Exercise 05 for MA-INF 2201 Computer Vision WS24/25
17.11.2024
Submission on 24.11.2024

1. A function is *submodular* when it satisfies the equation:

$$P(\beta,\gamma) + P(\alpha,\delta) - P(\beta,\delta) - P(\alpha,\gamma) \geq 0$$

for all $\alpha, \beta, \gamma, \delta$ such that $\beta > \alpha$ and $\delta > \gamma$. Show whether the following functions are submodular. If a function is not submodular, provide an example that violates the condition.

   1.1. $P(\omega_m, \omega_n) = \kappa(\omega_m - \omega_n)^2$ ***(1 points)***

   1.2. $P(\omega_m, \omega_n) = \kappa(1 - \boldsymbol{\delta}(\omega_m - \omega_n))$ ***(1 points)***

$\boldsymbol{\delta}(.)$ is the Kronecker delta function and $\kappa > 0$ is a constant.

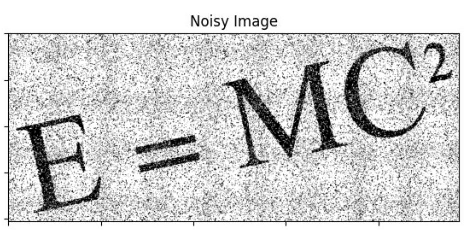
2. Provide a graph structure using the Alpha Expansion model for the 5 nodes (a,b,c,d,e) with states before: $\boxed{\beta|\gamma|\alpha|\alpha|\beta}$ where the label α is expanded. ***(3 points)***

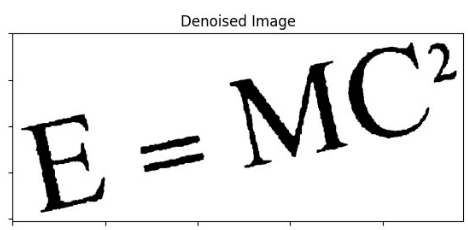
3. Show the cut on the graph in Question 2 with states after: $\boxed{\beta|\alpha|\alpha|\alpha|\beta}$. Write down the total cost which includes unary and pairwise costs. ***(3 points)***

Programming Exercises

In this part of the exercise, you need to install and use "PyMaxflow" package. Before starting to write code, please refer to the tutorial documentation for the package: `https://pmneila.github.io/PyMaxflow/tutorial.html`

4. **Binary MRF:** Denoise the binary image using a Markov random field (MRF). Read the noisy binary image in *images/noisy_binary.png*.





Follow these steps:

   4.1. Create a graph for the image using all the pixels as nodes.

   4.2. Connect each pixel (node) to the "source node" and the "sink node" using directed edges. Define the unary costs and assign them to these edges according to

$$\begin{aligned} P(x_n|w_n = 0) &= Bern_{x_n}[\rho] \\ P(x_n|w_n = 1) &= Bern_{x_n}[1-\rho] \end{aligned}$$

   where *Bern* is the Bernoulli distribution.

4.3. Create directed edges for neighboring pixels. Define the the pairwise costs and assign them to these edges. The pairwise costs are defined as follows:

$$P(w_m = 0, w_n = 0) = P(w_m = 1, w_n = 1) = \theta_s$$
$$P(w_m = 0, w_n = 1) = P(w_m = 1, w_n = 0) = \theta_d$$

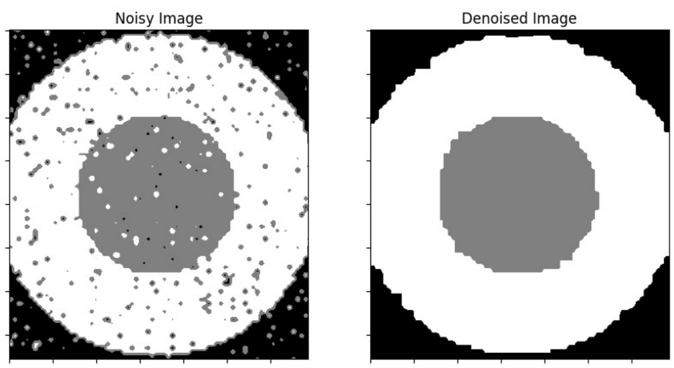
where θ_s and θ_d represent the cost of assigning identical and distinct labels to neighboring pixels, respectively. (In Pymaxflow, non-terminal edges are defined forwards and backwards in default.)

4.4. Perform Maxflow optimization.

4.5. Extract labels for each pixel and construct the denoised image.
Use different combinations of ρ, θ_s and θ_d by comparing visualizations. Then, visualize the best denoised output. What is your insight about changing ρ, θ_s and θ_d and their effects on the output?

(6 points)

5. **Multi-label MRF with Alpha Expansion:** Extend binary MRF algorithm solution for a denoised grayscale image in *images/noisy_grayscale.png* using Alpha Expansion. In this case, there are three labels $[l_1, l_2, l_3]$ for image pixels corresponding to gray values of (0, 128, 255) respectively.



In the Alpha Expansion function, define unary costs for keeping or changing the label based on the selected ρ value, or assign them independently if preferred. Define the pairwise costs by using

(a) Quadratic function: $P(\omega_m, \omega_n) = \kappa(\omega_m - \omega_n)^2$

(b) Truncated quadratic function: $P(\omega_m, \omega_n) = \min(\kappa_1, \kappa_2(\omega_m - \omega_n)^2)$

(c) Potts model: $P(\omega_m, \omega_n) = \kappa(1 - \boldsymbol{\delta}(\omega_m - \omega_n))$ where $\boldsymbol{\delta}(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{else} \end{cases}$

and consider all possible relationships between adjacent pixels. Test each function and select the one that produces the best denoised image for visualization.

(6 points)

* For programming exercises, you can also use Jupyter Notebook to submit your solution.