CSE 344: Computer Vision

Instructor: Dr Saket Anand

Automated Document Authenticator

Authors:

- Mohammad Atif Quamar [2020523]
- Vishesh Rangwani [2020154]
- Rahul Agrawal [2020108]

Problem Statement

To Automate the process of Authenticating documents from their captured Images using Computer Vision Techniques studied in the course

Aims:

- Increased Automation
- Faster Processing
- To reduce human involvement
  - Reduced human error
  - Increased efficiency
  - Reduced human biases
- Under the hood authentication
- Privacy
- Security
- Versatile across different document types

Dataset Description

Pretrained Models/Libraries used:

- OpenCV: For RANSAC, finding Homography, Feature Detection, canny and hough.
- Pytesseract: NLP based Text Detection
- Other Libraries: Numpy, Matplotlib and os

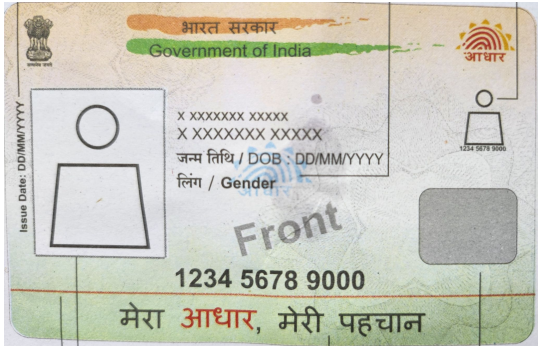
DataSet (Primary for testing):

Aadhar Card Images for Aadhar Authentication: Drive Link

- Dataset primary contains
  - Query Image: Perfectly cropped sample aadhar card
  - Test Images: Random pics of PVC aadhar card

Demonstration

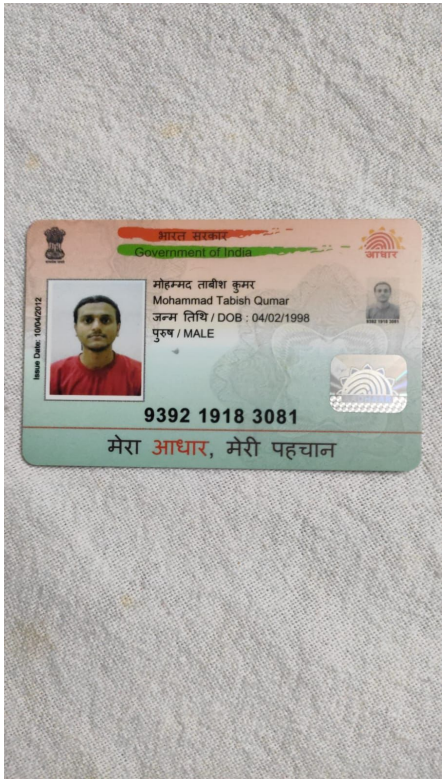
Structured Image



Our Query Image

Sample Test Image 1:

Sample Test Image 2



Methodology [1] (Pre-Processing for Feature Detection)

1. Capture the test image from the path
2. Grayscale the Image
3. Apply Gaussian Blur using kernel of size (5,5)
4. Next we apply thresholding on the blurred image
5. Finding contours using the Threshold
6. Contours with maximum area is selected as the desired image for Aadhar card
7. The background is darkened using the largest contour
8. Apply canny detection in the grayscale version of our new image
9. Hough line transform is used to detect finer edges and find the angle of the bottom-most edge wrt to horizontal axis.
10. Apply rotation to get proper orientation of the image
11. Again find contours for the rotated image
12. Use new contours to crop the image based on bounding rectangle and find corners of the image
13. Image is sharpened using gaussian kernel

Methodology [2] (Feature Detection and Matching)

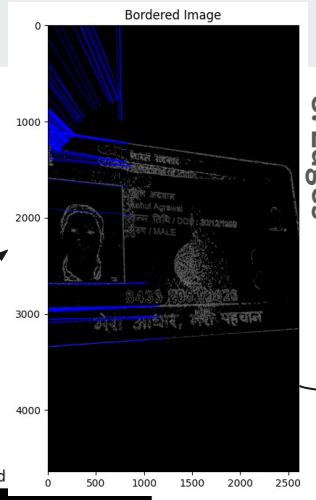
1. Query Image is taken as reference for other images (query image is a perfectly cropped image)
2. 5000 features from the Query Image are detected and used as key-descriptor pairs
3. Test Image features are also detected and saved as key-descriptor pairs
4. Brute-Force matcher is used with Hamming Distance to find the features matches
5. Feature Mapping are done between the descriptors of Query Image and Test Image
6. Best 10% of the matchings are taken and other matches are discarded
7. RANSAC is used to find homography matrix from the source and destination point of the selected matches.
8. The test image is aligned (scale & rotation) with the query image using the Homography matrix found before.
9. A rectangle is formed using the coordinates of Aadhaar Number in Query Image
10. This is used to crop out the area containing Aadhaar Number from test image
11. The text is detected using NLP based PyTesseract Library
12. UIDAI Aadhar Authentication API is used to validate the Aadhar Number

Results [1] (Image Preprocessing)

1. Raw Image

2. Bordered Image

3. Edges



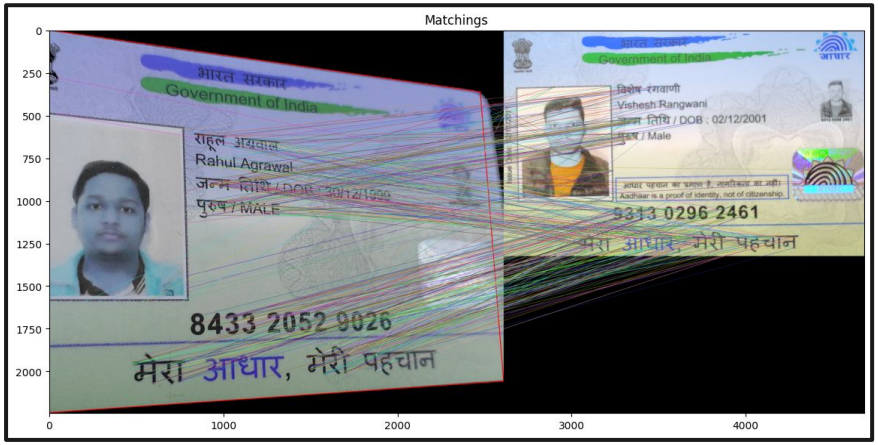
5. Rotated Image

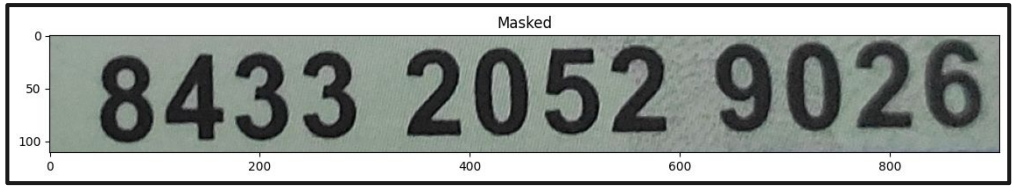
6. Cropped Image

7. Sharpened Image

Results [2] (Feature Extraction)

Feature mapping between Query Image and Test Image





Final extracted aadhar number from Query Image

Detected Features in Query Image

Detected Features in Test Image

Results [3] (Limitation and Failed Cases)

- Specific features from ‘PVC Aadhar Card’ can only be detected. Current implementation fails on long old Aadhar Cards
    - Our Implementation supports only one possible resolution which can just detect the presence of QR code in front of the old design Aadhar Card but cannot read the QR code. We will need to have a separate implementation to support this feature for Long Aadhar cards.
- Different Implementation is required for different types of documents.
- Further details such as name, address, DOB from Aadhar can also be extracted using same approach
- Very low resolution images will fail detection
- Currently implemented in Python Notebook which is not usable on a user level