autocrop Guide

Author: Kim Miikki
Date: 18.2.2022

1 Introduction

Unwanted image marginals should often be removed. This task can be automated by using *autocrop.py*. RGB mean values for each side of the image are calculated, then the selection rectangle is minimized column and row, wise until the new mean value differs from the last mean value. The required difference can be controlled with a threshold value. Also new marginals can be set from the the found crop rectangle.

2 System Requirements

Operating System: Raspberry Pi OS or Linux

Python 3 with OpenCV

3 Program Usage

The program should be executed in the directory where the images are located. All optional arguments can be listed with argument -h:

```
$ rgb-sgraph -h
usage: autocrop.py [-h] [-i I] [-o] [-x X] [-y Y] [-t T] [-f]

optional arguments:
  -h, --help  show this help message and exit
  -i I        input file name
  -o          preserve original file format
  -x X        maximum horizontal marginal as integer
  -y Y        maximum vertical marginal as integer
  -t T        threshold value as float, 1e-09 < t <= 255
  -f          fix odd dimensions by adjusting crop marginals
```

Cropped images are saved under a subdirectory named ‘crop’. Here is a list of currently supported file formats: .png, .jpg, .bmp, .tif, .tiff, .dib, .jpeg, .jpe, -pbm, .pgm, .ppm, .sr and .ras.

In single file mode (-i) a roi.in file is created. Unless the -i option is used, the program is executed in a batch mode, and it tries to autocrop all images in current directory. Original file format is preserved with argument -o. Fixed maximum marginals can be set with arguments -x and -y. Possible odd dimensions can be fixed with the then -f argument to even. It is done by extending or reducing the left and the bottom marginals by one pixel.

Without a threshold value (argument -t), even a smallest difference between the current and the last RGB mean values will stop the search of the marginal. Large threshold values often causes excess cropping. The *rgbxy.py* tool can be used for determining a optimal threshold value.

4 Use Cases

The usage of this program is demonstrated in this guide with two uses cases: cropping Finlands state flag and cropping text areas with marginals.

4.1 Cropping the coat of arms from the state flag of Finland

The state flag of Finald is shown in figure 1.

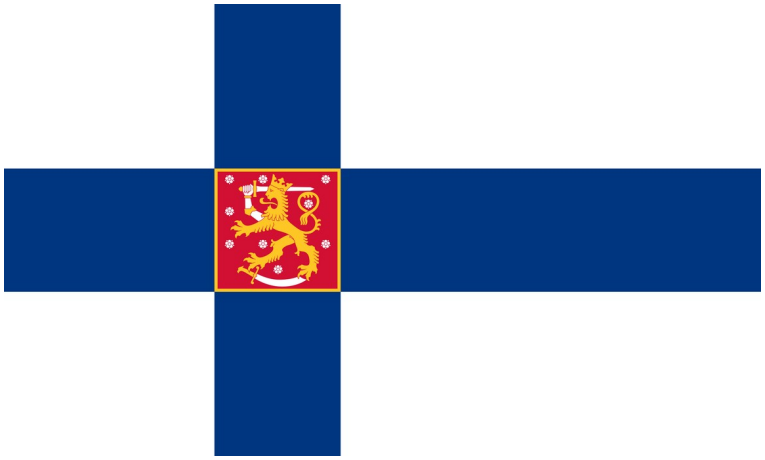

Figure 1. Flag of Finland (state).

The coat of arms is cropped easy with the following command:

```
$ autocrop.py -t 40
Autocrop 1.0, (c) Kim Miikki 2022

Current directory:
/media/pi/data

Processing:
Flag_of_Finland_(state)_2560px.png, size 2560x1564

Files processed: 1
Time elapsed: 0:00:01.134477
```

The cropped image is shown in figure 2.

Figure 2. Coat of arms of Finland.

4.2 Cropping Text with Marginals and even dimensions

Two ‘Lorem Ipsum’ images were cropped with autocrop with 16 pixels x and y marginals, and adjusted to even dimensions with this command:

```
$ autocrop.py -x 16 -y 16 -f
Autocrop 1.0, (c) Kim Miikki 2022

Current directory:
/media/pi/data/transfer/20220218-autocrop-doc

Processing:
1280px-Lorem_ipsum_design_part.png, size 388x550
lorem_ipsum_dark.png, size 640x281

Files processed: 2
Time elapsed: 0:00:00.091481
```

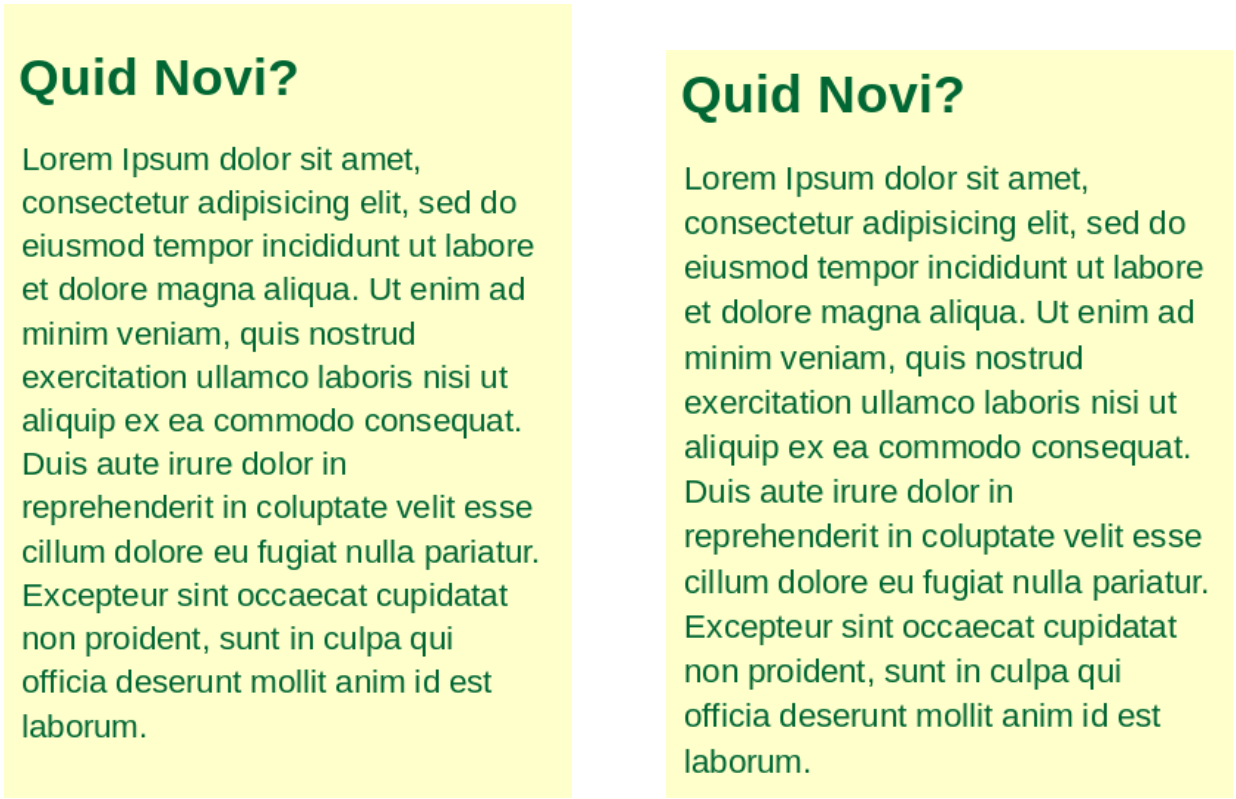

Figure 3. Dark text on light background - original and cropped images.

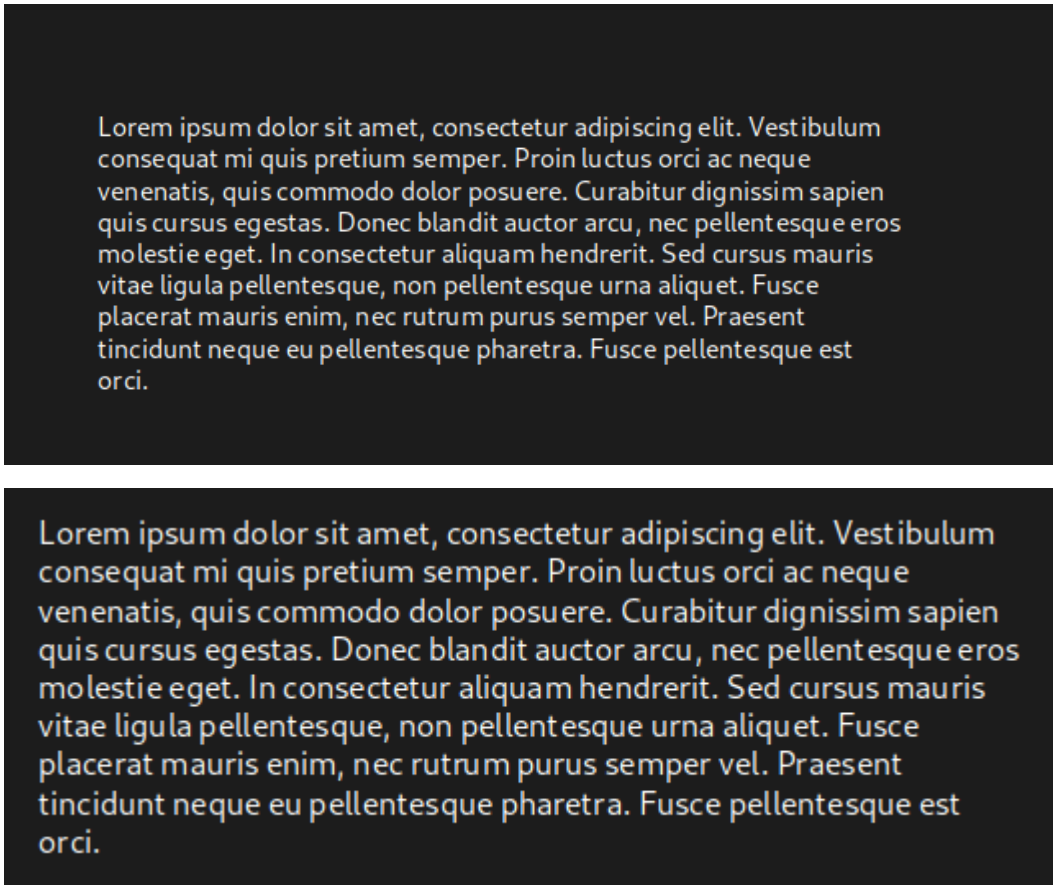

Figure 4. Automatically cropped image of light text on dark bakground.