Computer Vision I

Heiko Neumann, Christian Jarvers
Institute of Neural Information Processing
Ulm University

Assignment 1: Submission date (Moodle): May 18, 2023, 12:00.

1 Histogram Calculation (6 points)

In the first assignment we aim at calculating a histogram of an image using a custom-built algorithm. This means that no existing Julia functions for histogram calculation like `imhist` should be used.

1. Write a Julia function `myhistogram` which takes a greyscale intensity image (single channel) in the range $[0, 1]$ and returns a vector of length 256 with each element in the vector denoting the normalized frequency of the corresponding grey value. Apply a normalisation such that the resulting frequencies can be interpreted as probabilities. Template for the function:

```julia
"""
        myhistogram(img)

Takes the grayscale image `img` and returns its histogram.

# Arguments
- `img::Array{T,2}`: 2D array representing a grayscale image
  in range [0, 1]
"""
function myhistogram(img)
        # pro tip: check if the input img is indeed an image
        # with the defined intensity range
end
```

2. Load the images `fruitA.png` and `fruitB.png` (Figure 1) by using `load("imgname")` (from package `FileIO`). Use your function `myhistogram` to calculate a histogram for each image.
3. Show the original images and your calculated histogram together in one `plot`. Make sure you label your plots appropriately with commands like `title`, `xlabel`, and `ylabel`. Hint: use the function `bar` to plot the histograms.
4. Try to find the regions in the histogram which correspond to the objects in the image (fruits and background). Mark the regions by placing a text over the histogram (e.g. by

using the `annotate!` function from `Plots`).

5. Extract from both images the rows with index 50 and 200, respectively (counting from the top, first row is 1) and plot the profile of intensities along the x-axis. Again, `annotate` the visually distinct regions of these profiles. Be careful: an intensity profile is not the same thing as a histogram.

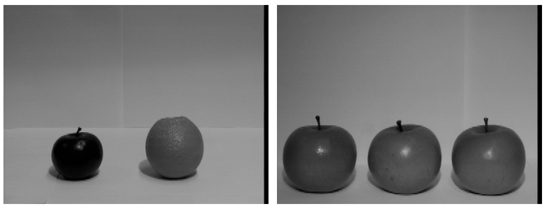

Figure 1: Images `fruitA.png` (left) and `fruitB.png` (right).

2 Neighborhood operations with local weighting (4 points)

Operations which take the local neighbourhood into account are very common in image processing. Filter masks are used to calculate the local weighted sum of intensities (or grey values) around each pixel position (c.f. course notes part II).

1. (*By hand*) Consider the following image I and mask F:

$$I = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & {\color{red}0} & {\color{red}1} & {\color{red}0} & 0 \\ 0 & {\color{red}0} & {\color{red}0} & {\color{red}0} & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad \text{and} \quad F = \begin{pmatrix} -1 & -1 & 0 \\ -1 & {\color{green}0} & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

   Calculate the response of the mask for the positions marked in red (six result values) with the reference position of the mask in the center (marked in green).

2. In Julia, such filtering operations with a mask can be achieved with the function `imfilter(img, kernel)` from the package `ImageFiltering`. Load the images `lake_gray` from the package `TestImages` and apply the mask F from above. Display the result (together with the original image). For which type of use cases may this mask be suitable?

Submission procedure

- Submission exclusively in Moodle.
- Please name all scripts or notebooks with the current assignment and exercise number, e.g. `sh01ex02.jl` for the second assignment of the first assignment sheet.

- Please present your results in a pdf-document or Jupyter notebook with
  - your name and matriculation number
  - and a brief description of your results and images.
- Please submit all files as a zip-document.

Have fun!