Image Processing – Questions

1 Image processing softwares and tools

List at least 5 software tools, which can help to solve image processing tasks! Describe them briefly!

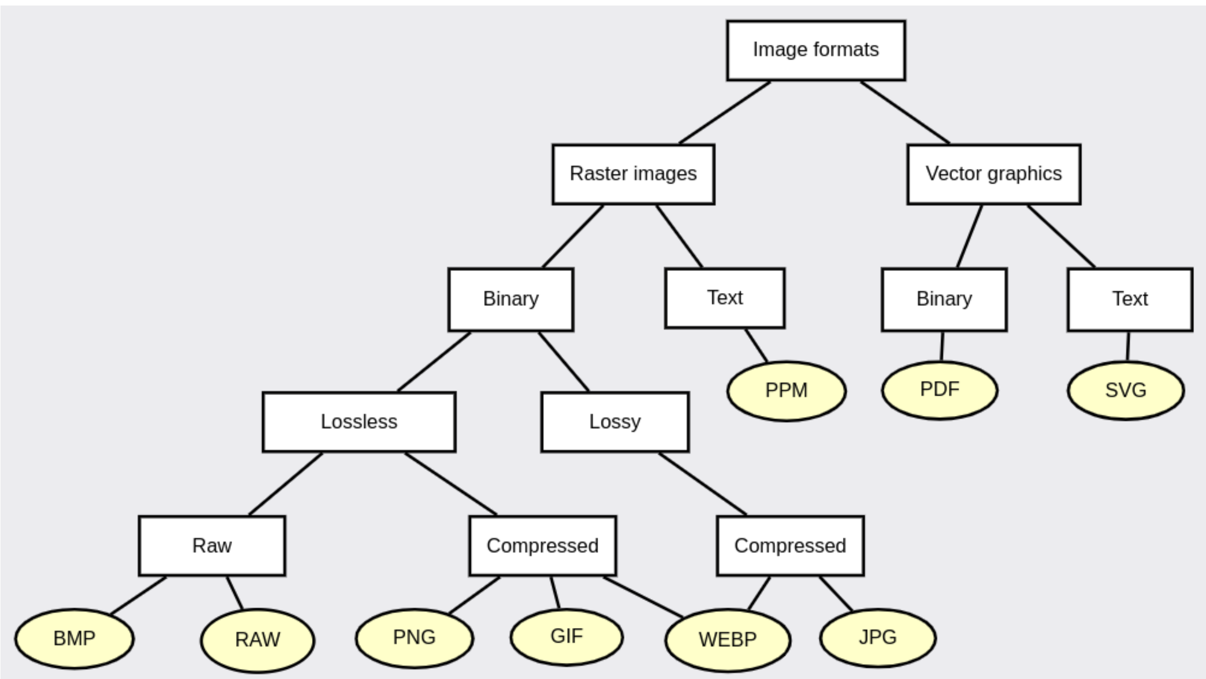
We can use for example the following softwares.

- *OpenCV*: Open source, standardized computer vision library, mainly for the C++ programming language. There is a wrapper for Python.
- *NumPy*: Numerical library for Python. It provides tools for linear algebra (vectors, matrices and their operations).
- *Matplotlib*: Graph plotter, and data visualization library for Python. It is able to use the types of NumPy.
- *Scikit Learn*: Python library for machine learning.
- *Jupyter*: Interactive, notebook based framework for research and development. It is mainly used with Python, however it can be used further programming languages also.
- *GIMP*: It is a free and open source image manipulation software.

The followings also could be mentioned: Seaborn, PhotoShop, PyTorch, TensorFlow, SciPy, Pillow/PIL.

2 Image formats

Let overview the main image formats and their commonly used classification! List the most frequently used file formats and describe them!



- BMP: *BitMap*, simple, bitmap format.
- RAW: Raw image format. Mainly used in photography.
- PNG: *Portable Network Graphics*, designed as the main image format of the Internet.
- GIF: *Graphics Interchange Format*, for smaller, animated images with transparent parts optionally.
- WebP: the next image format of the Internet (designed and proposed by Google).
- JPG: *Joint Photographic Experts Group*, optimized for photographs.
- PPM: *Portable PixMap*, simple, raster image format.
- PDF: *Portable Document Format*, it can be considered as a binary vector graphic format.
- SVG: *Scalable Vector Graphics*, XML-based vector graphic format.

3 Grayscale mapping

Let define at least 3 calculation methods, which are able to map an RGB color to a grayscale color!

The grayscale mapping is a $\mathbb{R}^3 \to \mathbb{R}$ (more precisely $[0,1]^3 \to [0,1]$) function. Some frequently used approaches:

- $f(r,g,b) = \frac{r+g+b}{3}$.
- $f(r,g,b) = \max(r,g,b)$.
- $f(r,g,b) = \frac{\min(r,g,b)+\max(r,g,b)}{2}$.

4 Hue-based color spaces

Let describe the HSI, HSV and HSL color spaces!

`https://en.wikipedia.org/wiki/HSL_and_HSV`

5 Noise types

List at least 5 types of noises and their reasons!

Some of the frequent noise types are the followings:

- Salt-and-pepper noise: it can be the result of faulty image capture hardware (for example a faulty CCD sensor).
- White noise: it can caused by uneven lighting, various kind of radiations.
- Blurred parts: bad camera calibration, inappropriate focus, depth of field.

- Colorspace shifting: colorful air or water (or any kind of light transfer medium).
- Overexposed: inappropriate exposition time.
- Perspective torsion: it is the reason of the physical properties of the lenses, cameras.
- Motion blur: moving camera in exposition time.
- Sampling, compression noise: for example the JPG noise.
- Missing or hidden parts: inappropriate camera setting, physical obstacles.

6 Histogram calculation

Let denote $I \in [0, 255]^{n \times m}$ a grayscale image matrix! Describe (by using mathematical formulas and/or pseudo code) the calculation of image histogram!

Let denote $h \in \mathbb{Z}^{256}$ the histogram (as a vector)! Let assume that, we use zero-based indexing. We can calculate the value of h as the followings:

$$h[k] = |\{(x, y) \in I : I[x, y] = k\}|.$$

Pseudo code:

$$
\begin{array}{l}
\text{CALC\_HISTOGRAM}(I, @h) \\
// \text{ Input} : I \in [0, 255]^{n \times m} \\
// \text{ Output} : h \in \mathbb{Z}^{256} \\
\text{FOR } k \leftarrow 0 \text{ TO } 255 \text{ DO} \\
\quad h[k] \leftarrow 0 \\
\text{FOR } i \leftarrow 1 \text{ TO } n \text{ DO} \\
\quad \text{FOR } j \leftarrow 1 \text{ TO } m \text{ DO} \\
\quad\quad h[I[i, j]] \leftarrow h[I[i, j]] + 1 \\
\text{RETURN}(h)
\end{array}
$$

7 Histogram operations

Let describe the histogram stretching and histogram equalization methods!

By using the *histogram stretching* method, we would like to map the original intensity range to the largest one (typically to $[0, 255]$). We can calculate the I' intensities from the original I intensities by the following formula:

$$I' = \frac{I - I_{\min}}{I_{\max} - I_{\min}} \cdot 255,$$

where $I_{\min}$ is the minimal and $I_{\max}$ is the maximal intensity of the picture.

The *histogram equalization* method tries to approximate the uniform distribution. In other words, it tries to apply a mapping, which results that the number of grayscale pixels try to be the same.

8 Linear convolution

Let describe the linear convolutional filter by using a two dimensional kernel matrix!

Let denote the linear convolutional filter by

$$g \in \mathbb{R}^{(2k+1)\times(2k+1)}.$$

In discrete case we can calculate by using the following formula:

$$(f * g)(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f(x + j, y + i) \cdot g(j, i),$$

where f is the intensity of the original image, and g is the convolutional kernel.

9 Gaussian filter

What is the Gaussian filter? Show an example as an approximation by using discrete linear convolution!

The Gaussian filter is a convolutional filter, where the convolutional kernel tries to approximate the Gaussian (normal distribution) curve. It can be applied in arbitrary dimensional space. In the case of images, it is the approximation of two-dimensional Gaussian distribution surface.

As a Gauss filter, we can use for example the following kernel matrix:

$$g = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

10 Median filter

Let define the method of median filtering! Describe the main characteristics of the median filtering!

The median filter is a non-linear filter. The name is come from the calculation method, because we sort the intensities which belong the the kernel (sliding window) and choose the element at the middle (the median).

- One benefit of the method is that, it does not add new intensity to the image.
- It is able to filter the point-like noise.
- Its main drawback is the computational complexity. (It requires sorting for all filtered pixels.)

11 Edge detection

How can we estimate the edges on a grayscale image? Let define a linear convolutional approximation method!

We can estimate the edges by using the magnitude of the gradient vector at the given point. Let denote $I \in \mathbb{R}^{n \times m}$ the matrix of the intensities! The gradient vector at (x, y) is

$$\nabla I(x,y) = \left[\frac{\partial I(x,y)}{\partial x}, \frac{\partial I(x,y)}{\partial y}\right].$$

A possible numerical approximation (by using linear convolution) is the Sobel operator. Its kernels:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

12 Local thresholding

Show a simple example of an adaptive, local thresholding algorithm!

For local thresholding, we can use (for example) the following algorithm:

- Let determine the set of points for all pixels of the image, by using a maximal distance from the considered point.
- Let calculate the mean of these pixels, and use it as the local threshold value.