AMERICAN UNIVERSITY OF ARMENIA
College of Science and Engineering
CS 260 / 371 Image Processing

HW2 – Face-Specific RGB

Deadline: Thursday, August 01 2024, no later than 22:00 **SHARP**
Textbook: W. Burger, M. J. Burge. "Digital Image Processing: An Algorithmic Introduction using Java", 2nd ed., 2016

Task 1: Download from **\Home Works** subfolder of the course Moodle site **RGB.xlsx** workbook, and enter in **B1** cell of its **Ovals Manual 11** worksheet the computed starting index from **Task 0** of **HW1**. For each of the indicated in the **Ovals Manual 11** worksheet *.jpg files construct a binary mask as following:

1. Apply a threshold based on the Hue, Saturation and Brightness values normalized from 0 to 255: draw pixels in black, if **3 ≤ Hue ≤ 24 AND 40 ≤ (Saturation)(Brightness) ≤ 80**, and in white – otherwise. You may use **HSV_Threshold.java PlugInFilter** from **\Home Works** subfolder;
2. Make it a binary image. Use **Process → Binary → Make Binary** command;
3. Invert the image. Use **Edit → Invert** command;
4. Start a new macro by **Plugins → Macros → Record…** command. Apply an oval selection as to select the facial region and ears, if visible. Exclude hairs, clothes, background pixels, etc. If the neck region is separated from the main facial region, exclude the neck as well. Once the oval selection is finalized, fill its parameters in the table in **Ovals Manual 11** worksheet of the **RGB.xlsx** workbook;
5. Invert the selection. Use **Edit → Selection → Make Inverse** command. Delete the inverted selection.
6. Save the produced binary facial masks in *.png format in subfolder **\HW2\mask\** of your repository.

Task 2: Extract the facial pixels in all images indicated in **Ovals Manual 11** worksheet by computing the **AND** operation with the corresponding mask. Use **Process → Image Calculator** command and select **AND** operation. Save the masked images with the extracted facial pixels in *.png format in subfolder **\HW2\face\** of your repository.

Task 3: Write a **PlugIn** to test a hypothesis that states a simple dependence of the red and blue components **(R + B) / 2** on the green component **G** in face-specific colors. Implement the following functionality:

1. The plugin collects all non-black face-specific colors from all masked images from **\HW2\face\** subfolder in a **TreeSet<Integer>** set, hence, keeping only the unique colors;
2. For each value of the green components 0 ≤ **G** ≤ 255, it computes the following quantities: **count[G]** – the amount of different colors in the set with the green component **G**, **min[G]** – the minimal value of **(R + B) / 2** among all colors with the green component **G**, **max[G]** – the maximal value of **(R + B) / 2** among all colors with the green component **G**, **mean[G]** – the mean value of **(R + B) / 2** over all colors with the green component **G**, **mean2[G]** – the mean value of **$((R + B) / 2)^2$** over all colors with the green component **G**;
3. Outputs all 256 values of **count[G]**, **min[G]**, **max[G]**, **mean[G]** and **mean2[G]**.

Fill in the table in worksheet **RB(G) 11** with the output results.

Submission Conditions:

1. This is an individual assignment. Identical or similar submissions / files / results / reports / diagrams etc. will be disqualified – both the source(s) and receiver(s) will collect 0 point.
2. Group work will be accepted only if all group members are explicitly indicated in the submission. The individual contribution of each group member must also be explicitly stated, including all reasons of forming the group.
3. The submission deadline is rigidly strict. Submit even an unfinished work to get points and feedback. Late submissions will be disqualified and collect 0 point.
4. Not only precise solutions, but also free-format descriptions of ideas, difficulties, algorithms, simplifications, assumptions, etc. may be submitted.
5. You are welcome to use external sources, but all of them must be explicitly acknowledged and the links / references provided.