Assignment 4 - Skin Detection

Start Assignment

Due Tuesday by 11:59pm **Points** 100 **Submitting** a file upload **File Types** zip
Available Sep 21 at 2pm - Oct 1 at 11:59pm 10 days

Task 1 [100 Points]:

Write a script, saved in a file called task1.m, that evaluates three skin detection modules on some test data, and plot their performance:

- **Module 1:** Module 1 uses a distribution for P(color | skin) that is trained based on training samples from image **training_A.bmp (https://canvas.txstate.edu/courses/1798576/files/202814233/download?download_frd=1)**. The distribution is a Gaussian distribution, defined on normalized rg values, NOT unnormalized RGB values. The training patches (subwindows) used for building the distribution should be:
  - Patch 1: top = 122; left = 297; bottom = 150; right = 338;
  - Patch 2: top = 246; left = 117; bottom = 265; right = 145;
- **Module 2:** Module 2 uses a distribution for P(color | skin) that is trained based on training samples from image **training_B.bmp (https://canvas.txstate.edu/courses/1798576/files/202814243/download?download_frd=1)**. As for module 1, the distribution is a Gaussian distribution, defined on normalized rg values, NOT unnormalized RGB values. The training patches (subwindows) used for building the distribution should be:
  - Patch 1: top = 189; left = 213; bottom = 230; right = 293;
  - Patch 2: top = 376; left = 63; bottom = 423; right = 120;
  - Patch 3: top = 413; left = 404; bottom = 471; right = 455;
- **Module 3:** The detect_skin function, called with histograms read from positives.bin and negatives.bin. The following is an example of how to call the detect_skin function:

```
negative_histogram = read_double_image('negatives.bin');

positive_histogram = read_double_image('positives.bin');

frame20 = double(imread('frame20.bmp'));

result = detect_skin(frame20, positive_histogram,  negative_histogram);
```

Note: the functions read_double_image and detect_skin used in the above code snippet can be found in the 00_common code folder provided in the resources.

Evaluation

All modules should be evaluated on test windows from image **test.bmp** **(https://canvas.txstate.edu/courses/1798576/files/202814241/download?download_frd=1)**. In particular, the following patches (subwindows) should be used to evaluate accuracy:

- Skin Patch 1: top = 119; left = 299; bottom = 177; right = 333;
- Skin Patch 2: top = 224; left = 229; bottom = 261; right = 253;
- Skin Patch 3: top = 224; left = 387; bottom = 278; right = 410;
- Non-skin Patch 1: top = 332; left = 36; bottom = 457; right = 570;

Evaluation should be done as follows: for many different thresholds, compute:

- X(threshold) = the percentage of skin pixels, among pixels of all skin patches, that are correctly classified as skin. NOTE: For each threshold, compute a single number for skin pixel accuracy, NOT a separate number for each skin patch.
- Y(threshold) = The percentage of non-skin pixels, among pixels of the non-skin patch, that are correctly classified.

*Note: a sample solution to the above step can be found in **eval_module.m** **(https://canvas.txstate.edu/courses/1798576/files/202814229/download?download_frd=1)**

After computing many such values X(threshold) and Y(threshold), plot those numbers against each other, with the X values on the plot to generate the figure. Your script should display such a figure with the X values on the x axis, and the Y values on the y axis. Use the built-in Matlab function plot to plot the results.

The values that you plot for each module should include the following X values (given as percentages): 0, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95, 100. Naturally, for each of those X values you should also compute and show the corresponding Y value. Feel free to use additional values, but at a minimum compute the X values specified above. **plot_sample.png** **(https://canvas.txstate.edu/courses/1798576/files/202814237/download?download_frd=1)** shows an example of a possible output.

In your README.txt file, discuss how the different modules compare to each other in performance.

IMPORTANT: Look at the attached **plotting_example.m** **(https://canvas.txstate.edu/courses/1798576/files/202814239/download?download_frd=1)** example to see how you can plot your results in Matlab. Adapt it accordingly.

IMPORTANT: your code should assume that all images that need to be read are located in the current directory.

Task 2 (optional, 20 points extra credit): Build skin-color histograms using the training patches specified for image training_B.bmp. Evaluate histograms in RGB space. Try different numbers of bins, and report (in the README.txt file) which number of bins gave the best results. Write a script called task2.m that builds, directly from the training patches, the skin color histogram that gave the best results for this task. Your script should also produce a figure that compares the results of this best histogram with the results of the three modules of Task 1.

How to submit

Submit your assignment solution through Canvas. The submission should include, as an attachment, a zip file containing your Matlab code and a README.txt file. The zip file should be named NetID_lastname_firstname.zip. Your solution should definitely contain:

- A Matlab file called task1.m that implements the solution for task 1.
- If you did task 2, a Matlab file called task2.m that implements the solution for task 2.
- The README.txt file should contain the name and Net ID of the student, in addition to any additional comments/instructions useful for running the code and understanding the underlying ideas. The README.txt file should also discuss how the different modules performed compared to each other.

We try to automate the grading process as much as possible. Not complying precisely with the above instructions causes a significant waste of time during grading, and thus points will be taken off for failure to comply, and/or you may receive a request to resubmit.

Submission checklist

- Was the attached zipped file called NetID_lastname_firstname.zip?
- Did you include a README.txt file, as specified?
- Was the attachment zipped? We will not accept .rar, .tar, .gz, or any other filetypes, and we will not accept submissions where multiple files are attached separately.
- Did you make sure that the attachment includes only code and text files, and NO IMAGE files?
- Did your code assume that all images that need to be read are located in the current directory?

Additional resources for assignment:

training_A.bmp (https://canvas.txstate.edu/courses/1798576/files/202814233/download?download_frd=1)

training_B.bmp (https://canvas.txstate.edu/courses/1798576/files/202814243/download?download_frd=1)

test.bmp (https://canvas.txstate.edu/courses/1798576/files/202814241/download?download_frd=1)

positives.bin (https://canvas.txstate.edu/courses/1798576/files/202814227/download?download_frd=1)

negatives.bin (https://canvas.txstate.edu/courses/1798576/files/202814235/download?download_frd=1)

plotting_example.m (https://canvas.txstate.edu/courses/1798576/files/202814239/download?download_frd=1)

plot_sample.png (https://canvas.txstate.edu/courses/1798576/files/202814237/download?download_frd=1)

eval_module.m (https://canvas.txstate.edu/courses/1798576/files/202814229/download?download_frd=1)