book page to be taken by a classification algorithm that may attempt to classify them.

To achieve this, we use multiple image processing techniques to try to predict where each line of text in the image is to later deduce where each character inside the line is located.

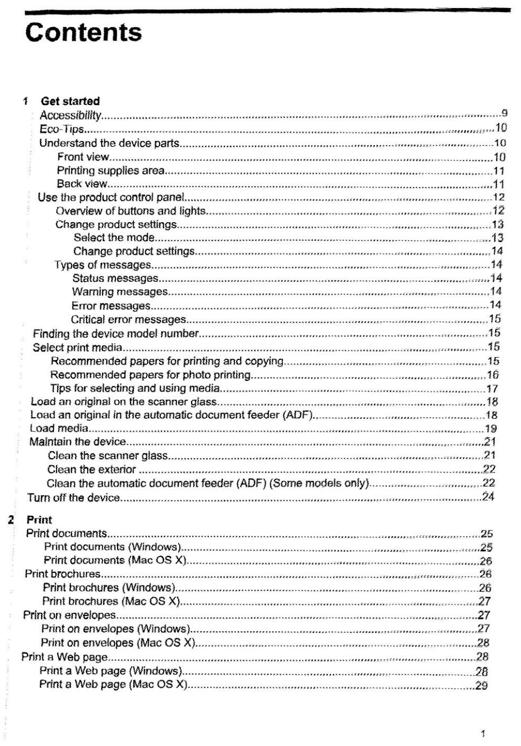
Given that our target is to identify text from books, we can safely assume that most of the page will be dominated by text.

3. Image Acquisition

Image acquisition for this project was mostly made by taking pictures of book pages with a cellphone camera, as those are our target.

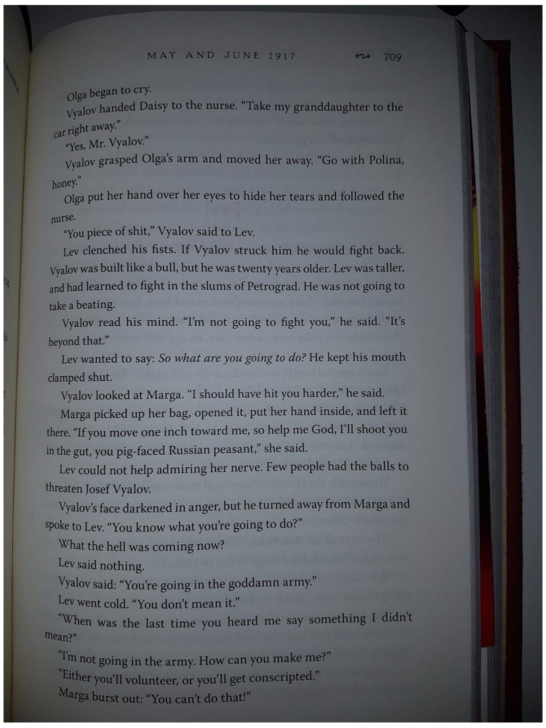
A couple images were generated on the computer, to have test cases without noise. We also used an image that is a scanned book page, in order to test the algorithm without the camera lens reflection. Some images were also generating by taking a picture and extracting a part of the text.

All images used can be found on the github project, under the folder Images. The following are a few scaled down examples of images we used:



Contents

1



MAY AND JUNE 1917 709

Olga began to cry.

Vyalov handed Daisy to the nurse. "Take my granddaughter to the car right away."

"Yes, Mr. Vyalov."

Vyalov grasped Olga's arm and moved her away. "Go with Polina, honey."

Olga put her hand over her eyes to hide her tears and followed the nurse.

"You piece of shit," Vyalov said to Lev.

Lev clenched his fists. If Vyalov struck him he would fight back. Vyalov was built like a bull, but he was twenty years older. Lev was taller, and had learned to fight in the slums of Petrograd. He was not going to take a beating.

Vyalov read his mind. "I'm not going to fight you," he said. "It's beyond that."

Lev wanted to say: *So what are you going to do?* He kept his mouth clamped shut.

Vyalov looked at Marga. "I should have hit you harder," he said.

Marga picked up her bag, opened it, put her hand inside, and left it there. "If you move one inch toward me, so help me God, I'll shoot you in the gut, you pig-faced Russian peasant," she said.

Lev could not help admiring her nerve. Few people had the balls to threaten Josef Vyalov.

Vyalov's face darkened in anger, but he turned away from Marga and spoke to Lev. "You know what you're going to do?"

What the hell was coming now?

Lev said nothing.

Vyalov said: "You're going in the goddamn army."

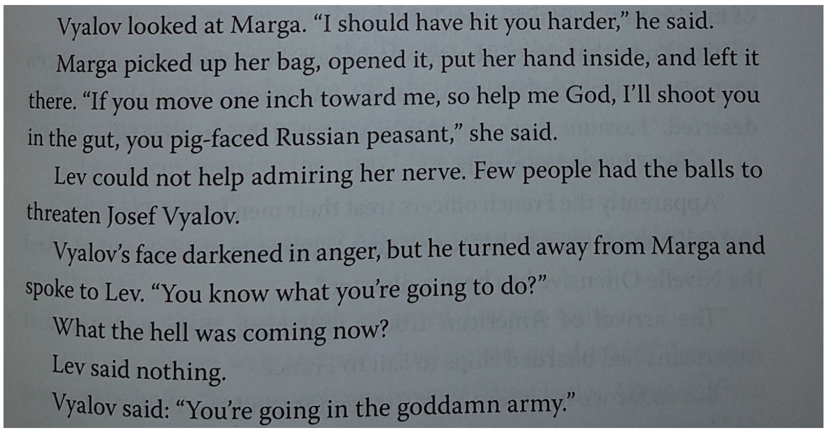
Lev went cold. "You don't mean it."

"When was the last time you heard me say something I didn't mean?"

"I'm not going in the army. How can you make me?"

"Either you'll volunteer, or you'll get conscripted."

Marga burst out: "You can't do that!"



Vyalov looked at Marga. "I should have hit you harder," he said.

Marga picked up her bag, opened it, put her hand inside, and left it there. "If you move one inch toward me, so help me God, I'll shoot you in the gut, you pig-faced Russian peasant," she said.

Lev could not help admiring her nerve. Few people had the balls to threaten Josef Vyalov.

Vyalov's face darkened in anger, but he turned away from Marga and spoke to Lev. "You know what you're going to do?"

What the hell was coming now?

Lev said nothing.

Vyalov said: "You're going in the goddamn army."

4. Methods Used

The following image processing techniques were used to pre-process and extract characters from the images:

- **Convert to grayscale**:
  - Since text is generally not affected by color, we converted all images to grayscale in order to more easily manipulate them.
- **Gaussian Blur**:
  - A small blur was added to try to smooth out any noise in the image, allowing the next algorithm to better separate the text from the background.
- **Gaussian Adaptive Thresholding**:
  - To try to remove the page's background and any possible noise in the image, this thresholding technique is used.
  - We also remove from any pixel being thresholded 25% of the median pixel intensity in the image, further helping differentiate text from background.
  - Through tested this seemed to be the most effective algorithm for removing background textures from the image.
- **Canny Edge Detector**:
  - We create an edge image from the thresholded image, to be used with the Hough Lines algorithm.
- **Hough Lines**:
  - This algorithm finds every contiguous line of a certain length (defined as the algorithm's threshold).
  - As the OpenCV version does not output the total length of the lines, the algorithm is ran multiple times on the edge image with decreasing thresholds until it outputs at least one line.
  - The angle of each line outputted is calculated, and a rotation angle is calculated as the average of the lines' angles.
- **Rotation**:
  - The thresholded image is rotated by the angle found in the step above. This rotate the lines so that they are parallel to the image's top and bottom.
- **Erosion**:
  - To help separate the characters, an erosion is applied to the rotated image.
  - A 3x3 with a cross shape kernel is used
- **Horizontal Projection**:
  - This algorithm calculates the sum of every pixel value for every horizontal line in the image.

    - This is ran over the inverse of the eroded image, so that text pixels are white (high value) while background pixels are black (low value)
    - For each line result, we subtract 50% of the average line value, to help ignore some of the background noise the image may still have. The negative values are then clipped to 0.
- **Text Line Detection**:
    - We use the horizontal projection calculated on the previous step to detect lines of text.
    - A line of text will generally appear as a sequence of non-zero lines in the projection, surrounded by lines with zero value (that represents the space between the lines).
    - We record the positions of the upper and lower limits of every line.
- **Vertical Projection**:
    - Like the horizontal projection, but sums all the pixel values in a vertical line
    - This is ran for every line detected in the step above, with the vertical length set to the line’s vertical size.
    - Unlike the horizontal projection, no treatment is done here to attempt to reduce noise, as characters may have a single pixel on each line.
- **Character Detection**:
    - With the vertical projection of every line, we once again find every sequence of non-zero lines, surrounded by zero'd lines.
    - This gives us the left and right limits of every character, as long as there is at least one full vertical line of white pixels separating them.

The current algorithm stops here, after that the detected bounds for pixels are drawn into the rotated line and it’s saved on the current folder as ***rotated.png***. Other intermediary representations (edges, grayscale and eroded) are also saved for debugging.

5. Problems not solved

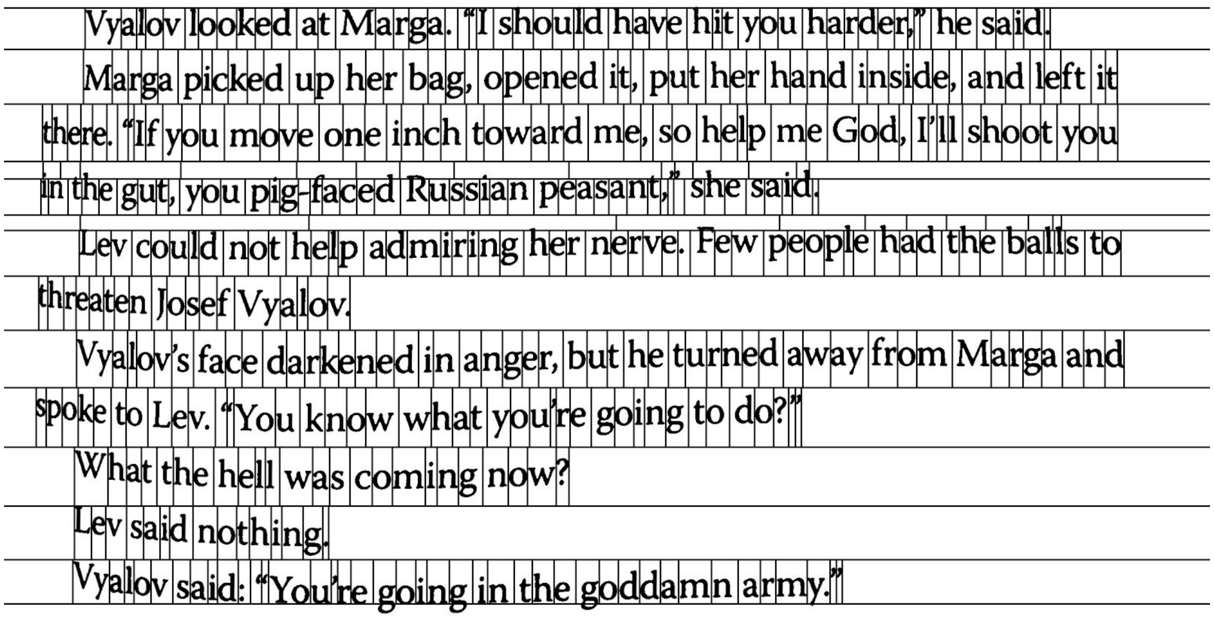
A few problems were found during development, that were not predicted:

- Camera distortion: cellphone cameras seem to distort pictures the further away from the center. This leads to lines on the top and bottom half of page to be not straight. This makes the line detection part not work as it depends on lines being completed separated by at least a blank line.

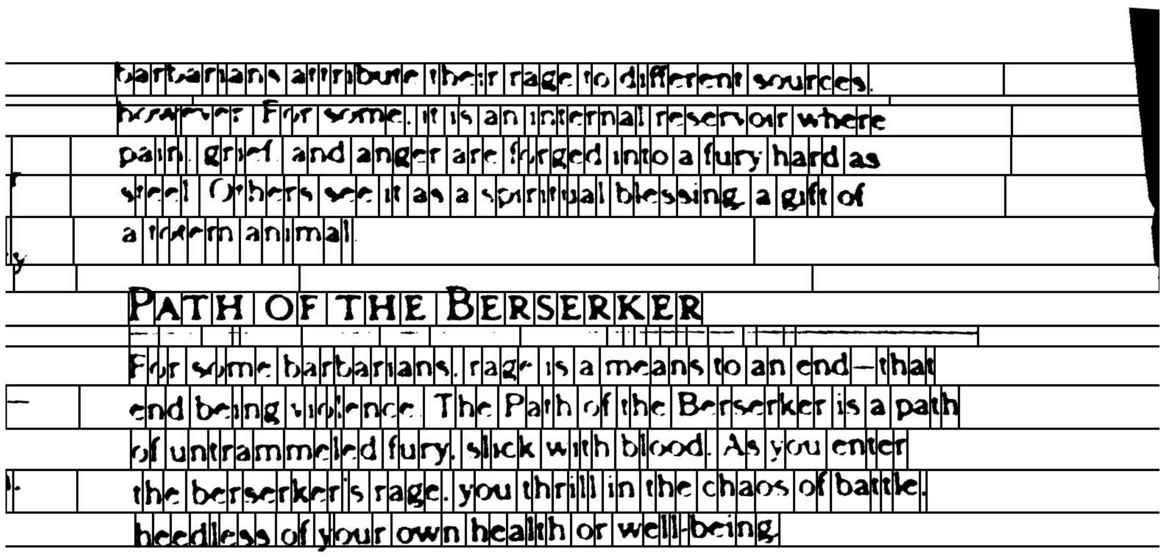
- Paper distortion: if you ever opened a book, you'll know that paper does not usually sit completely straight. This seem to have the same effect as the camera distortion, making text lines not straight
- Background noise: any noise that isn't picked up by the thresholding stage can negatively affect the line detection algorithm.
- Feature extraction: due to time constraints, we weren't able to implement the feature extraction part, as adjusting the pre-processing and character detection algorithms was a very time consuming task. We also both work, so the time we had to work on this was very limited.

6. Results

The following images are a few results of the algorithm. These are the rotated images with the detected lines and characters bounds draw onto them:



Vyalov looked at Marga. "I should have hit you harder," he said.
Marga picked up her bag, opened it, put her hand inside, and left it
there. "If you move one inch toward me, so help me God, I'll shoot you
in the gut, you pig-faced Russian peasant," she said.
Lev could not help admiring her nerve. Few people had the balls to
threaten Josef Vyalov.
Vyalov's face darkened in anger, but he turned away from Marga and
spoke to Lev. "You know what you're going to do?"
What the hell was coming now?
Lev said nothing.
Vyalov said: "You're going in the goddamn army."



barbarians attribute their rage to different sources.
However. For some, it is an internal reservoir where
pain, grief, and anger are forged into a fury hard as
steel. Others see it as a spiritual blessing, a gift of
a totem animal.
PATH OF THE BERSERKER
For some barbarians, rage is a means to an end—that
end being violence. The Path of the Berserker is a path
of untrammeled fury, slick with blood. As you enter
the berserker's rage, you thrill in the chaos of battle,
heedless of your own health or well-being.

Olga began to cry.

Vyalov handed Daisy to the nurse. "Take my granddaughter to the car right away."

"Yes, Mr. Vyalov."

Vyalov grasped Olga's arm and moved her away. "Go with Polina, honey."

Olga put her hand over her eyes to hide her tears and followed the nurse.

"You piece of shit," Vyalov said to Lev.

Lev clenched his fists. If Vyalov struck him he would fight back. Vyalov was built like a bull, but he was twenty years older. Lev was taller, and had learned to fight in the slums of Petrograd. He was not going to take a beating.

Vyalov read his mind. "I'm not going to fight you," he said. "It's beyond that."

Lev wanted to say: *So what are you going to do?* He kept his mouth clamped shut.

Vyalov looked at Marga. "I should have hit you harder," he said.

Marga picked up her bag, opened it, put her hand inside, and left it there. "If you move one inch toward me, so help me God, I'll shoot you in the gut, you pig-faced Russian peasant," she said.

Lev could not help admiring her nerve. Few people had the balls to threaten Josef Vyalov.

Vyalov's face darkened in anger, but he turned away from Marga and spoke to Lev. "You know what you're going to do?"

What the hell was coming now?

Lev said nothing.

Vyalov said: "You're going in the goddamn army."

Lev went cold. "You don't mean it."

"When was the last time you heard me say something I didn't mean?"

"I'm not going in the army. How can you make me?"

"Either you'll volunteer, or you'll get conscripted."

Marga burst out: "You can't do that!"

The function runs the Harris edge detector on the image. Similarly to cornerMinEigenVal() and cornerEigenValsAndVecs(), for each pixel (x, y) it calculates a 2×2 gradient covariance matrix $M^{(x,y)}$ over a blockSize $\times$ blockSize neighborhood. Then, it computes the following characteristic:

Contents