OCR with Python: Extracting Text from PDFs

A

Aman dubey · Follow

3 min read · Apr 15, 2024

Listen Share More

Optical Character Recognition (OCR) is a technology that enables computers to extract text from images or scanned documents. This is a valuable capability as it allows us to process and analyze large volumes of data without having to manually transcribe the content. In this article, we will explore how to perform OCR on PDF files using Python.

Python Libraries for PDF OCR:
To perform OCR on PDF files, we will utilize the following Python libraries:

pytesseract:
pytesseract is a Python wrapper for the powerful Tesseract OCR engine. It supports many languages and can recognize various text formats, including alphanumeric characters and symbols.

pdf2image:
pdf2image is a Python library that converts python pdf ocr files into a sequence of images. This library will convert each page of the PDF file into an image, which can then be processed by pytesseract.

pip install pytesseract pdf2image

After the installation is complete, we can proceed with extracting text from PDF files.

Step 1: Converting PDF to Images:
Before we can perform OCR on a PDF file, we need to convert each page of the PDF into

```
def convert_pdf_to_images(pdf_path):
pages = convert_from_path(pdf_path)
for i, page in enumerate(pages):
page.save(f"page_{i}.jpg", "JPEG")
```

In the above code snippet, the `convert_from_path` function is used to convert the PDF file into a list of PIL (Python Imaging Library) image objects. Each image corresponds to a single page of the PDF. We then save each image to disk using the `save` method.

Step 2: Performing OCR on Images:
Once we have converted the PDF pages into images of rest api s3, we can apply OCR using the pytesseract library. Before running the OCR, make sure you have Tesseract installed on your system. Instructions for installing Tesseract can be found on the Tesseract GitHub page (https://github.com/tesseract-ocr/tesseract).

```
import pytesseract
from PIL import Image
def perform_ocr_on_images():
for i in range(num_pages):
image_path = f"page_{i}.jpg"
image = Image.open(image_path)
text = pytesseract.image_to_string(image)
print(f"Page {i + 1}:\n{text}\n")
```

In the above code snippet, we loop through each image file and open it using the `Image.open()` method from the PIL library. We then pass the image object to `pytesseract.image_to_string()` to extract the text. The resulting text is printed for each page. python ocr pdf for further processing or analysis. We can achieve this by modifying our previous code as follows:

```
import pytesseract

from PIL import Image

def perform_ocr_on_images():

combined_text = “”

for i in range(num_pages):

image_path = f”page_{i}.jpg”

image = Image.open(image_path)

text = pytesseract.image_to_string(image)

combined_text += text + “\n”

with open(“extracted_text.txt”, “w”) as f:

f.write(combined_text)

print(“Extraction complete. Text saved to ‘extracted_text.txt’”)
```

In the above code snippet, python ocr pdf we initialize an empty string called `combined_text`, which will store the text extracted from each page. We concatenate the text from each page to the `combined_text` variable. Finally, we write the `combined_text` to a text file called `extracted_text.txt`.

In this article, we explored how to perform OCR on PDF files using Python. We used the pytesseract library to extract text from images, generated from PDF pages using the pdf2image library. With the ability to extract text from PDFs, we can now automate text processing tasks and efficiently analyze large volumes of data.

Python Pdf Ocr

Follow

Written by Aman dubey

3 Followers

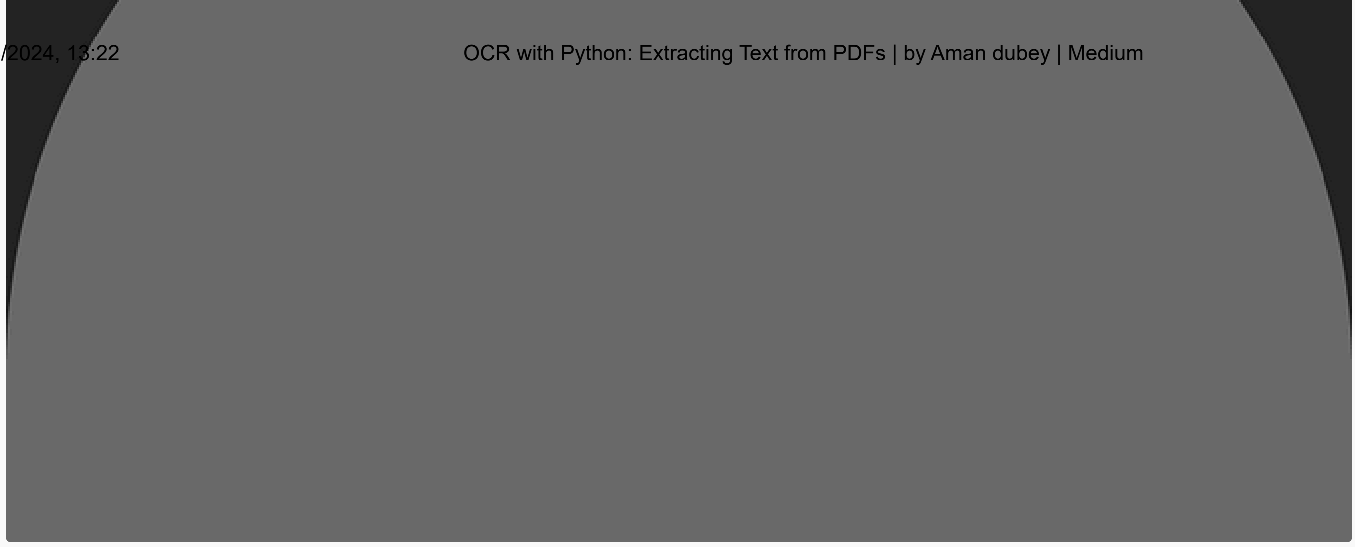
More from Aman dubey

Aman dubey

The Ultimate Guide To Swift Game Development: Building Your First Game

Hey there! Are you ready to dive into the exciting world of Swift game development? In this article, we will explore the basics of building...

9 min read · Apr 10, 2024



Aman dubey

Using Top Level Await In Typescript: Simplifying Asynchronous Javascript Code

TypeScript code where you needed to use the “await” keyword outside of an asynchronous function? TypeScript 3.8 introduced a new feature...

6 min read · Apr 12, 2024

Aman dubey

3 min read · Apr 9, 2024

Aman dubey

Comprehensive Guide to Linux Firewalls: iptables, nftables, ufw, and firewalld

In the dynamic landscape of network security, firewalls play a pivotal role in fortifying systems against potential threats. Within the…

4 min read · May 8, 2024

See all from Aman dubey

Recommended from Medium

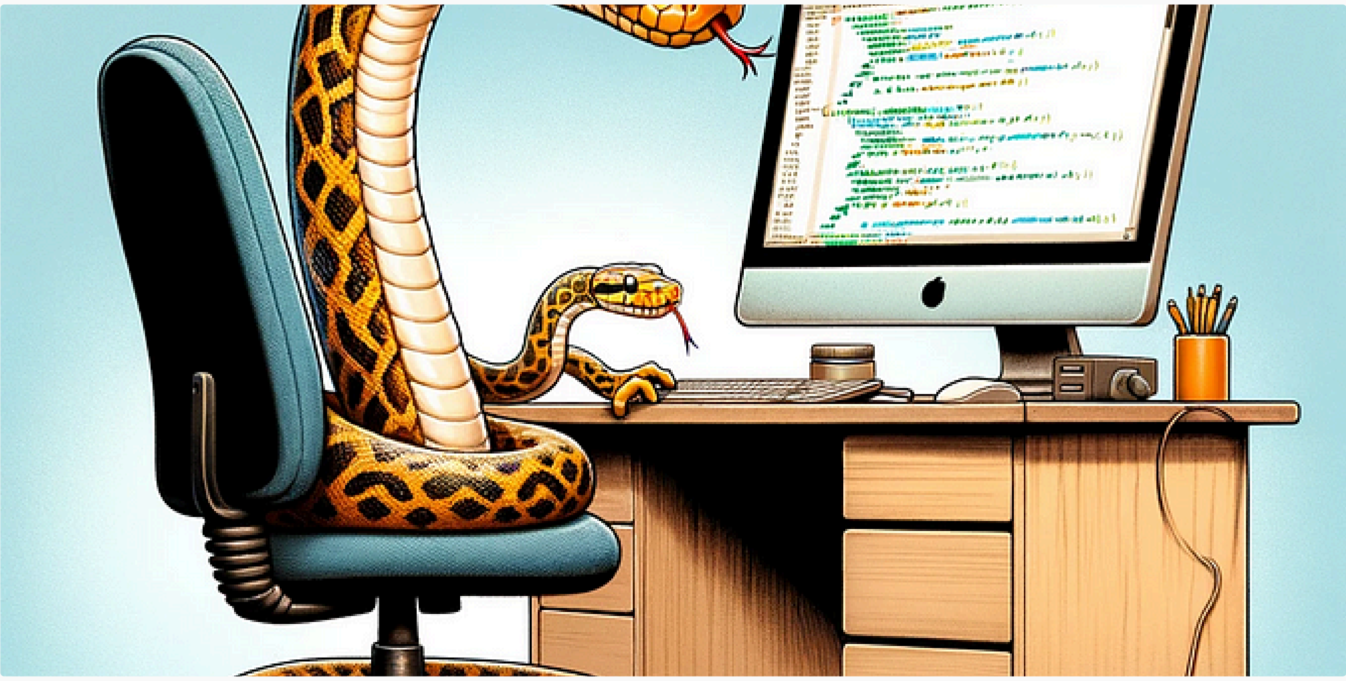
Alex Nadein

How to build Optical Character Recognition fast with Python

What is OCR (Optical Character Recognition)?

3 min read · Mar 12, 2024

100



Eivind Kjosbakken in Towards AI

How To Run Your Fine-Tuned EasyOCR Model In Python

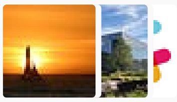
✦ · 5 min read · Jan 21, 2024

14

Lists

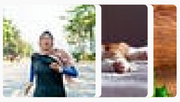
Staff Picks

671 stories · 1086 saves



Stories to Help You Level-Up at Work

19 stories · 663 saves

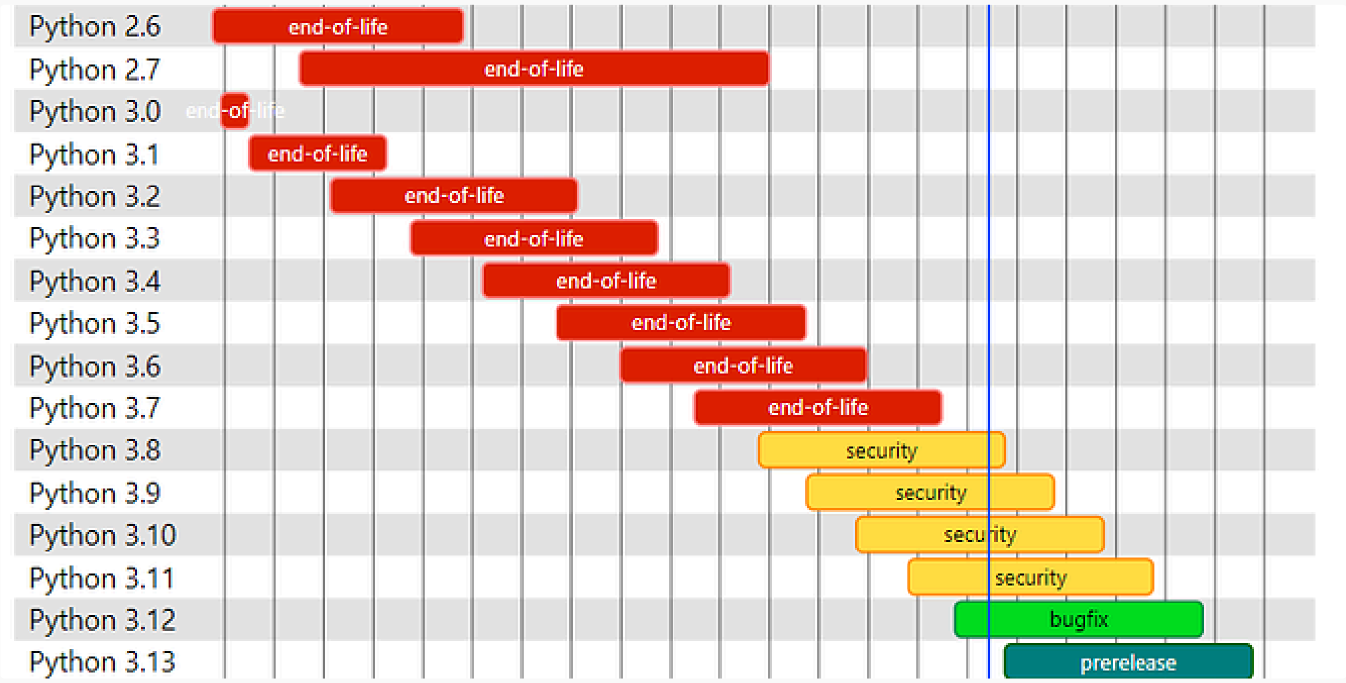


Self-Improvement 101

20 stories · 2175 saves

Productivity 101

20 stories · 1937 saves



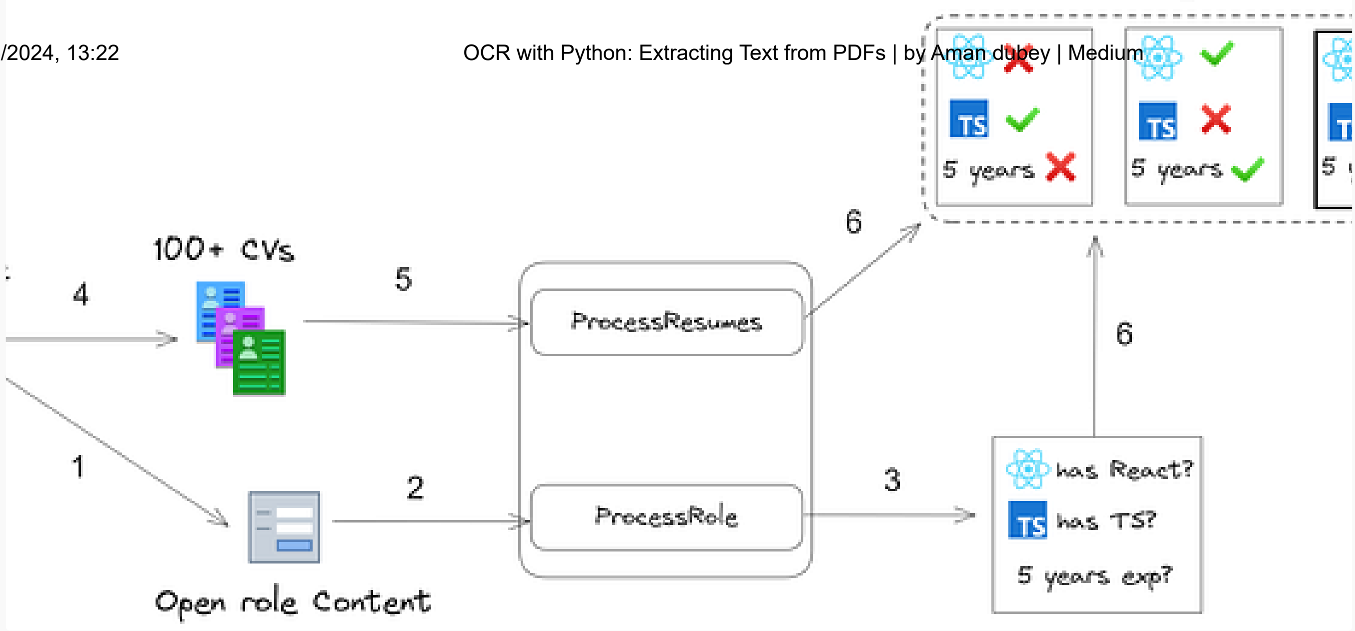
Gabe in Level Up Coding

Speculating on Python 4.0: Could These 12 Beloved Features Disappear?

Explore Potential Changes in Python 4.0 and Their Possible Impact on Your Code

✦ · 9 min read · May 17, 2023

2.1K 13



Júlio Almeida

How Companies Use LLMs to Process 4,000 CVs for $1 | ExtractThinker

Automate CV screening with ExtractThinker: Efficiently process 4,000 resumes for $1 using LLMs, reducing recruiter workload

· 8 min read · May 27, 2024

118 1

Classification, Parsing and…

Donut model is an innovative solution that eliminates the need for traditional OCR engines, offering an efficient end-to-end solution.

4 min read · Feb 21, 2024

7

Hein Burgmans

Which Python library is the best for extracting text from PDFs?

The Case In my current job, I receive numerous PDFs daily for various purposes. Customers send requests for quotations and sales orders…

7 min read · Feb 26, 2024

See more recommendations