

Convert text into your handwriting



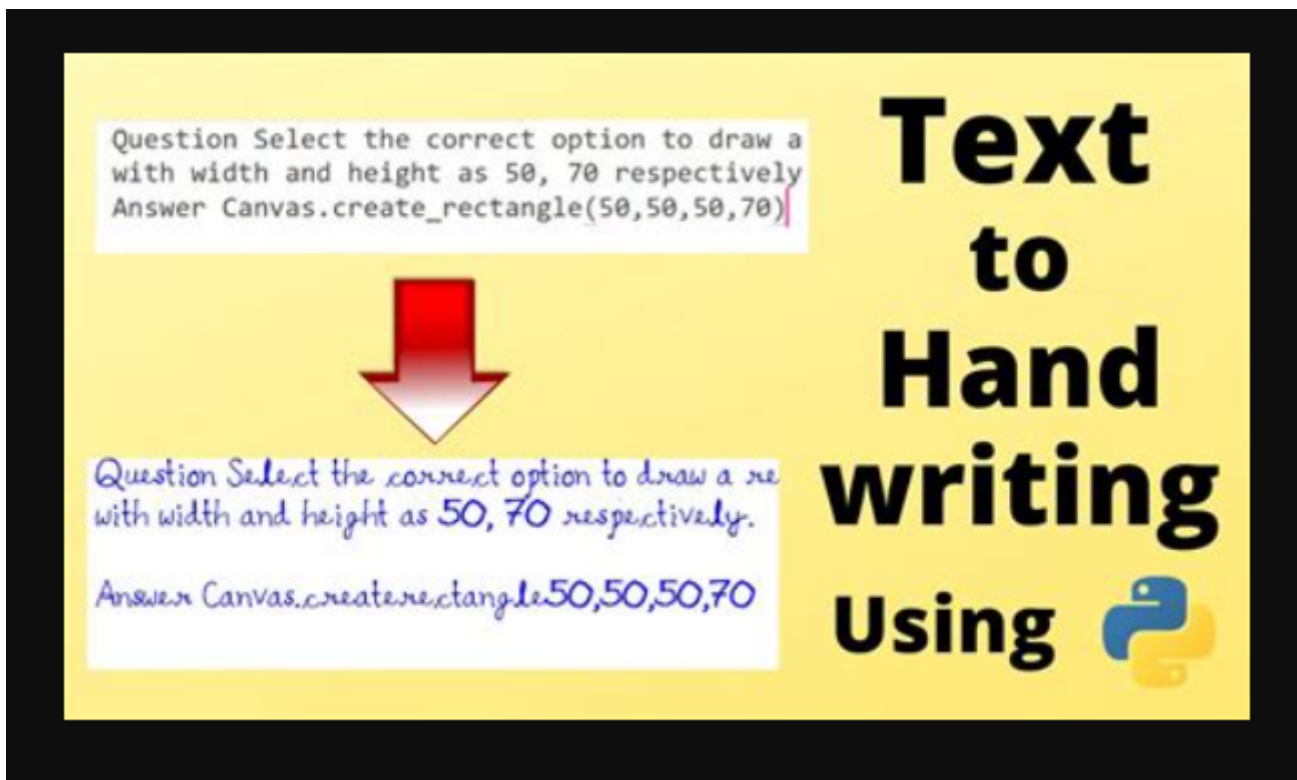
Pranav Kushare

Follow



Oct 19 · 3 min read

A simple python script to convert the text into your handwriting



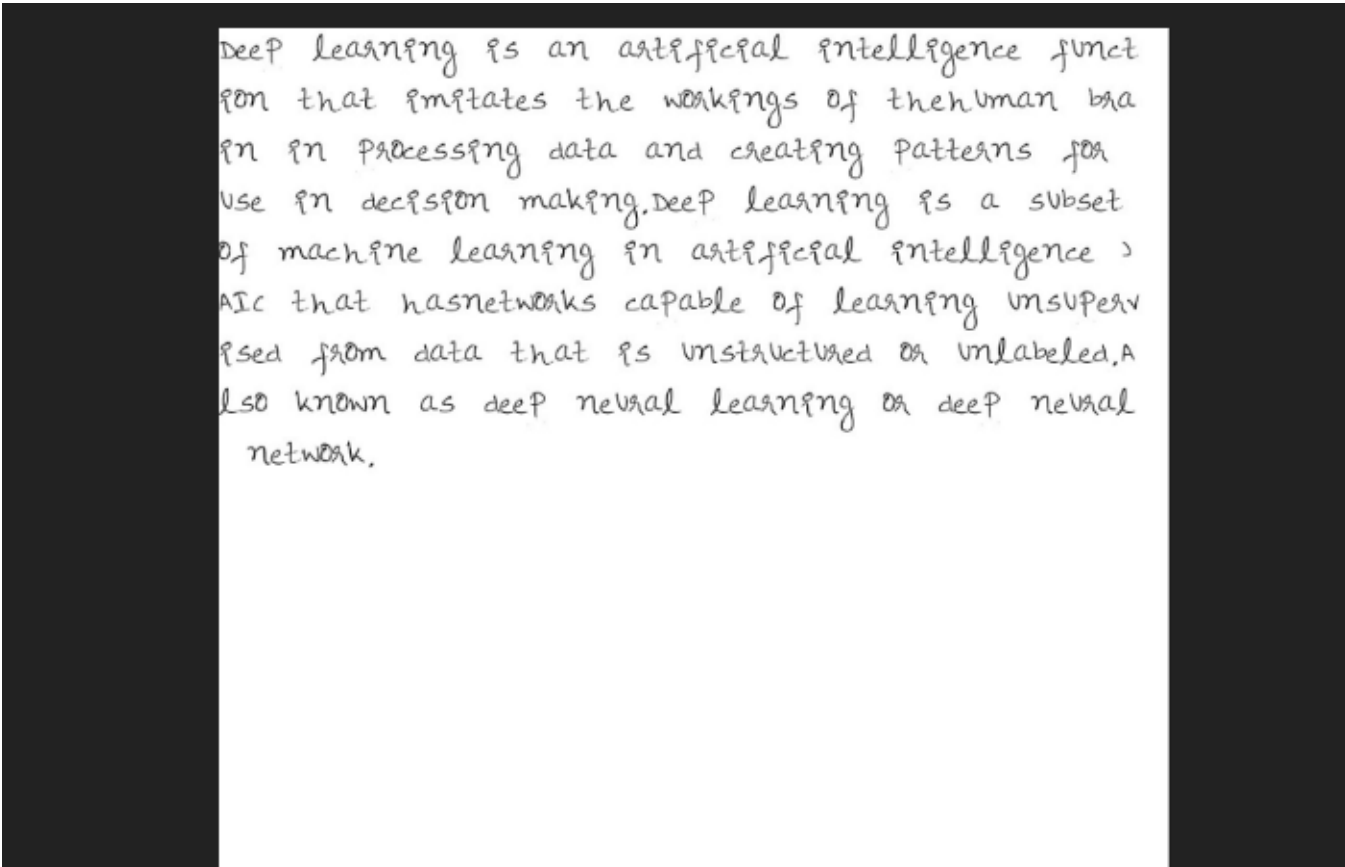
Do you ever get irritated by pointless college/school assignments? Hell yeah!!! Let's imagine you have a school writing assignment in a word or text file that you must complete in your own handwriting. Don't worry, I've got you covered. This python script will transform your digital text (writeups or assignment) into handwriting.

Input

Deep learning is an artificial intelligence function that imitates the workings of the human

brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Output

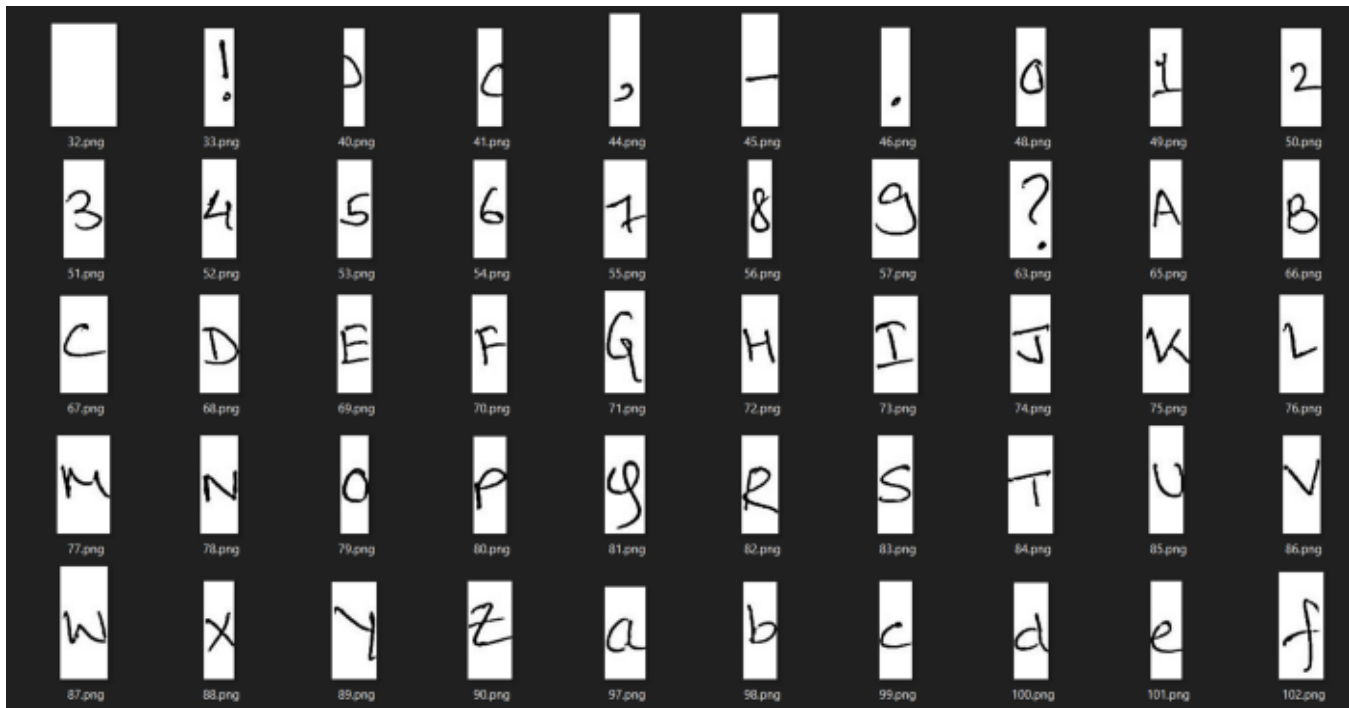


Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

Output (text converted into your handwriting)

Handwritten character data-

You will have to manually create your own handwriting dataset. Write alphabets, numbers, punctuation on a paper. Then crop all those characters and save with their respective Ascii number



Source- My computer

Simple logic

The logic behind the script is very easy and straightforward.

- We're just sequentially pasting the character's cropped images on a blank paper according to our input text

Code

Python's power can be seen through the code below. We can turn the text into our own handwriting with just 25 lines of code and the help of the built-in module PIL (Python imaging library). Let's take a peek at the code now.

```
#Importing Library
from PIL import Image

#Open the text file which you have to convert into handwriting

txt=open("dummy.txt") # path of your text file

#path of page(background)photo (I have used blank page)
BG=Image.open("myfont/bg.png")
```

```

sheet_width=BG.width
gap, ht = 0, 0

for i in txt.read().replace("\n",""):
    cases = Image.open("myfont/{}.png".format(str(ord(i))))
    BG.paste(cases, (gap, ht))
    size = cases.width
    height=cases.height
    #print(size)
    gap+=size
    if sheet_width < gap or len(i)*115 > (sheet_width-gap):
        gap,ht=0,ht+140

print(gap)
print(sheet_width)
BG.show()

```

As you know we are only using a single library called as PIL. **Python Imaging Library** is a free and open-source library for the Python programming language that adds support for opening, *manipulating*, and saving many different image file formats. By using built-in *open* method we are reading the text file in which our assignment is saved. The entire process is dependent on 2 main parameters, *ht* and *gap*. *ht* is *y position* and *gap* is *x position of respective character from origin* (In our case top left corner of the page)

Then we are using open method of Image class to read the background image. For this demonstration we are using plain white paper, but you can use any other image of your choice by adding the appropriate path. The *.replace("\n","")* method is used to remove newlines present in input text.

Now moving towards the looping part, where we'll iterate through each character in our input text and read the corresponding image. Here comes the role ascii code , remember I have asked to you save handwriting character images with their respective ascii code. Ascii code made the image reading process very easy. The function *ord(i)* outputs the ascii number of input character *i* . And the rest code is simple is very straightforward to understand we are just updating the values of variable *ht* and *gap*. Consider (*gap, ht*) as a index position of you cursor (*x,y*). Add image width of every input character to a *gap variable* . Check whether the the line width is exceeding the

width of page, if yes then move to next line by adding *140 px in ht variable* (Note the unit of width , height is pixel). This may vary in you case depending upon size of handwritten character and background page.

Thank you!

Keep an eye out for more outstanding blogs :)

[Python](#) [Image Processing](#) [Computer Vision](#) [Handwriting](#) [Script](#)

[Buy me a coffee](#)

[Git-Repo](#)

[Linkedin](#)

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

