Poppler On Windows

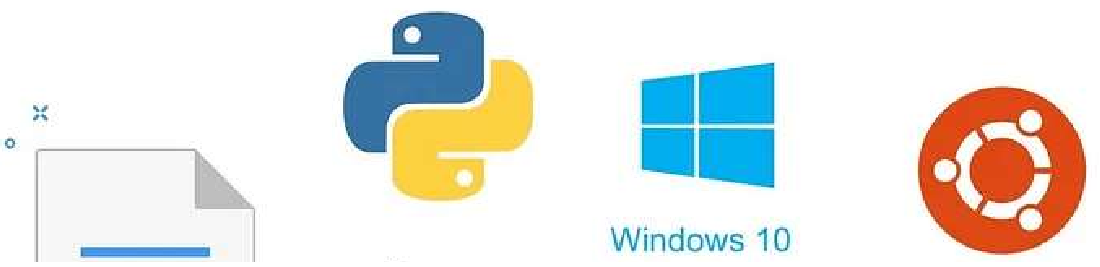
Python, PDFs, and Window’s Subsytem for Linux

Matthew Earl Miller · Follow

Published in Towards Data Science

4 min read · Jan 9, 2020

Listen Share



Poppler On Windows

Intro:

Portable Document Format (PDFs) are everywhere and importing a popular python-package like PDF2Image, PDFtoText, or PopplerQt5 is a common approach to dealing with them. Unfortunately, unless you are working with a Linux machine, many users are reporting that these packages are returning errors because they rely on Poppler.

Never heard of Poppler?

Poppler is a utility for rendering PDFs and it’s common to Linux systems, but not Windows. So, naturally, if you want to use Poppler and its associated packages, we need to bridge the gap.

Let’s visit google and see what our options are…

A quick Google (StackOverflow) search reveals that there are many other people having this problem and they are still looking for solutions.

- PDF to JPG (Poppler)
- Install Poppler on Windows?
- Cannot Install ‘PDFtoText’ on Windows (Poppler)
- Running PyPDFOCR on Windows — Requires Poppler?
- ModuleNotFoundError — No Module Named ‘SipDistUtils’ (Poppler)

The Problem:

Poppler and Python’s PDF-libraries, which leverage Linux-utilities, don’t play well with Windows.

When we look for solutions, many of them are outdated, ineffective, too difficult, etc…

The Solution:

Of the purposed solutions, one solution appears to work well.

Windows Subsystem for Linux (WSL).

Actually, because of how powerful Windows Subsystem for Linux is, it’s a great solution for other problems which require Linux tools on a Windows machine.

So, what is WSL?

Windows Subsystem for Linux is a compatibility layer for Linux binary executables natively on Windows 10. It recently entered version two (WSL 2) and introduced a real Linux kernel. To put it plainly, WSL makes it feel like you’re working on a real Linux machine (and you are).

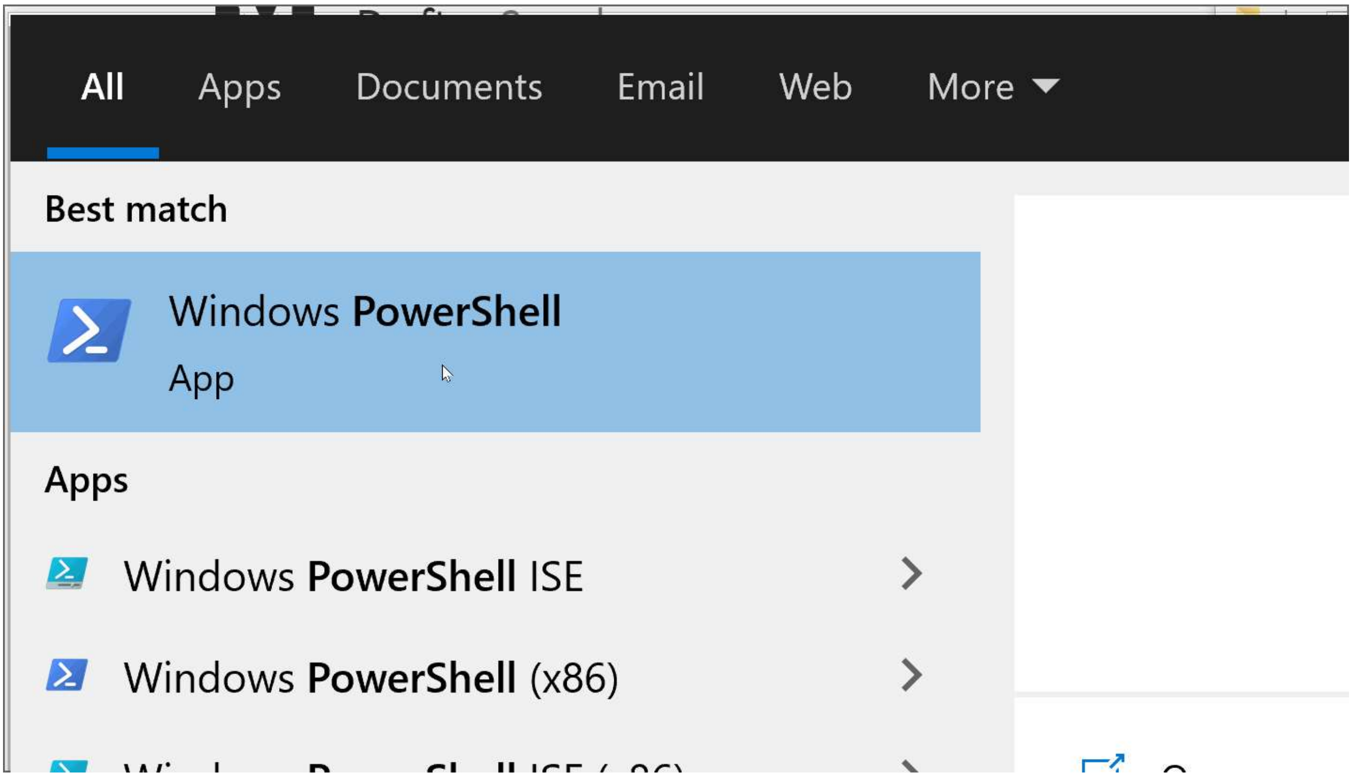
Installation and Usage Guide — WSL

Reference

In this section, we will, in five short steps, install and setup WSL. Afterwards, we will install and setup Poppler in a few short steps.

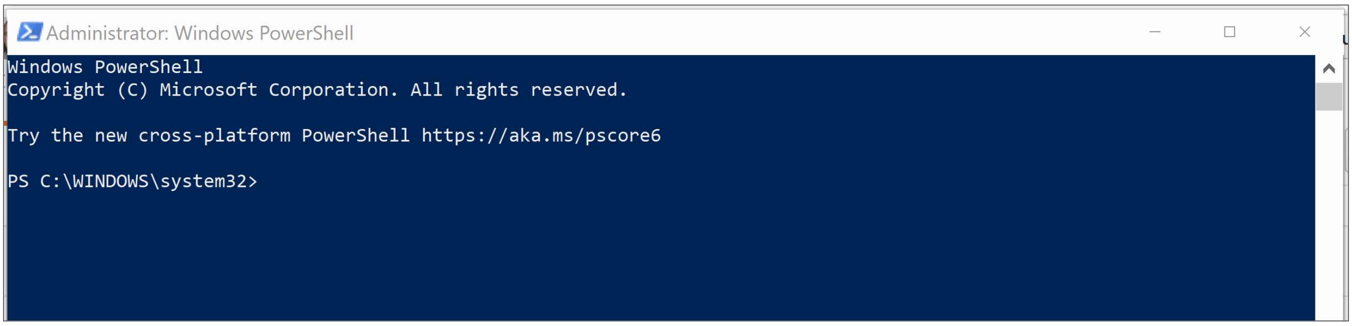
Step 1:

Run Window's Powershell as an administrator.



Step 2:

Enable WSL by executing the 'Enable-WindowsOptionalFeature' command:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32>
```

Enabling WSL

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

Windows_Enable_WSL hosted with ❤ by GitHub view raw

Step 3:

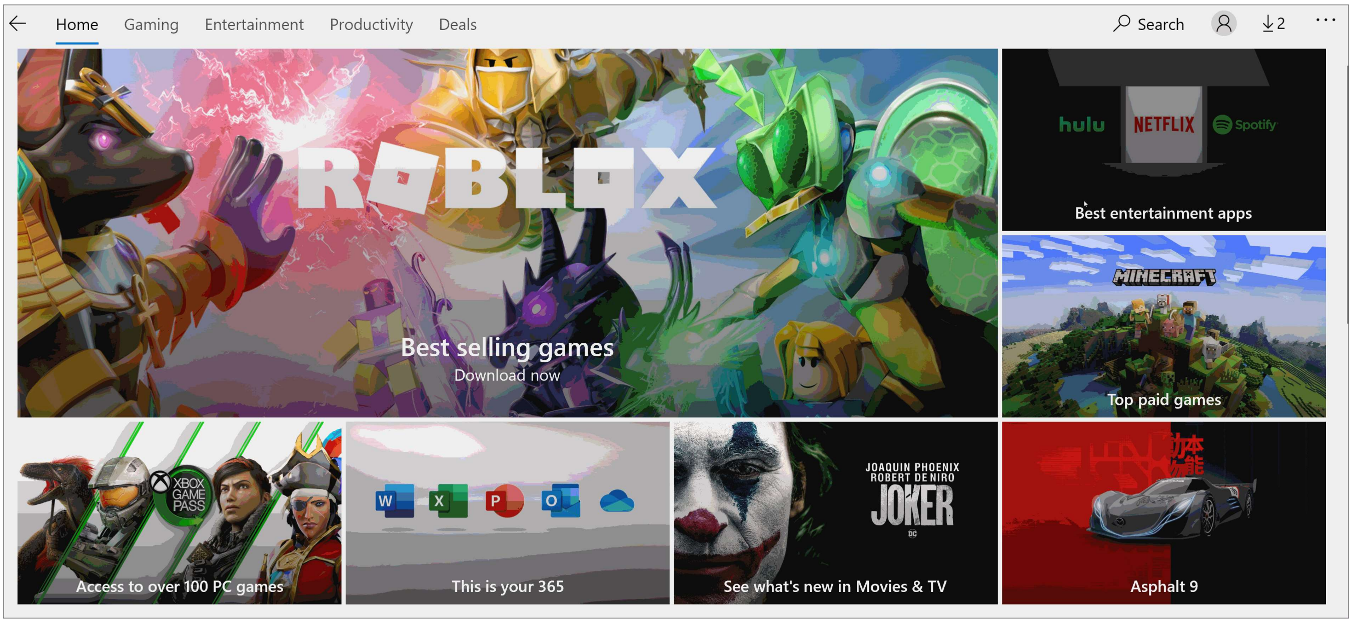
Activate the changes by restarting your computer.

Note that, Microsoft says, “**This reboot is required** in order to ensure that WSL can initiate a trusted execution environment.”

Step 4:

Now, you’re back from a restart, your system’s WSL is enabled, and you are ready to install a Linux distribution.

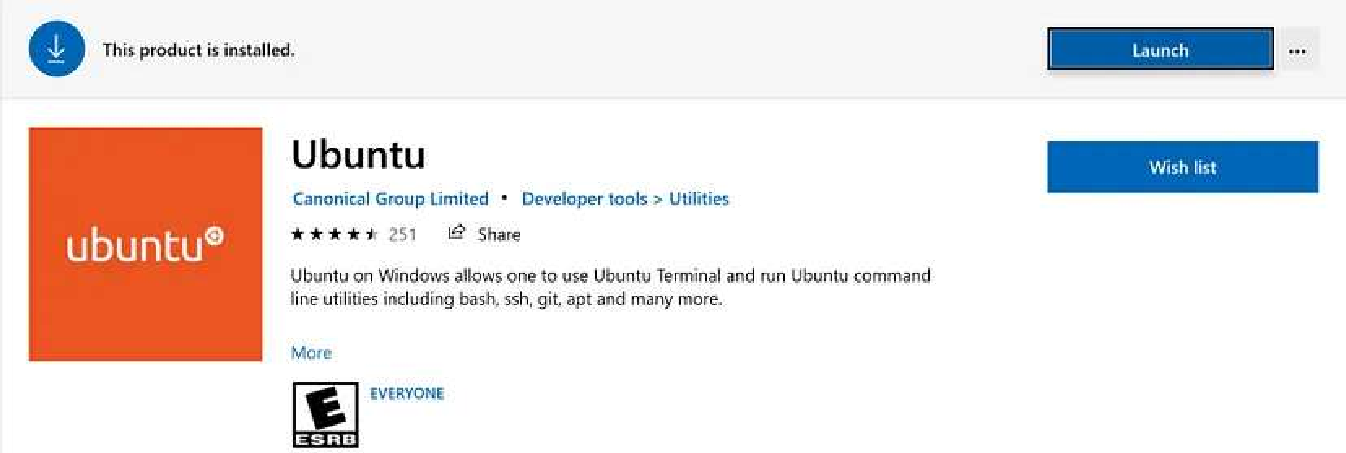
Go to the Window’s Store and search for WSL.



Getting WSL from Windows Store

Step 5 (final):

Click Ubuntu and choose to install. Note, mine is already installed, so you have to do some imagining here.


This product is installed.

Ubuntu

Canonical Group Limited • Developer tools > Utilities

★★★★★ 251 Share

Ubuntu on Windows allows one to use Ubuntu Terminal and run Ubuntu command line utilities including bash, ssh, git, apt and many more.

More

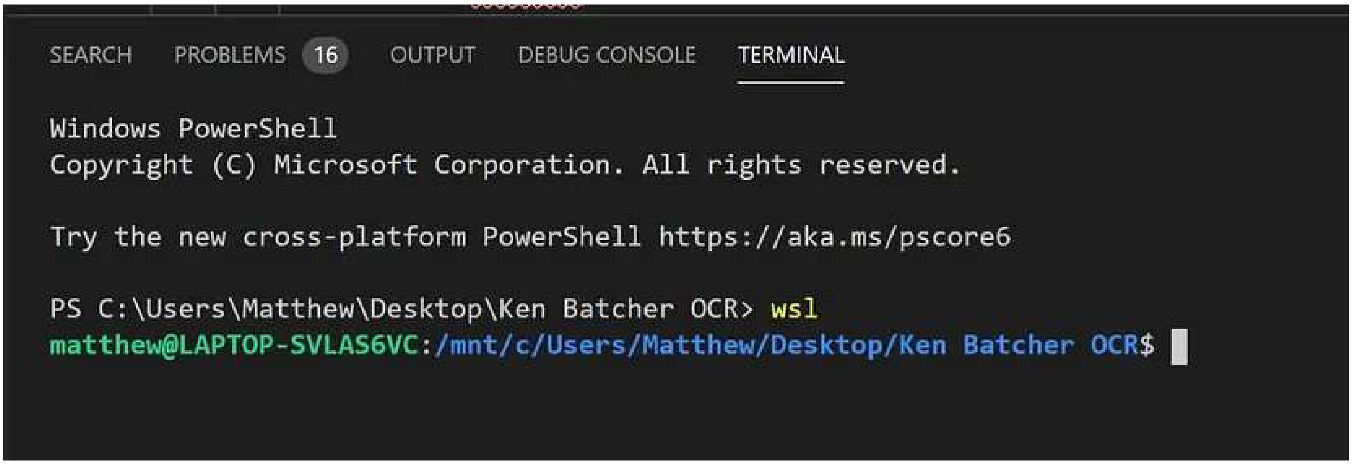
EVERYONE

Installing Ubuntu

Installation and Usage Guide — Poppler:

Step 1:

Enter WSL through a terminal like this one in VS Code. Notice that, once you enter WSL, the terminal prompt will change. You are now operating within a Linux machine! Exciting!



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Matthew\Desktop\Ken Batcher OCR> wsl
matthew@LAPTOP-SVLAS6VC:/mnt/c/Users/Matthew/Desktop/Ken Batcher OCR$
```

Enter WSL

Step 2:

Conduct the following commands within the WSL-prompt. Note that, you can ignore some of the steps that deal with Tesseract-OCR and PyTesseract. These are for the demo-project which I share at the end of the article.

```
# Author:  Matthew E. Miller
# Date: 1/1/2020
# Medium: https://medium.com/@matthew_earl_miller (where this is being published)
# Github: https://github.com/matmill5
# Linkedin: https://www.linkedin.com/in/matthew-miller-engineer/
# StackOverflow: https://stackoverflow.com/users/11937169/matthew-e-miller?tab=profile

# Command 1: Enter Windows Subsystem for Linux
PS C:\Users\Matthew\Desktop\Project> wsl

# Command 2: Cleanup
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ sudo apt-get clean

# Command 3: Update
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ sudo apt-get update

# Command 4: Get Python 3 on your WSL
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ sudo apt install python3

Command 5: Get Python PIP
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ sudo apt install python-pip

Command 6: Get poppler-utils
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ sudo apt install poppler-utils

Command 7: Get pdf2image (dependant on poppler and inspiration for article)
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ pip install pdf2image

Command 8: Get pathlib
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ pip install pathlib

Command 9: Get pytesseract (if you're doing OCR)
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ pip install pytesseract

Command 10: Get tesseract-ocr (if you're doing OCR)
user@device_name:/mnt/c/Users/Matthew/Desktop/Project$ sudo apt-get install tesseract-ocr
```

Poppler_WSL_Commands hosted with ❤ by GitHub view raw

Step 3 — Testing (final):

Run a program with your newly acquired, ready-to-use, Poppler utilities.

I've created this **demo script**, so you can use it if you don't have your own. **Although**, you will need a PDF to mess with.

```
# Tesseract OCR
import pytesseract
from PIL import Image
import sys
from pdf2image import convert_from_path
import os
import io

# If you need to assign tesseract to path
# pytesseract.pytesseract.tesseract_cmd = r'C:\Users\Matthew\AppData\Local\Tesseract-OCR\tesser

pdf_path = 'pdfs/A Production Implementation of an Associative Arran Processor -STARAN - Rudolp
output_filename = "results.txt"
pages = convert_from_path(pdf_path)
pg_cntr = 1

sub_dir = str("images/" + pdf_path.split('/')[-1].replace('.pdf','')[0:20] + "/")
if not os.path.exists(sub_dir):
    os.makedirs(sub_dir)

for page in pages:
    if pg_cntr <= 20:
        filename = "pg_"+str(pg_cntr)+'_'+pdf_path.split('/')[-1].replace('.pdf','.jpg')
        page.save(sub_dir+filename)
        with io.open(output_filename, 'a+', encoding='utf8') as f:
            f.write(unicode("======================================================== PAGE " +
            f.write(unicode(pytesseract.image_to_string(sub_dir+filename)+"\n"))
            f.write(unicode("======================================================== =========
        pg_cntr = pg_cntr + 1
```

Demo Poppler On Windows OCR hosted with ❤ by GitHub view raw

This code works by converting a PDF to JPG. Then, it conducts OCR and writes the OCR-results to an output-file.

Conclusion:

That's it. You are certified Poppler-On-Windows.

Enjoy the spoils of war! You have gained some seriously new and powerful skills. You are well on your way to becoming a more flexible developer (if you aren't already).

Newly Acquired Skills:

- Ability to successful manipulate PDFs with Python.

- Access to PDF2Image, PDFToText, or other Poppler-utils.
- Windows Subsystem for Linux. ** A seriously powerful dev-tool **

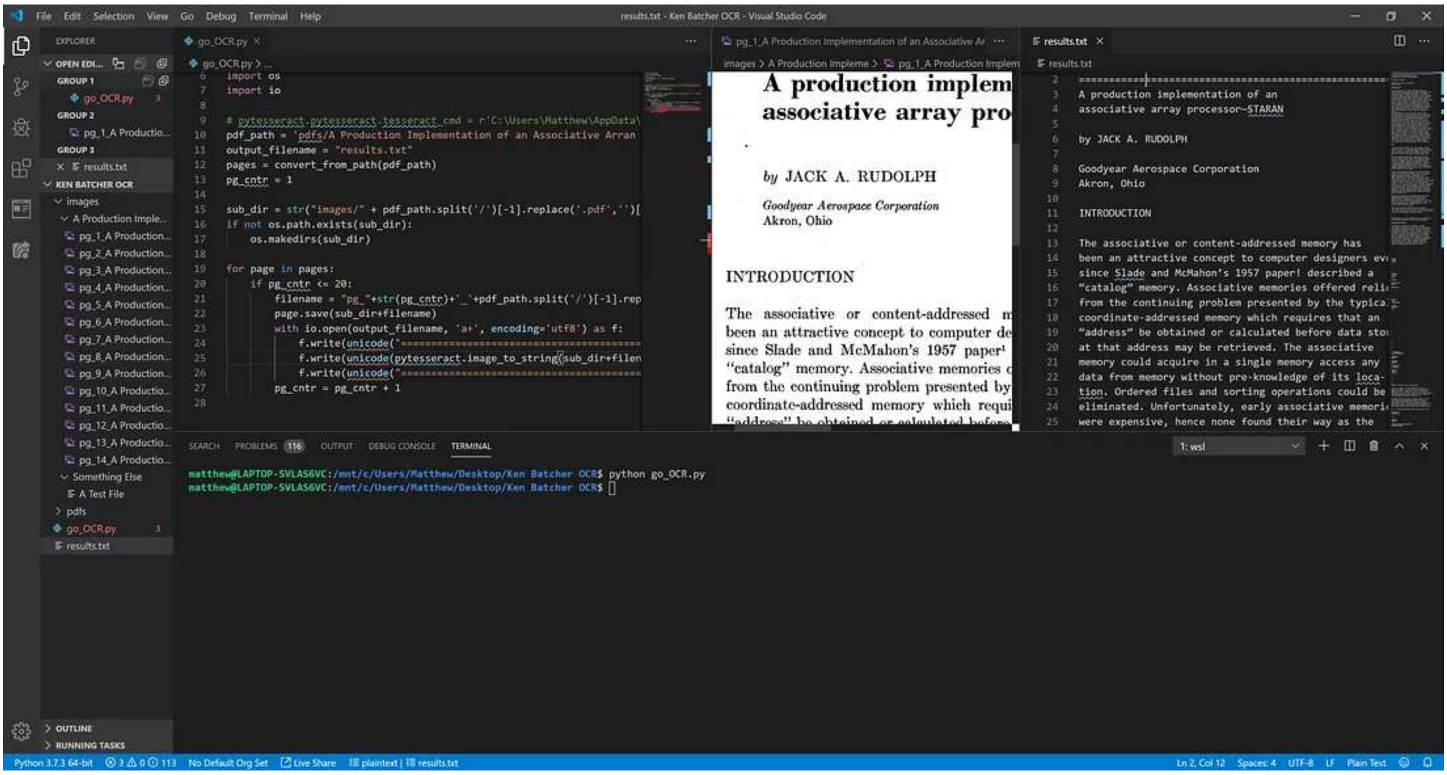
Now What… What Can You Build?

It's so important to experiment with these new skills and solidify your understanding. True understanding comes with experience.

My Poppler-On-Windows Project:

I built an OCR application to help document the historical work of emeritus professor and famous computer scientist, Dr. Kenneth E. Batcher. It uses a PDF to image tool for JPEG-conversion. Then, it does OCR on the image and writes the results to an output file. Since this proof of concept works well enough, it'll eventually be used on document-scans instead of PDFs.

You can find the project here.



File Edit Selection View Go Debug Terminal Help
results.txt - Ken Batcher OCR - Visual Studio Code
go_OCR.py
pg_1_A Production Implementation of an Associative Ar
results.txt
EXPLORER
OPEN EDITORS
GROUP 1
GROUP 2
GROUP 3
KEN BATCHER OCR
images
pdfs
SEARCH PROBLEMS 116 OUTPUT DEBUG CONSOLE TERMINAL
1: wsl
Python 3.7.3 64-bit
No Default Org Set
Live Share
Ln 2, Col 12 Spaces: 4 UTF-8 LF Plain Text

```python
import os
import io

# pytesseract.pytesseract.tesseract_cmd = r'C:\Users\Matthew\AppData\
pdf_path = 'pdfs/A Production Implementation of an Associative Arran
output_filename = "results.txt"
pages = convert_from_path(pdf_path)
pg_cntr = 1

sub_dir = str("images/" + pdf_path.split('/')[-1].replace('.pdf','')[
if not os.path.exists(sub_dir):
    os.makedirs(sub_dir)

for page in pages:
    if pg_cntr <= 20:
        filename = "pg_"+str(pg_cntr)+'_'+pdf_path.split('/')[-1].rep
        page.save(sub_dir+filename)
        with io.open(output_filename, 'a+', encoding='utf8') as f:
            f.write(unicode("==========================================
            f.write(unicode(pytesseract.image_to_string(sub_dir+filen
            f.write(unicode("==========================================
        pg_cntr = pg_cntr + 1
```

A production implem
associative array pro

by JACK A. RUDOLPH

Goodyear Aerospace Corporation
Akron, Ohio

INTRODUCTION

The associative or content-addressed m
been an attractive concept to computer de
since Slade and McMahon's 1957 paper[1]
"catalog" memory. Associative memories c
from the continuing problem presented by
coordinate-addressed memory which requi

```
=================================================
A production implementation of an
associative array processor—STARAN

by JACK A. RUDOLPH

Goodyear Aerospace Corporation
Akron, Ohio

INTRODUCTION

The associative or content-addressed memory has
been an attractive concept to computer designers ev
since Slade and McMahon's 1957 paper! described a
"catalog" memory. Associative memories offered reli
from the continuing problem presented by the typica
coordinate-addressed memory which requires that an
"address" be obtained or calculated before data sto
at that address may be retrieved. The associative
memory could acquire in a single memory access any
data from memory without pre-knowledge of its loca-
tion. Ordered files and sorting operations could be
eliminated. Unfortunately, early associative memori
were expensive, hence none found their way as the
```

```
matthew@LAPTOP-SVLAS6VC:/mnt/c/Users/Matthew/Desktop/Ken Batcher OCR$ python go_OCR.py
matthew@LAPTOP-SVLAS6VC:/mnt/c/Users/Matthew/Desktop/Ken Batcher OCR$ 
```

OCR App — In Development

Python3 Python Programming Programming Windows 10 Vscode

Follow

Written by Matthew Earl Miller

14 Followers · Writer for Towards Data Science

BSCS Kent State 2019, MSCS Case Western Reserve 2021, Software Developer and Graduate Student. LinkedIn — https://www.linkedin.com/in/matthew-miller-engineer

More from Matthew Earl Miller and Towards Data Science

Matthew Earl Miller

React Google Lighthouse FAB

Easily Showcase Your App's Performance

2 min read · Jul 27, 2021

8

Cristian Leo in Towards Data Science

The Math behind Adam Optimizer

Why is Adam the most popular optimizer in Deep Learning? Let's understand it by diving into its math, and recreating the algorithm.

16 min read · Jan 30, 2024

1.8K 13

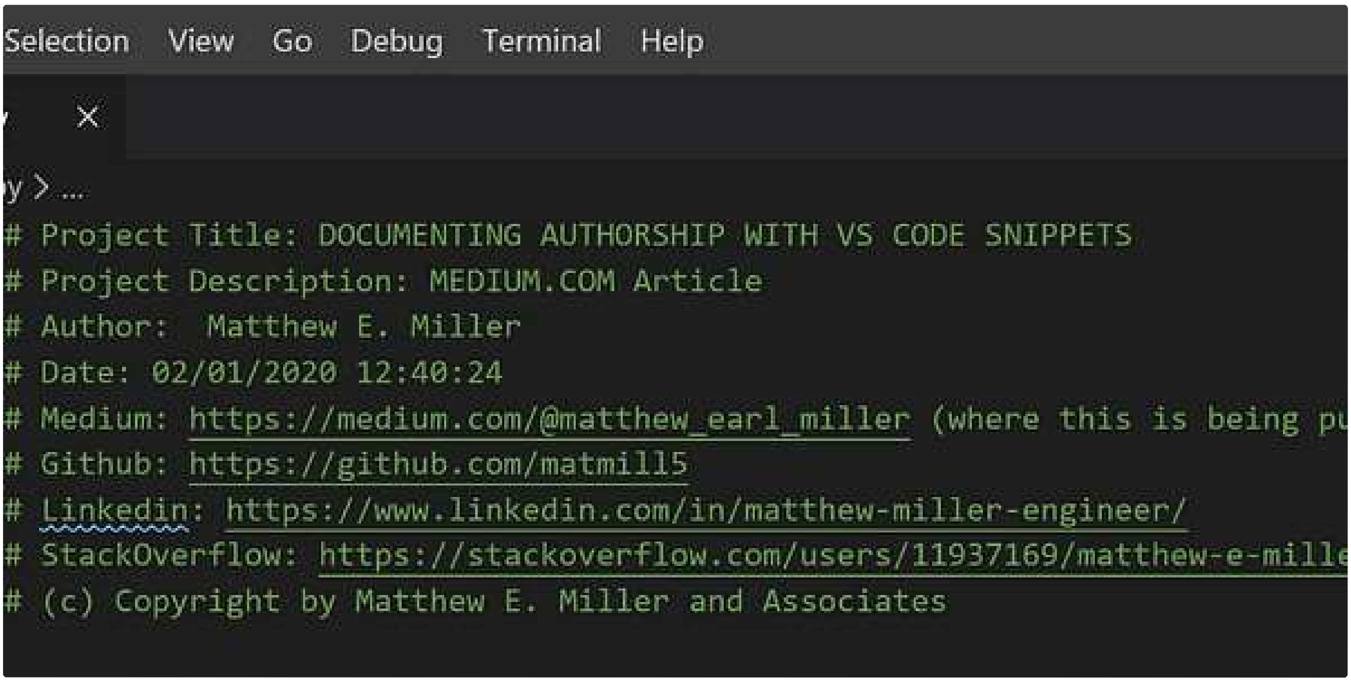
Siavash Yasini in Towards Data Science

Python's Most Powerful Decorator

And 5 ways to use it in data science and machine learning

✦ · 11 min read · Feb 2, 2024

2K 15



```
# Project Title: DOCUMENTING AUTHORSHIP WITH VS CODE SNIPPETS
# Project Description: MEDIUM.COM Article
# Author:  Matthew E. Miller
# Date: 02/01/2020 12:40:24
# Medium: https://medium.com/@matthew_earl_miller (where this is being pu
# Github: https://github.com/matmill15
# Linkedin: https://www.linkedin.com/in/matthew-miller-engineer/
# StackOverflow: https://stackoverflow.com/users/11937169/matthew-e-mille
# (c) Copyright by Matthew E. Miller and Associates
```

Matthew Earl Miller in Towards Data Science

Rapidly Document Authorship with VS Code Snippets

Be confident that you're getting credit for your work.

2 min read · Jan 3, 2020

4

See all from Matthew Earl Miller

See all from Towards Data Science

Recommended from Medium

George Stavrakis in Towards Data Science

Extracting text from PDF files with Python: A comprehensive guide

A complete process to extract textual information from tables, images, and plain text from a PDF file

✦ · 17 min read · Sep 22, 2023

1.2K 23

Gaurav Garg

How to Extract Text from PDFs and Images for LLMs Use

Large language models like GPT-3 rely on vast amounts of text data for training. While there are many open datasets available, sometimes...

4 min read · Aug 22, 2023

175 1

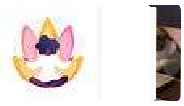
Lists

General Coding Knowledge

20 stories · 924 saves

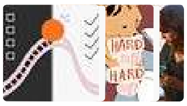
Coding & Development

11 stories · 449 saves



Stories to Help You Grow as a Software Developer

19 stories · 817 saves



Our Favorite Productivity Advice

9 stories · 383 saves

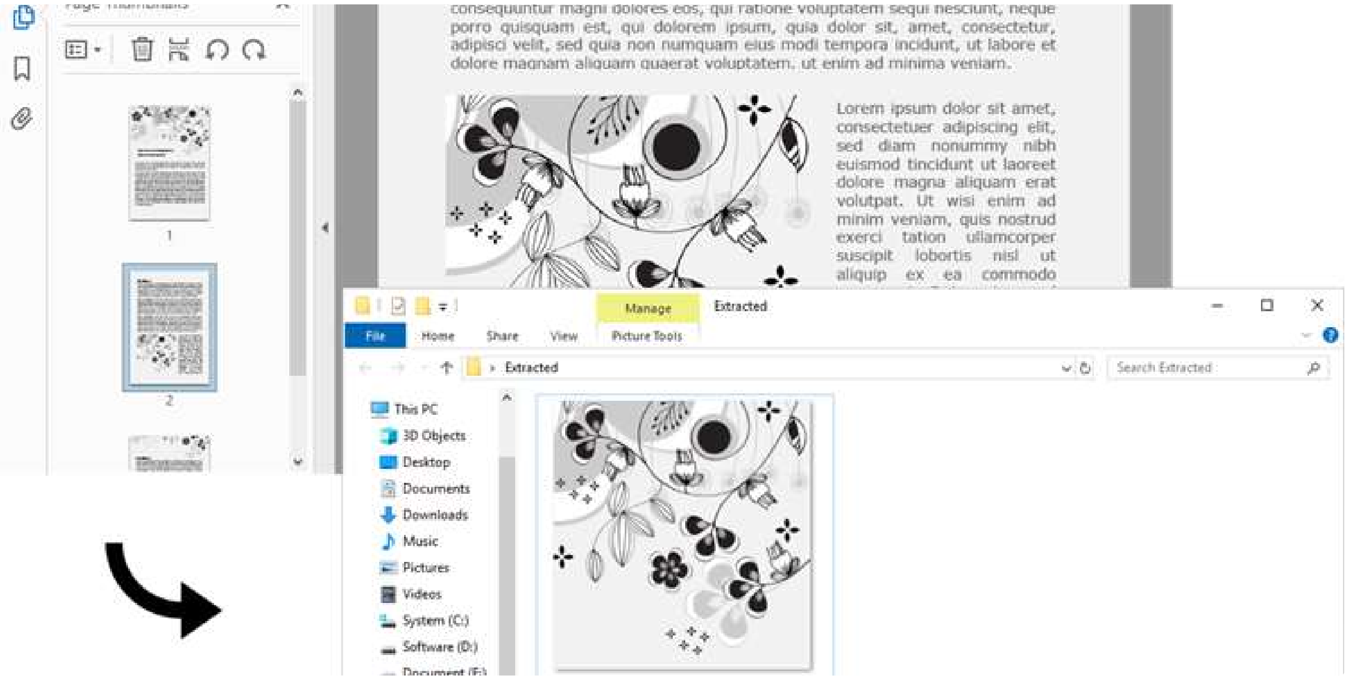
Alice Yang

with Read or Extract Text from PDF with Python — A Comprehensive Guide

PDF documents such as research papers, legal documents, contracts, or reports often contain important textual information. By extracting...

5 min read · Sep 5, 2023

6 1


consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia dolor sit, amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem. ut enim ad minima veniam.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo

Alex Stock

Extract Images from PDF Documents in Python

Extracting images from a PDF file can be a useful and practical task in various situations. Whether you need to repurpose images for a...

3 min read · Oct 12, 2023

26

VOXEL51
10 Weeks of FiftyOne Plugins
Optical Character Recognition with PyTesseract
Parse PDFs and filter by contents in FiftyOne
FiftyOne

Jacob Marks, Ph.D. in Voxel51

Optical Character Recognition with PyTesseract

Parse PDFs and filter by contents in FiftyOne

10 min read · Sep 21, 2023

278

Jason Roell

Ultimate Python Cheat Sheet: Practical Python For Everyday Tasks

This Cheat Sheet was born out of necessity. Recently, I was tasked with diving into a new Python project after some time away from the...

33 min read · Jan 30, 2024

1.3K 22

See more recommendations