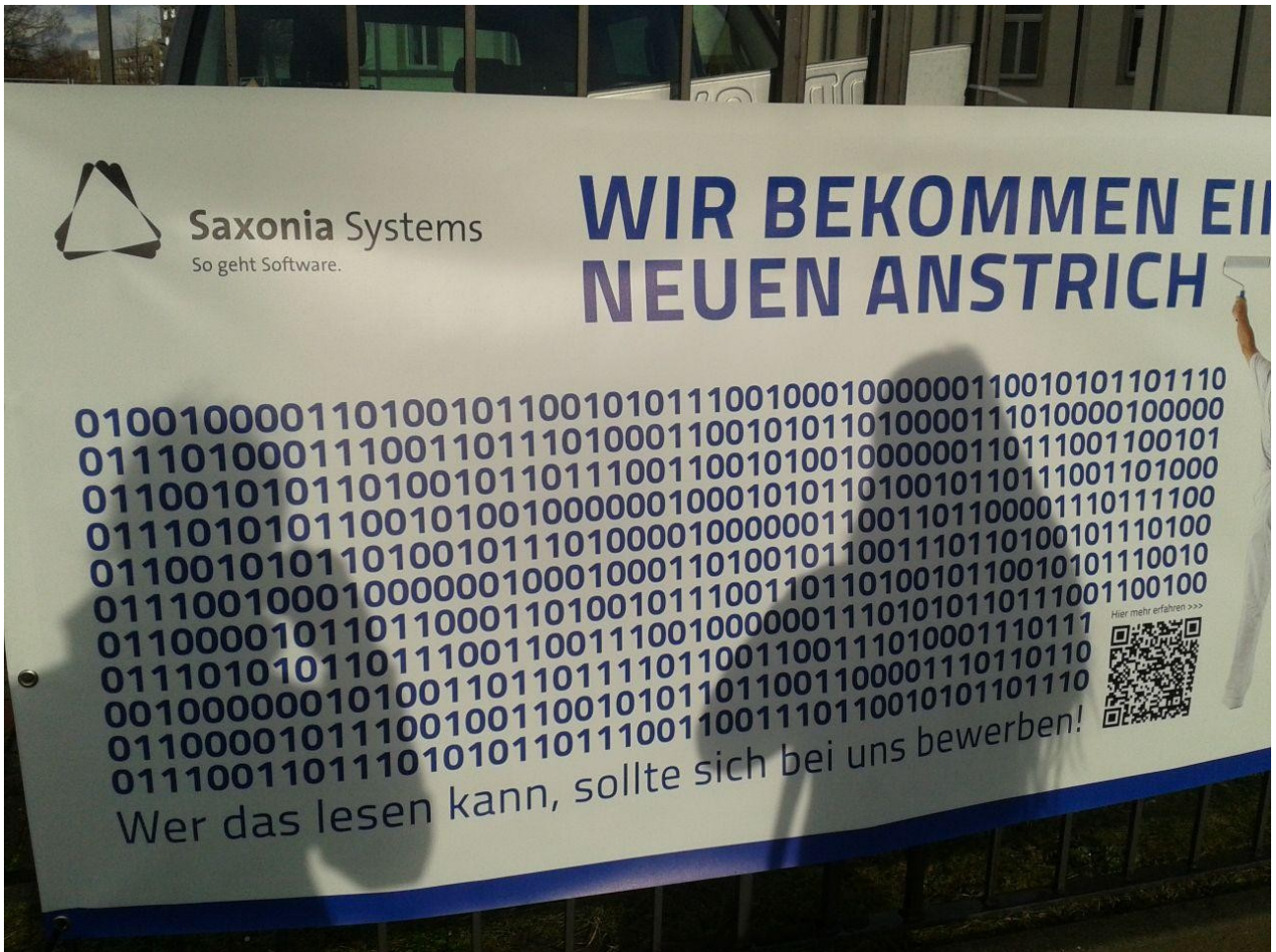


# “Bewerbung” ;)

Sehr geehrte Damen und Herren der Saxonia Systems AG,

ich habe heute mittag das folgende Plakat gesehen und konnte nicht widerstehen:



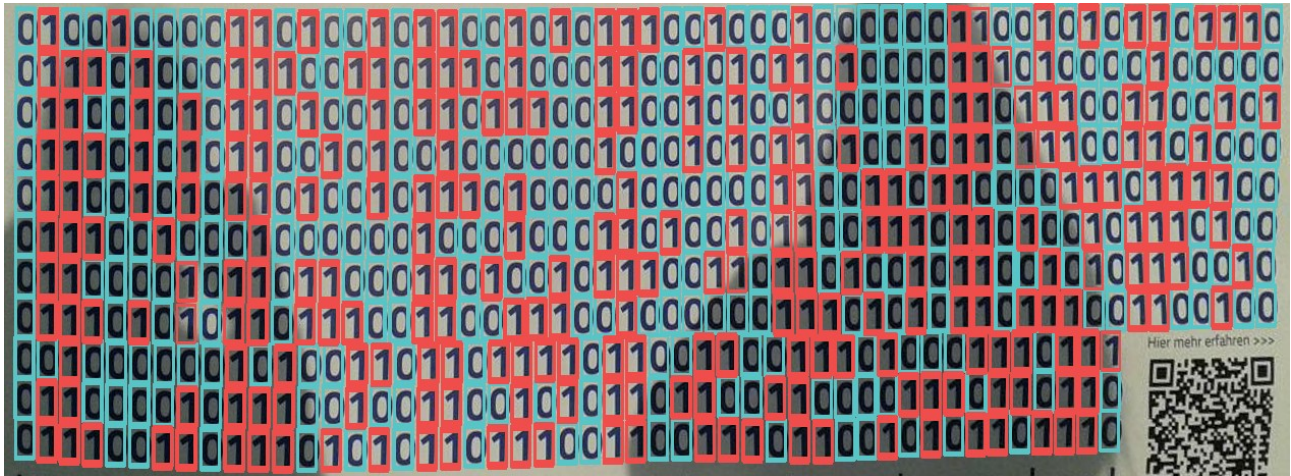
Erstmal ein wenig entzerren um die folgende Arbeit zu erleichtern:



Dann die beiden Zeichen ausschneiden:



Ein wenig template matching:



Zuletzt die Ergebnisse noch zeilenweise gruppieren, doppelte matches entfernen und dekodieren:

```
Terminal
File Edit View Search Terminal Help
one_zero_detector.py
01001000011010010110010101110010001000000110010101101110
01110100011100110111010001100101011010000111010000100000
01100101011010010110111001100101001000000110111001100101
01110101011001010010000001000101011010010110111001101000
01100101011010010111010000100000011001101100001110111100
01110010001000000100010001101001011001110110100101110100
01100001011011000110100101110011011010010110010101110010
01110101011011100110011100100000011101010110111001100100
001000000101001101101111011001100111010001110111
011000010111001001100101011011001100001110110110
011100110111010101101110011001110110010101101110
Hier entsteht eine neue Einheit für Digitalisierung und Softwarelösungen
>>> |
```



Quellcode: (Syntax Highlighting courtesy of <https://tohtml.com/python/>)

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

import sys
print(sys.argv[0])

img_rgb = cv2.imread('./saxonia_werbung_entzerrt_1.png')
img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)

c1 = (89, 197, 199)
c2 = (240, 78, 77)

chars = []

# tutorial on how to do template matching with python and opencv:
# https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/
py_imgproc/py_template_matching/py_template_matching.html

for template_path, color, value in (("0.png", c1, 0), ("1.png", c2, 1)):
    template = cv2.imread(template_path, 0)
    w, h = template.shape[::-1]

    res = cv2.matchTemplate(img_gray, template, cv2.TM_CCOEFF_NORMED)
    threshold = 0.7
    loc = np.where( res >= threshold)
    for pt in zip(*loc[::-1]):
        cv2.rectangle(img_rgb, pt, (pt[0] + w-5, pt[1] + h-3), color[::-1], 1)
        chars.append((pt[::-1], value))

chars.sort()

text = bytearray()

THRESHHOLD = 30
while chars:
    y_max = chars[0][0] + THRESHHOLD
    line = []
    while chars and chars[0][0] < y_max:
        y, x, value = chars.pop(0)
        line.append((x, value))
    line.sort()
    linestring = ""
    x_last = -1000
    for x, v in line:
        if abs(x-x_last) < 10:
            continue
        linestring += str(v)
        x_last = x
    print(linestring)
    bytes_as_strings = map(''.join, zip(*[iter(linestring)]*8))
    for bytestring in bytes_as_strings:
        byte = int(bytestring, 2)
        text.append(byte)

text = bytes(text).decode("utf-8")
print(text)

cv2.imwrite('res.png', img_rgb)
cv2.imshow('res.png', img_rgb)
cv2.waitKey()
```