HW2 – submission 27.12.17 23:55

Guide lines

1. Include all your personal details including name, id, and e-mail address.
2. You should submit all function and script files written in MATLAB or Python. Your code should be well documented and clear. The code should run from any computer and include all path definitions (You should take care of this in the code).
3. Please divide the code by questions.
4. Final report – should include explanations on the implementation and the execution, answers to the questions, results, conclusions and visual results. Do elaborate on all parts of the algorithms/solution. **Please submit a PDF file and not a DOC file.**
5. Please post question regarding this HW on the facebook group: https://www.facebook.com/groups/294298727746314/
6. The grades are highly depended upon the analysis depth of the report.
7. HW can be submitted in pairs.
8. Eventually submit one compressed file including the code + images PDF.

   Good luck!
Task 1 – Saliency detection (50 points):

In this question you are asked to implement a simple version of the saliency algorithm we describe in the tutorial. The relevant paper is: “What Makes a Patch Distinct?” (attached). For running and evaluating this task, you are asked to pick an image of your own. Follow the instructions below.

1. **Structure map – based on patches PCA**
   a. Read an image (pick an image of your own) and convert the image to LAB space (use rgb2lab).
   b. Convert the image to all its overlapping patches of size $k \times k$. If the input image is of size $h \times w$ then output is $N \times k^2$ where $N = (h - k + 1) \times (w - k + 1)$ (You can use im2col). This should work separately for each color channel. Use $k = 5$.
   c. Compute the PCA matrix (using princomp MATLAB function for example, or the similar python function e.g. matplotlib.mlab)
   d. Project all patches to the new PCA space. Compute the L1 norm of the pca representation of each patch. Resize to get the structure saliency map of size $h \times w$ (use padding if needed). Read section 2.1 for additional details.
   e. Repeat for each image channel in LAB color space. Present the structure saliency map of each color channel. The final structure map is the normalized sum of the three channel’s maps. Present the final map.

2. **Global color map – based on super pixels and color**
   a. Convert the image to LAB space.
   b. Use SLIC (or any other super pixel method) – you do not need to re-implement. For MATLAB you can use vl_slic. For python you can also use vbf or search the web for slic e.g. slic-python. (We suggest to use regionsize = 16, regularization = 300 and minRegionSize = 16)
   c. Compute the mean intensity (and color) for each super pixel in each channel (L,a,b), the output should be $m \times 3$, where $m$ is the number of super pixels.
   d. Compute the distance matrix $D$ between each super pixel to all the other super pixels. You can use the matlab function pdist. The output should be a distance matrix of size $m \times m$
   e. Build the color saliency map: Each super pixel is assign with the mean absolute distance from all other super pixels: $\text{mean(abs}(D))$. Each pixel in the image will get the color saliency score of its super pixel. The output is a matrix with the size of the input image $h \times w$. Read section 2.2 for additional details.

3. Normalize the color and structure maps between 0 to 1 and multiple them to get the final saliency map.
4. Analysis:
   a. Compute the structure and color saliency maps at 3 different scales (i.e. pyramid) and compare. Show also the final saliency map of each scale.
   b. Show the structure and color saliency maps for 3 different images, when does the color fails? When the structure fails? Show also the final saliency map for each image.
   c. Compare and discuss the final result for 3 different $k$ values (for a single image)
   d. Suggest a way to improve the run time of the algorithm
   e. Suggest a way to evaluate your algorithm. What is missing? What are the challenges?
Task 2 – Laplacian pyramid for style transfer (50 points):

On this question you will implement a style transfer for headshot portraits, inspired by the paper attached to this exercise "Style Transfer for Headshot Portraits" (attached).

The goal: transfer the style of an example headshot photo onto a new one. This will be done by transferring the local statistics of the example portrait at different scales onto a new one. By that, we could match the properties of the input image (such as the local contrast and the overall lighting direction) to the given example image, while being tolerant to the natural differences between the faces of two different people.

(a) Implement a function that decomposes a gray-level image to its Laplacian pyramid. The function should accept an input image $I$ and the number of pyramid levels $n$, and should return the pyramid’s levels $L_l[I]$. The pyramid level $L_l[I]$ are defined as follows:

$$L_l[I] = \begin{cases} 
I - I \ast G(2^l), & l = 1 \\
I \ast G(2^l) - I \ast G(2^{l-1}), & 2 \leq l < n \\
I \ast G(2^n), & l = n 
\end{cases}$$

Where $G(\sigma)$ is a 2D Gaussian kernel with a standard deviation of $\sigma$ and $\ast$ is the convolution operator. Note: On this Laplacian pyramid do not use down sampling (all the pyramid levels have the same size).

Tips:
- You may first construct the Gaussian pyramid of the image, and then construct the Laplacian pyramid by calculating the differences between the Gaussian pyramid levels.
- You may use the function `fspecial` to create the filter $G$. Make sure that the matrix representing the filter is about 5 times larger than the filter width $\sigma$.
- You may use the function `imfilter` to perform convolution.

(b) Implement a function that reconstructs an image from its Laplacian pyramid. The function should get the pyramid levels and should return the reconstructed image.

Test: take an image, construct its Laplacian pyramid using the function of section (a) with $n=6$ levels, and reconstruct the image using the function of (b). Is the reconstruction accurate? In your answer, discuss how using down sampling would effect the accuracy of the reconstruction.
In the following sections, you will implement the style transferring process. This is done by multiplying each of the levels of the input image Laplacian pyramid by a gain, which depends on the proportion between the input image pyramid and the example image pyramid. For sections (c)-(f) use the image “data\Inputs\imgs\0004_6.png” as the input image $I$ and “data\Examples\imgs\6.png” as the example image $E$. Use the images with a dynamic range of $[0\ 1]$ \textit{(i.e. use im2double)}.

(c) **Background:** Before transferring the style we would like to change the background of the input image to be similar to the background of the example images. Each of the example images has a corresponding background image (located at “data\Examples\bgs”). In addition, each input image has a binary mask (located at “data\Inputs\masks”). This mask has values of “0” at pixels correspond to the background and “1” at the face. Implement a function that accepts an input image, its mask and the example background. See example below:

![Input image with Original Background](image1)
![Input image with Example Background](image2)

(d) **Calculate the energy and the Gain:** construct the Laplacian pyramid of both the Input image $I$ (with the new background) and the Example image $E$ with $n=6$ levels. **Construct a separated pyramid for each of the image channels** $c$ ($c=R,G,B$ – three pyramids for each image) using the function that was implemented in section (a) . For each pyramid and for each level $l$, calculate the local energy $S_l$ according to:

$$S_l[I^c] = \left( L_l[I^c] \right)^2 \otimes G(2^{l+1})$$

Finally, calculate the gain map of each level as:
\[ Gain^c_l = \sqrt{\frac{S_l[E^c]}{S_l[I^c] + \epsilon}} \]

Where \( \epsilon = 10^{-4} \). Clip the Gain of each level to be with maximal value of 2.8 and minimal value of 0.9.

(e) **Construct the output image pyramid:** The output image \( O \) is constructed from a new pyramid:

\[ L^c_l[O] = \begin{cases} 
Gain^c_l \times L^c_l[I] & 1 \leq l < n \\
L_l[I^c] & l = n 
\end{cases} \]

For each of the output image channels (RGB) construct a new pyramids, according to the formalism above. Note: the last pyramid level equals to the last level of the example image.

(f) **Reconstruct the output image:** using the reconstructing function of section (b), reconstruct the RGB channels of the output image from their corresponding pyramids. Fuse the three channels to create an RGB image and present the results. The get better results, replace the background of the output image as in section (c).

(g) Repeat this process for transferring the style of images 16 & 21 to the input image 0004_6.png, and of the images 0,9,10 to the image 0006_001.png. Present all the results.

(h) Run the algorithm on another input or example image which was not given in the data files.

Example for the style transfer result:

Remark: replicate padding was used on some of the example images, to insure all images have the same size.