PDF-XTRACTION ON STRUCTURED PDFS

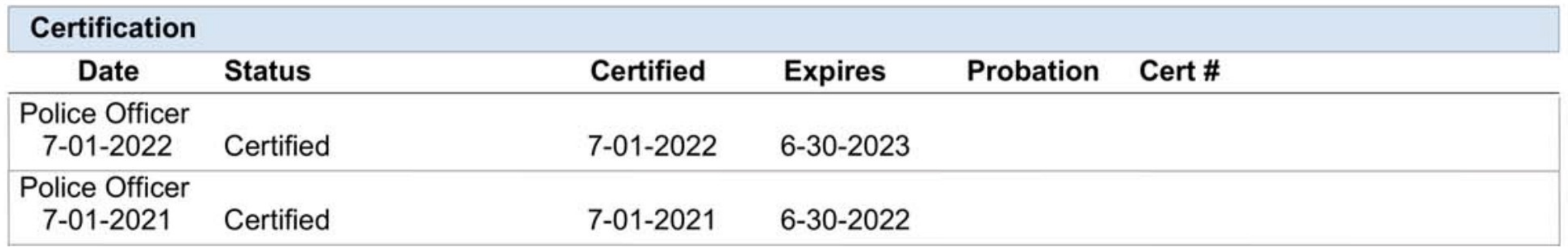
MAWIL HASAN, ADITYA PARAMESWARAN, ALVIN CHEUNG

Background

The Maryland Police Profile PIA PDF is composed of Images – scanned documents, composing of every Police Officer’s certification and employment history within their respect department in the state.

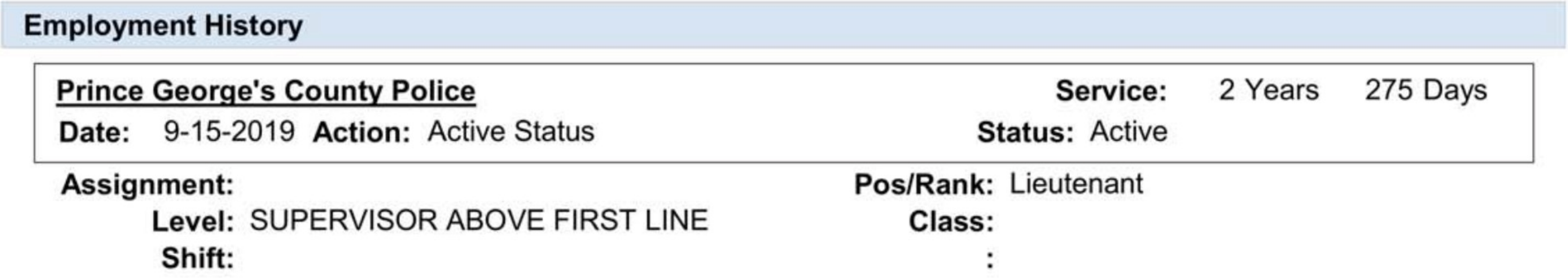
The certification table and employment history table per officer are structurally formatted, such that Certification and Employment History are headers to their tables and each reach is unique formatted.

Idea: Extract all the certification records and employment history records associated with each Police Officer using OCR and data cleaning techniques



Certification

Date	Status	Certified	Expires	Probation	Cert #
Police Officer 7-01-2022	Certified	7-01-2022	6-30-2023		
Police Officer 7-01-2021	Certified	7-01-2021	6-30-2022		

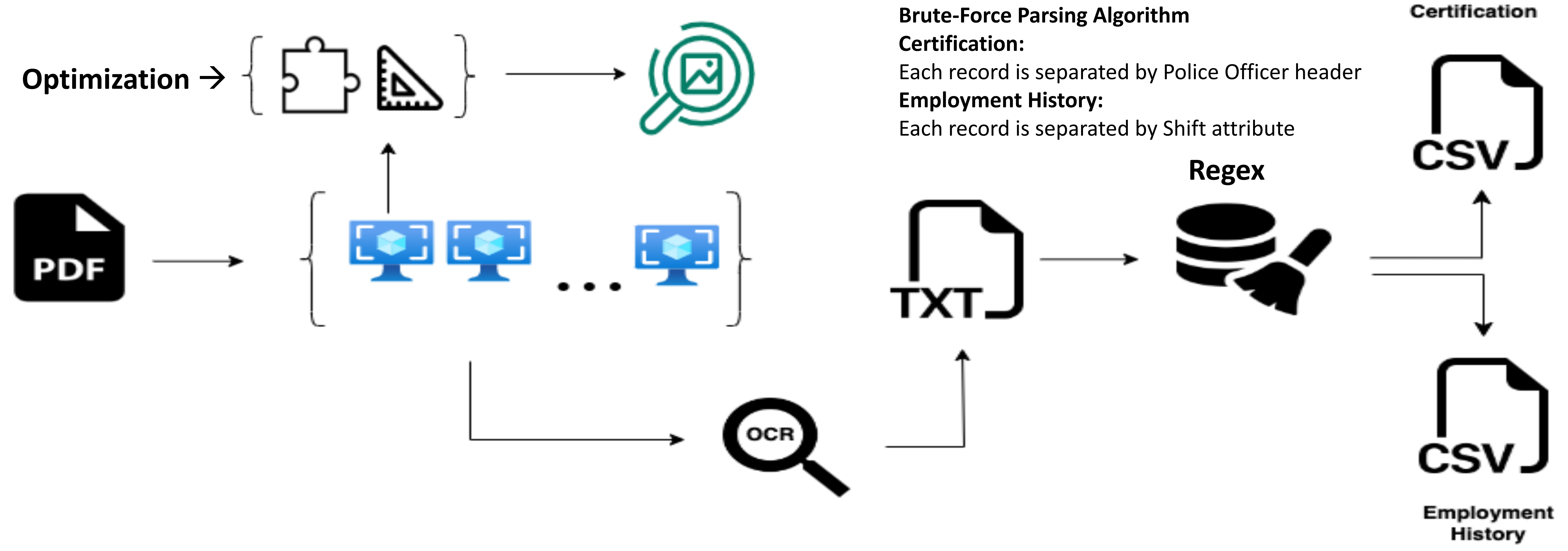


Employment History

Prince George's County Police **Service:** 2 Years 275 Days
Date: 9-15-2019 **Action:** Active Status **Status:** Active

Assignment: **Pos/Rank:** Lieutenant
Level: SUPERVISOR ABOVE FIRST LINE **Class:**
Shift: :

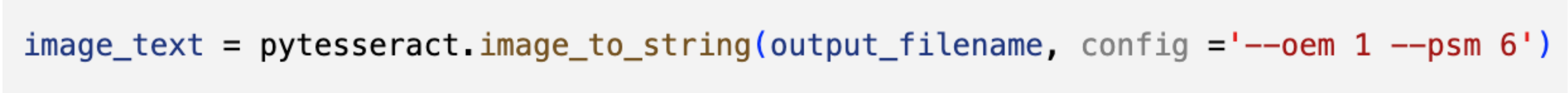
Design Architecture



Brute-Force Parsing Algorithm
Certification:
Each record is separated by Police Officer header
Employment History:
Each record is separated by Shift attribute

Optical Character Recognition(OCR)

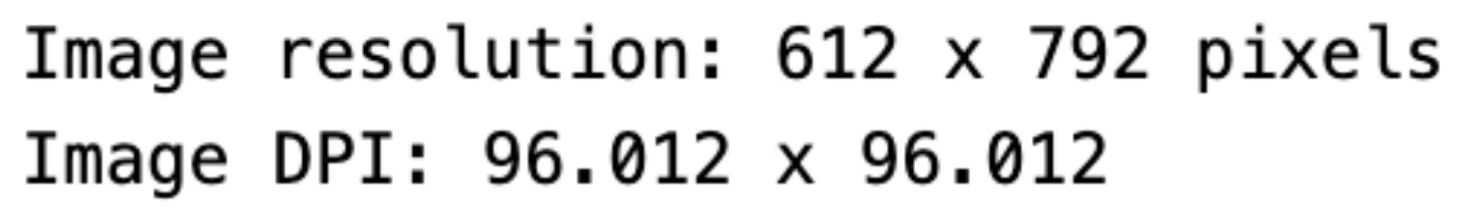
Using **Pytesseract**, a Python wrapper for Google’s Tesseract-OCR engine to extract text from PDF:
image_to_string(image_filename, config)
image_filename – filename to run OCR on
config – changing the behavior of the Tesseract OCR in how it reads text



```
image_text = pytesseract.image_to_string(output_filename, config ='--oem 1 --psm 6')
```

Notable Challenges
Pytesseract is error-prone. The following factors were discovered when converting every page in a PDF to ‘png ‘ format using **PDF2Image**, python package.

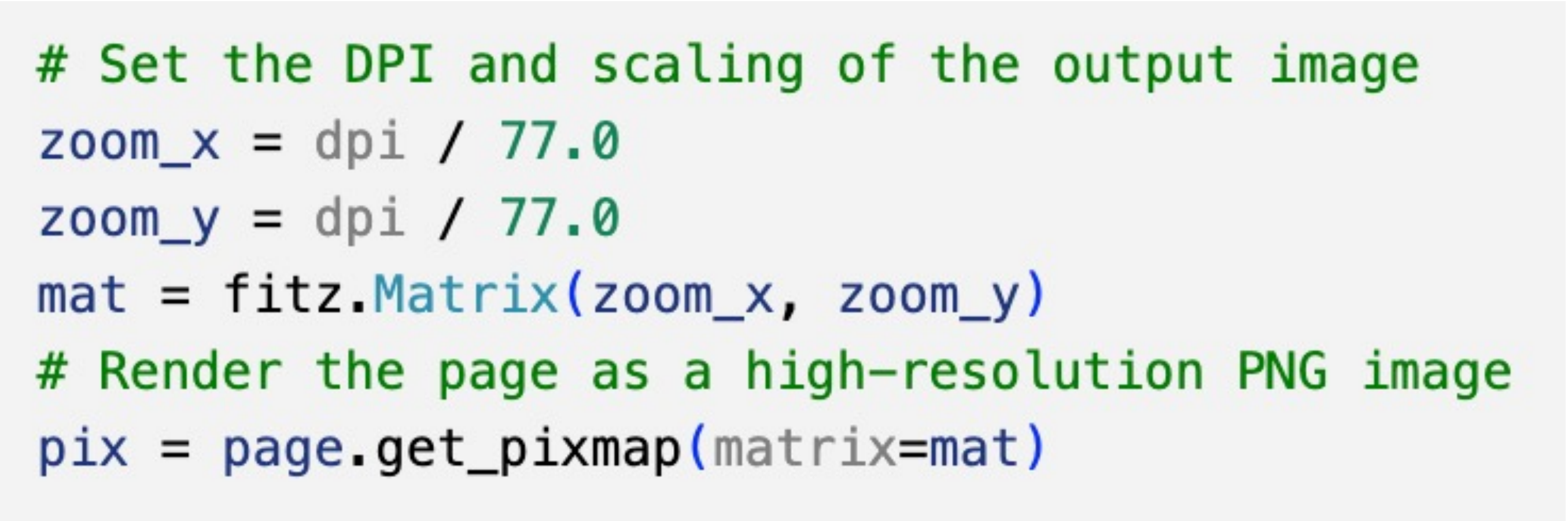
1. Poor Resolution
2. Poor Font Style
3. Conversion to ‘png’ format took long processing times ~ Averaged around 6 seconds per page



```
Image resolution: 612 x 792 pixels
Image DPI: 96.012 x 96.012
```

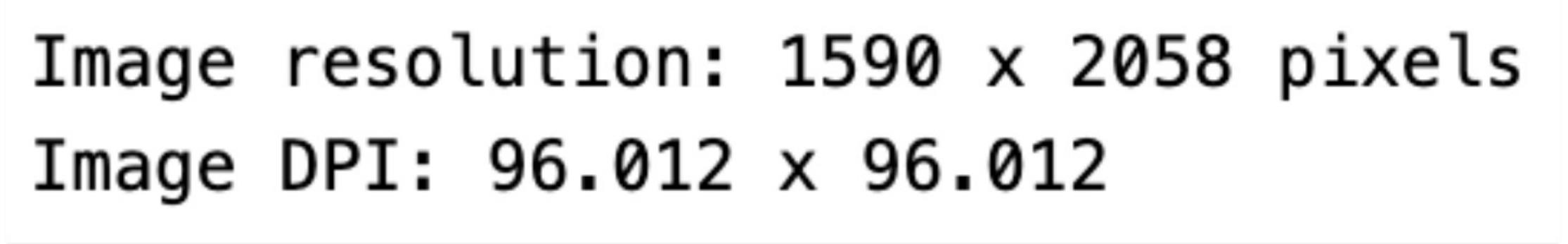
Objective: Maximize resolution rescaling every image to produce an accurate OCR scan on every page of our scanned PDF document and optimize processing time for ‘png’ format conversion

Optimization – PyMUPDF

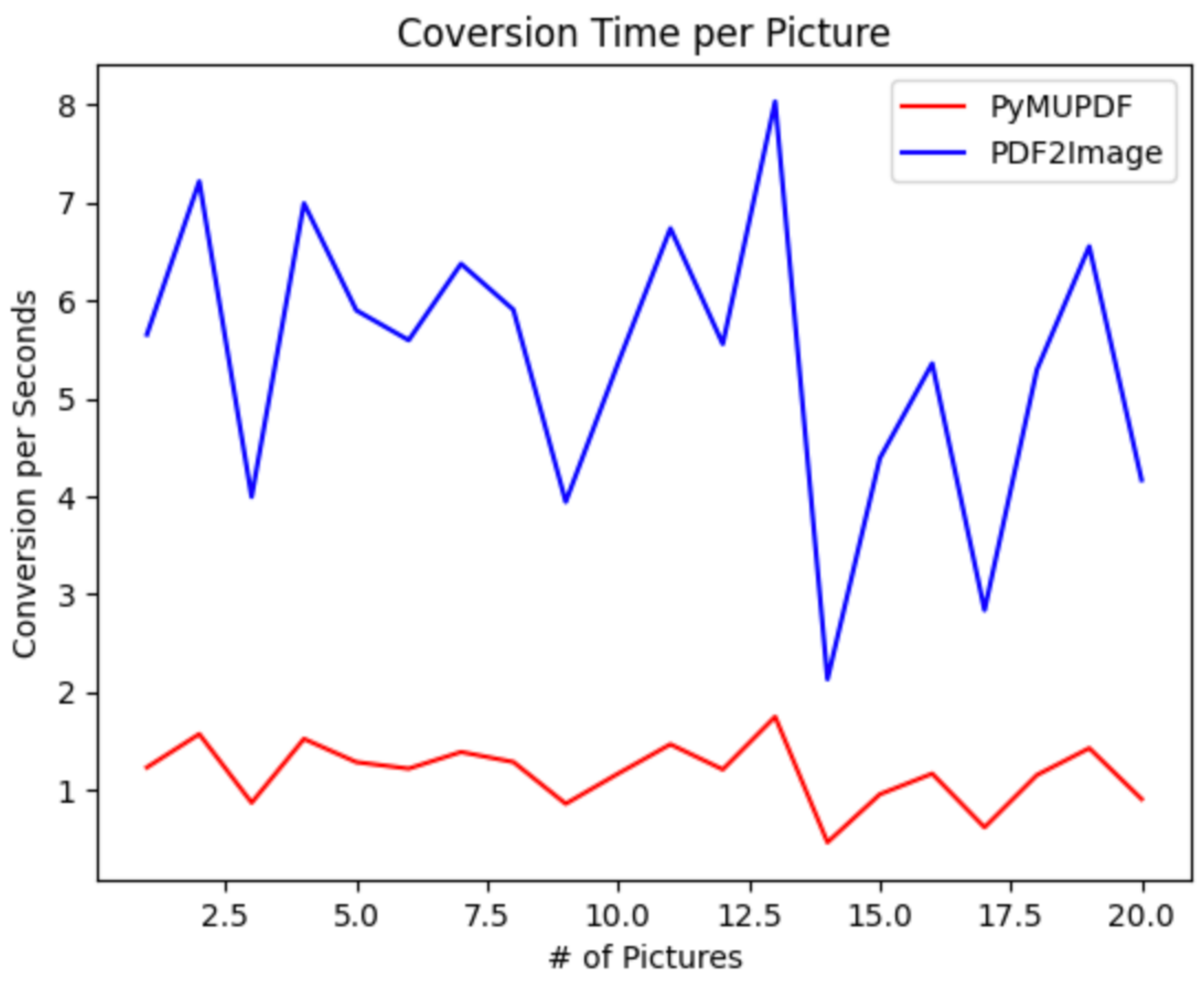


```
# Set the DPI and scaling of the output image
zoom_x = dpi / 77.0
zoom_y = dpi / 77.0
mat = fitz.Matrix(zoom_x, zoom_y)
# Render the page as a high-resolution PNG image
pix = page.get_pixmap(matrix=mat)
```

→



```
Image resolution: 1590 x 2058 pixels
Image DPI: 96.012 x 96.012
```



```
The Average Time to Process is: 1.1735363693558611
```

Conversion rates decreased by nearly **80%**.

Processing larger PDF datasets will take a shorter time vs PDF2Image, while maintaining a **100%** accuracy for pytesseract recognizes every necessary character in the PDF to differentiate records in the Certification and Employment History table