CY SHIELD

Extract MRZ From Passports

Overview

The goal of this project is to develop a system that can extract Machine-Readable Zones (MRZ) from a set of images. MRZ is a region on passports and identification documents that contains important information like the document holder's name, date of birth, and passport number.
Two methods used to extract MRZ ,which are EasyOCR and Pytesseract.

Data set

The dataset used in this project consists of seven images containing identification documents with MRZ regions.

Task

Write an algorithm to extract machine readable zone "MRZ" in attached images .

Input is image .

Output is a txt file has 2 lines ("MRZ") .

1.Data preprocessing

- **ROI(Region of Interest):**

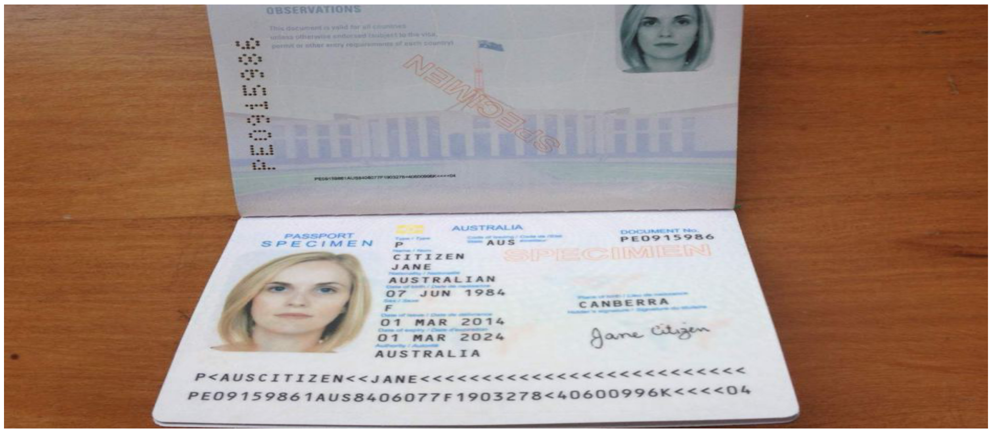
  ROI stands for Region of Interest. In the context of data preprocessing, ROI refers to a specific area or region within an image that contains the relevant information for further analysis or processing.

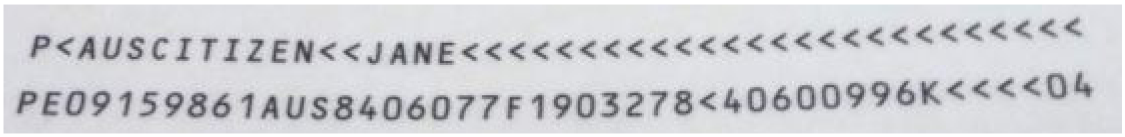
  In the project mentioned, the ROI serves as a data preprocessing step to isolate the Machine-Readable Zone (MRZ) region within the identification document images. The MRZ region contains the important textual information that needs to be extracted using OCR techniques.

The ROI is defined by specifying the starting coordinates **(`x` and `y`)** of a rectangle and its width and height **(`width` and `height`)**. These **coordinates** determine the position and size of the rectangular region within the image that will be extracted.

Using array slicing in the form of `img’, the ROI is cropped from the original image (`img0_bgr`). This operation **extracts** only the pixels within the specified rectangular region, discarding the rest of the image.This process is done on **all images**

Results




P<AUSCITIZEN<<JANE<<<<<<<<<<<<<<<<<<<<<<<<<<
PE09159861AUS8406077F1903278<40600996K<<<<04

EasyOCR

1.The image is upscaled using a specified `upscale_factor`. Upscaling refers to increasing the size of the image by a certain factor to enhance its quality and resolution.

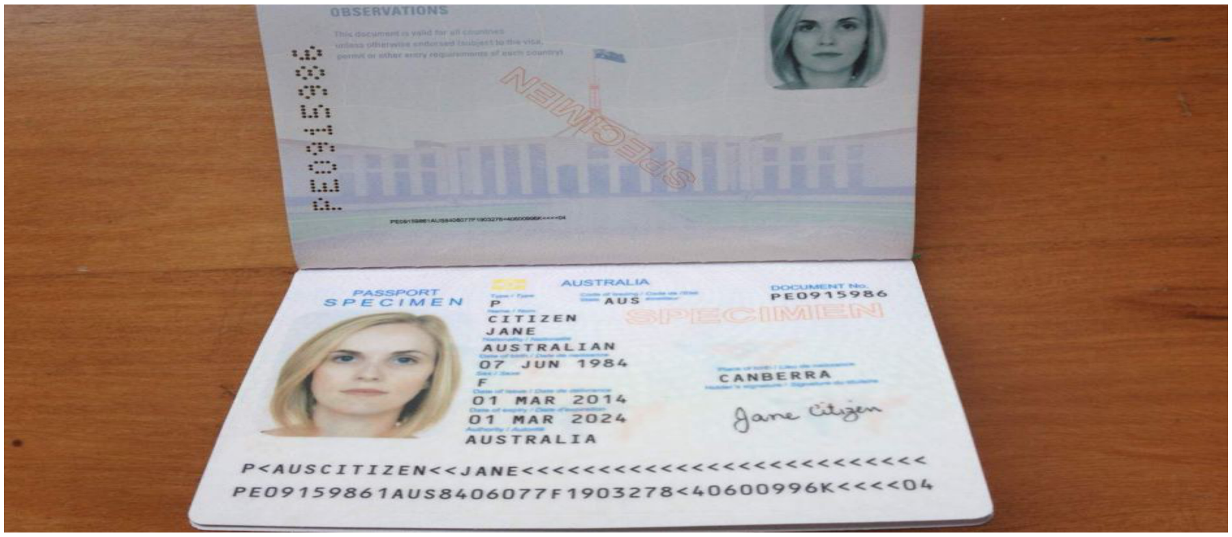
2. The upscaled image is converted to grayscale. Grayscale conversion involves transforming the image from a color representation to a single-channel grayscale representation. This conversion simplifies the image by removing color information, making it easier for optical character recognition (OCR) algorithms to process the image.

3. OCR is performed on the grayscale image using the EasyOCR library. EasyOCR is a deep learning-based OCR tool that can recognize text from images. The library is initialized with the desired language(s) for text recognition.

3. The OCR algorithm analyzes the grayscale image and extracts text lines, including the MRZ lines. MRZ refers to the Machine Readable Zone, which contains important information such as passport details.

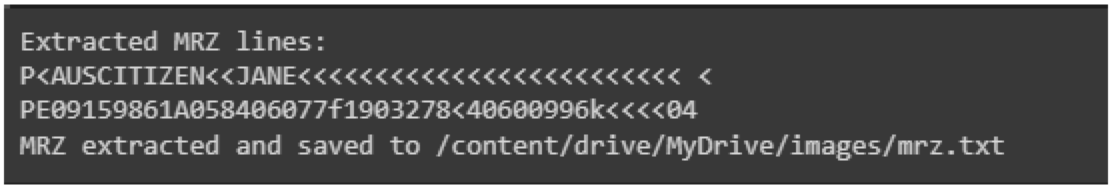
4. The extracted MRZ lines are printed to the console. Then saved to a text file specified by the output path.

EasyOCR Result


OBSERVATIONS
PASSPORT
SPECIMEN
AUSTRALIA
SPECIMEN

P
AUS
PE0915986
CITIZEN
JANE
AUSTRALIAN
07 JUN 1984
F
CANBERRA
01 MAR 2014
01 MAR 2024
AUSTRALIA

```
P<AUSCITIZEN<<JANE<<<<<<<<<<<<<<<<<<<<<<<<<<
PE09159861AUS8406077F1903278<40600996K<<<<04
```



```
Extracted MRZ lines:
P<AUSCITIZEN<<JANE<<<<<<<<<<<<<<<<<<<<<<<<< <
PE09159861A058406077f1903278<40600996k<<<<04
MRZ extracted and saved to /content/drive/MyDrive/images/mrz.txt
```

Other Experiments

In addition to the baseline experiments.I used Tesseract OCR library to compare its performance with EasyOCR. Tesseract is a widely-used OCR engine with robust text recognition capabilities.

Tesseract

1. Grayscale Conversion: The input image region of interest (ROI) `roi_0` is converted from the BGR color space to grayscale using the OpenCV function `cv2.cvtColor()`. This step is performed to simplify subsequent image processing operations.

2. Noise Reduction: Gaussian blur is applied to the grayscale image using the `cv2.GaussianBlur()` function. The blur operation smoothens the image by averaging the pixel values in a local neighborhood. This helps to reduce noise and improve the quality of the image for subsequent processing.

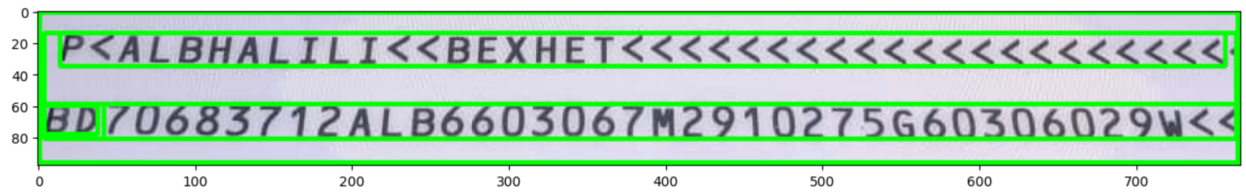
3. White Region Detection: A rectangular structuring element called `rectKernel` is created using `cv2.getStructuringElement()`. This structuring element defines a neighborhood for morphological operations. In this case, it is used for dilation. Dilation is applied to the blurred image using `cv2.dilate()`, which expands the white regions in the image. This step enhances the text regions and makes them more distinguishable.

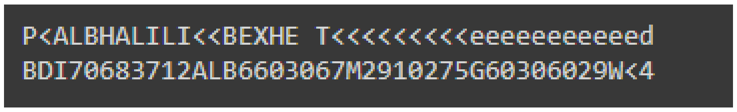
4. Contour Detection: The dilated image is processed using the `cv2.findContours()` function to detect the contours, which are the boundaries of white regions in the image. The `cv2.RETR_TREE` retrieval mode retrieves all the contours and builds a full hierarchy of nested contours. The `cv2.CHAIN_APPROX_SIMPLE` approximation method compresses horizontal, vertical, and diagonal segments and leaves only their end points, resulting in more efficient storage of contour information.

5.OCR and Text Box Extraction: The `pytesseract.image_to_data()` function from the Tesseract OCR library is used to perform OCR on the image with the detected contours. It returns a dictionary containing information about each detected text region, including the coordinates of the bounding boxes around the text.

Text Box Visualization

Bounding rectangles are drawn around each detected text box using `cv2.rectangle()`. The rectangles are drawn on a copy of the original ROI image using





```
P<ALBHALILI<<BEXHE T<<<<<<<<<<eeeeeeeeeeeed
BDI70683712ALB6603067M2910275G60306029W<4
```

Overall Conclusion:

Conclusion

In conclusion, the project aimed to extract MRZ information from identification document images using OCR techniques. Baseline experiments using EasyOCR showed promising results, but further experiments using Tesseract OCR provided additional insights.However Tesseract OCR MRZ extractions was not very accurate compared to EasyOCR .

Tools Used:

1. Python programming language
2. OpenCV library for image processing
3. PIL (Python Imaging Library) for image handling
4. EasyOCR library for OCR using pre-trained models
5. Tesseract OCR library for OCR with advanced text recognition capabilities

Challenges Faced:

The biggest challenge faced during this project was the variability in image quality and text characteristics. Some images had low resolution, noise, or distorted text, which affected the accuracy of OCR. Adjusting preprocessing techniques and OCR parameters were necessary to improve results.

Key Learnings:

Familiarity with OCR techniques and libraries like EasyOCR and Tesseract OCR.