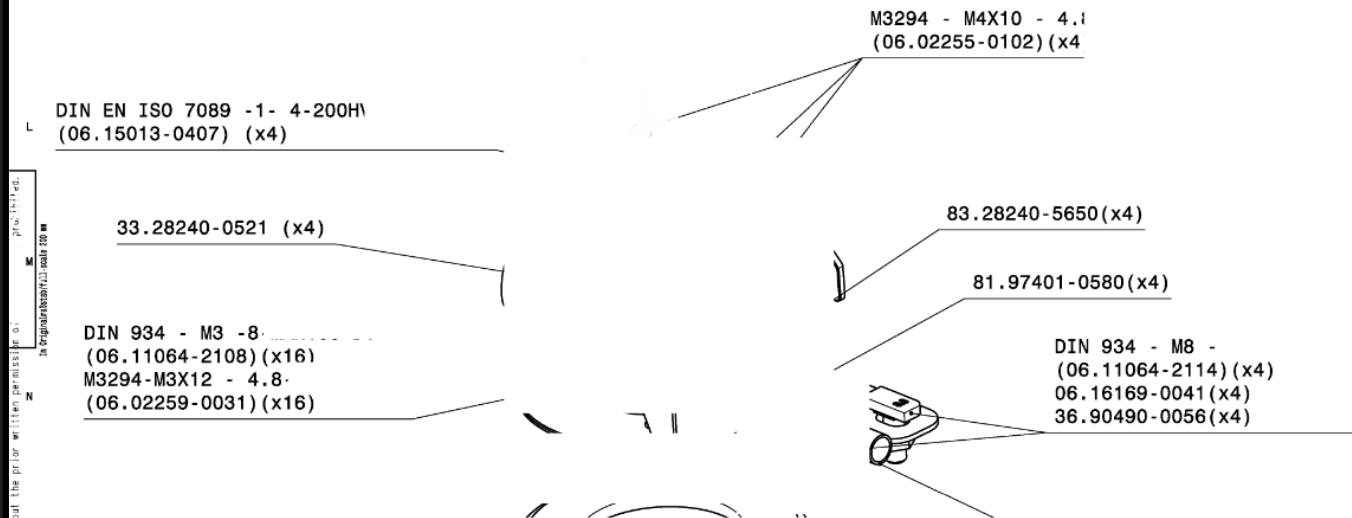


example.tif



```
(pr env) C:\>python readPart.py Run Script readPart.py via command line
'example.tif' Technical drawing name(s) will be displayed in command line
['pronttttd,', ' ', 'out the prior written permission of', ' ', 'L', ' ', 'In Originalma8stab/full-scale 200 mn', ' ',
DIN EN ISO 7089 -1- 4-200H\\', '(06.15013-0407) (x4)', ' ', '33.28240-0521 (x4)', ' ', 'a', ' ', 'DIN 984 - M38 -8 ss'
'(06.11064-2108) (x16)', 'M3294-M3X12 - 4.8.', ' ', '(06 .02259-0031 ) (x16) -', ' ', 'M3294 - M4X10 - 4.i', '(06 .0
2255-0102) (x4)', ' ', '83 .28240-5650 (x4)', ' ', '81.97401 -0580(x4)', ' ', 'Z DIN 934 - M8 -', '(06. 11064-2114) (x4)
', ' ', '06.16169-0041 (x4)', '36 .90490 -0056 (x4)', ' ', 'y']
```

06A15013A04074 is aligned from the beginning
Cleaning the unnecessary characters from the end
33A28240A05214 is aligned from the beginning
Cleaning the unnecessary characters from the end
06A11064A210816 is aligned from the beginning
Cleaning the unnecessary characters from the end
06A02259A003116 is aligned from the beginning
Cleaning the unnecessary characters from the end
06A02255A01024 is aligned from the beginning
Cleaning the unnecessary characters from the end
83A28240A56504 is aligned from the beginning
Cleaning the unnecessary characters from the end
81A97401A05804 is aligned from the beginning
Cleaning the unnecessary characters from the end
06A11064A21144 is aligned from the beginning
Cleaning the unnecessary characters from the end
06A16169A00414 is aligned from the beginning
Cleaning the unnecessary characters from the end
36A90490A00564 is aligned from the beginning
Cleaning the unnecessary characters from the end

Raw text is:

-Filtered & Separated
-Aligned

According to part number's format.

Filter Rules:

- Part numbers does not contain alphabetical values.
- Part Numbers only contain two special characters, «.» & «-»
- Part Numbers only contain digits(D)
 - DD.DDDDD-DDDD

NOTE THAT, «.» & «-» are replaced with char «A».

This can be modified to re-insert «.» & «-» ilo «A»s

Raw Text Output of Tesseract
(This print out is for demonstration
Purposes only, commented out in the code)

```
readPart.py > ...
1 | #Tesseract Python packages
2 | import pytesseract
3 | #OpenCV
4 | import cv2
5 | #Python OS
6 | import os
7 | #Python REGEX
8 | import re
9 |
10 | #Tesseract Installation path
11 | pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
12 |
13 | dirname = os.path.dirname(__file__)
14 | #Checks if a string is only numbers? If it is all letters, neglected
15 | > def isNumbers(someString):...
16 |
17 | #Check if a string with numbers fit in "Part Number" definition: "DD.DDDDD-DDDD"
18 | > def isPartNumber(stringwNumbers):...
19 |
20 |
21 |
22 |
23 |
24 |
25 |
26 |
27 |
28 |
29 |
30 |
31 |
32 |
33 |
34 |
35 |
36 |
37 |
38 |
39 |
40 |
41 |
42 |
43 |
44 |
45 |
46 |
47 |
48 |
49 |
50 |
51 |
52 |
53 | path_of_images =dirname + r"\drawings"
54 | #This function reads the .tif file, runs Tesseract and then checks elements
55 | def check_drawing(image_path, image_name):
56 |     image_name_2 = image_path + str("\\") + image_name
57 |     img_cv = cv2.imread(image_name_2)
58 |     img_rgb = cv2.cvtColor(img_cv, cv2.COLOR_BGR2RGB)
59 |     converted_string = pytesseract.image_to_string(img_rgb)
60 |     all_text_list = converted_string.strip().split('\n')
61 |     #print(all_text_list)
62 |     only_part_numbers = []
63 |     for element in all_text_list:...
64 |
65 |
66 |
67 |
68 |
69 |
70 |
71 |
72 |
73 |
74 |
75 |
76 |
77 |
78 |
79 |
80 |
81 |
82 |
83 |
84 | #Every Part Number found is in a list and with this, written into a text file
85 | with open(image_name + ".txt", "w") as text_file:
86 |     text_file.write("Here is the Part Numbers in this drawing:" + '\n')
87 |     for found_numeric in only_part_numbers:
88 |         text_file.write(found_numeric+ '\n')
89 |
90 |     text_file.write('\n' + "****Experimental Code Results****")
91 |
92 |
93 | #Main Function, repeats all steps above for EACH .tiff file in \drawings folder
94 | if __name__ == "__main__":
95 |     # execute only if run as a script
96 |     image_path = path_of_images
97 |     image_list = os.listdir(dirname + r"\drawings\.")
98 |     print(image_list)
99 |     for image in image_list:
100 |         check_drawing(image_path + str("\\") + image)
```

drawings	02-Apr-22 23:47	File folder	
pr_env	31-Mar-22 12:09	File folder	
readPart.py	03-Apr-22 02:04	Python File	4 KB
1.tiff.txt	03-Apr-22 01:15	Text Document	1 KB
2.tiff.txt	03-Apr-22 01:15	Text Document	1 KB
3.tiff.txt	03-Apr-22 01:15	Text Document	1 KB
4.tiff.txt	03-Apr-22 01:16	Text Document	1 KB
5.tiff.txt	03-Apr-22 01:16	Text Document	1 KB
6.tiff.txt	03-Apr-22 01:16	Text Document	1 KB
7.tiff.txt	03-Apr-22 01:16	Text Document	1 KB
10.tiff.txt	03-Apr-22 01:15	Text Document	1 KB

Part Number .txt files
are placed into same
directory with the
script. Each .tiff will
have its own .txt file