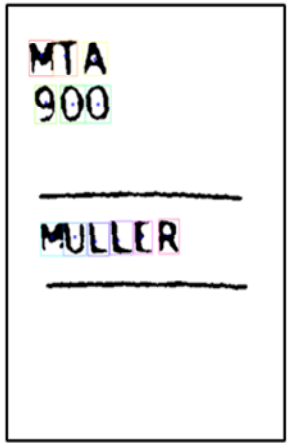
lib-indexer

EDBV WS 2019/2020: AG C 3
Anand Eichner (11808244)
Laurenz Edmund Fiala (11807869)
Anna Nieto-Berezhinskaya (01223066)
Aleksandar Vucenovic (01635282)



json

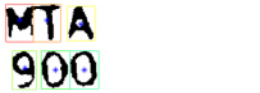
```
letterCode: MTA
digitCode : 900
author : MULLER
```

Final Prerequisites

- New
  - Labels have to be bright enough for thresholding
- Old
  - The books must be vertically aligned (+-5°)
  - The image must not be tilted more than 30°
  - Resolution must be high enough for OCR
  - The image must be a RGB image
  - The image must high enough contrast
  - White balance must be applied in the image

Pipeline

- Perspective Correction
  - Preprocessing
  - Hough-Transformation
  - Geometric Image Transformation
- Label Detection
  - Preprocessing
  - Integral Imaging
  - Otsu Threshold
- Preprocessing for OCR
  - Dilation
  - Regionprops + Segment (“Word”) sorting
- OCR
  - Normalized Cross Coefficient Algorithm
  - Sum of squared Differences Algorithm



Actual: “MTA 900 MÜLLER”

SSD: “HTA SQQ YVLLLR”

NCC: “LLL LLL FLJJR”

Problems

- Label Detection
  - Some labels aren't rectangles
  - Some labels are too dark
  - Some labels are very thin
  - False positives
- Perspective Correction
  - Wrong line detection bloated RAM
- Preprocessing OCR
  - Sorting and Segmenting Words
- OCR
  - Resolution of letters is small
  - NCC padding might be a problem (giving too many L's)
  - SSD isn't the most precise algorithm

X still a problem

X solved

Evaluation

- Qualitative
  - Which patterns do we have to recognize?
    - Shelves -> Labels -> Blobs (Potential characters) -> Characters
  - How does the image have to be transformed, to enable a good label detection
    - Gauss-Filter
    - Convolution Kernel for Harris choosen bigger
  - What indexing is useful, and how do we make the results human-readable?
    - We use a Matlab struct for the data, which is being transformed into a .json format
    - Additionally the labels and words are sorted from top left to bottom right
  - Which pre-processing do we need for the OCR to work properly?
    - Binarizing, Dilation, Regionprops, Slicing, Segmentation into Patches/Blobs

Evaluation

- Quantitative
  - How many books/labels do we recognize in average?
    - About ~80%
  - Welche Auflösung ist notwendig, um akzeptable Ergebnisse zu erzielen?
    - Hängt von der Entfernung zu den Labels ab
  - How does the tilting of the image affect the label detection
    - Through image straightening we ensure, that there are only minimal changes in the OCR
    - The more tilting there is the better the resolution must be
  - How does the degree of fullness of the shelves affect the label detection
    - Less background is good as we get less false positives
    - Thin books could be seen as one big label

Result

- Used Dataset:

- One of the labels detected:

- Result in json:

{"wordOne":"BAU","wordTwo":"107","author":"SCHNLTDL","bounds":[3428,2462,3603,2727]},

THANKS FOR LISTENING