

Übungsblatt 3

Übung 3.1 Schreiben Sie eine Funktion, die aus einem übergebenen Grauwert Histogramm den Otsu-Schwelwert berechnet. Das Histogramm sei von der Form wie es die Bildmethode *histogram()* zurückgibt.

Sie können Ihre Funktion überprüfen, indem Sie Ihren berechneten Wert mit dem Rückgabewert der Bildmethode *otsu_find_threshold()* vergleichen.

Übung 3.2 Wenden Sie auf das bereitgestellte Greyscale-Bild *grey_a_poor.png* verschiedene lokale Binarisation-Methoden an (z.B. *sauvola_threshold*, *white_rohrer_threshold* oder *shading_subtraction*), siehe Dokumentation “Plugins/Binarization”) und stellen Sie die Unterschiede zum Otsu-Thresholding wie folgt grafisch dar:

- Pixel, die bei Otsu schwarz sind, aber beim anderen nicht werden rot markiert
- Pixel, die bei Otsu weiß sind, aber beim anderen schwarz werden grün markiert

Die Markierung erfolgt durch die *highlight* Methode, z.B.

```
rgb = img.to_rgb()
rgb.highlight(other, RGBPixel(255,0,0))
```

Die Pixel, die nur bei einem Bild schwarz sind, erhalten Sie mittels Differenzbildung (*subtract_images*).

Übung 3.3 Bestimmen Sie in den bereitgestellten Bildern die häufigste schwarze und weiße vertikale Lauflänge (siehe Dokumentation Abschnitt “Plugins/Runlength”) und markieren Sie alle Lauflängen der jeweils häufigsten Lauflänge rot bzw. grün.

Anleitung: Alle schwarzen vertikalen Lauflängen der Länge n erhalten Sie, indem Sie von einem Bild alle Lauflängen größer n (*filter_tall_runs*) und alle kleiner n (*filter_short_runs*) abziehen. Achtung: diese Funktionen arbeiten direkt auf dem Bild. Sie müssen dies also auf einer Kopie des Bildes durchführen, die Sie mit *image_copy* erzeugen können.

Die weißen Lauflängen der Länge n erhalten Sie, indem Sie dieselben Operationen auf einem invertierten Bild (*invert*) durchführen.

Übung 3.4 Führen Sie eine *cc_analysis* auf einem Bild durch und geben Sie alle verwendeten Labels in sortierter Reihenfolge aus. Alle Labels können Sie elegant über eine “List-Comprehension” auslesen:

```
labels = [c.label for c in ccs]
```

Sind die Labels konsekutiv oder gibt es Lücken?