Image Processing (EE 5720)
Final Examination

Instructor: Chia-Hsiang Lin
December 24, 2020. Time: 15:10~18:00

Total score 100 points

In image processing, image inpainting is a key technology to restore an incompletely acquired image for subsequent analysis and applications. Image inpainting can be considered as Tensor Completion Problem (TCP), which also draws considerable attentions in cross-disciplinary research areas, such as wireless communications as well as bioinformatics.

In this final examination, you are asked to implement the inpainting algorithm introduced in Chapter 7, *"Image Inpainting in Computer Vision and Matrix/Tensor Completion in Wireless Communications"*. The main algorithmic flow (without math details), together with the guide of data preparation, is summarized here to help you get a quick access:

- **(INPUT)** Add stripes (or other missing patterns) to the image used in your midterm examination, which is viewed as a 3-way tensor (i.e., $\boldsymbol{Y} = \boldsymbol{AS} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]$ in my lecture notes). To this end, please set the entries of those missing data in the image/tensor as "0" (displayed as a black areas), where each entry is defined by a pixel and a spectral band. Please also record the set $\boldsymbol{\Omega}$ of entries of available data to be used in the recovery stage.

  Then, this stripe-corrupted image (i.e., $\boldsymbol{Y}_\Omega$ in my lecture notes) will serve as the input of your implemented algorithm.

- **(STEP 1)** Identify those complete pixels in $\boldsymbol{Y}_\Omega$, and feed them into a spectral unmixing algorithm (e.g., HyperCSI) to estimate $\boldsymbol{A}$.

- **(STEP 2)** For each pixel $\boldsymbol{y}_n$, $n = 1, \ldots, L$, use the estimated $\boldsymbol{A}$ (obtained above), the information of $\boldsymbol{\Omega}$ (recorded above), as well as Lemma 3 in my lecture notes, to compute $\boldsymbol{s}_n$. By collecting all the $\boldsymbol{s}_n$'s, one can obtain the estimate of $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L]$.

- **(OUTPUT)** Fusing the information of the estimated $(\boldsymbol{A}, \boldsymbol{S})$ to reconstruct the image $\boldsymbol{Y} = \boldsymbol{AS}$.

Below are some hints and guides:

1. **[Hint 1]** In Step 2, to compute the Barycentric Coordinate (BC) of a point $\boldsymbol{p}$ with respect to (w.r.t.) a simplex $\text{conv}\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N\}$, one can first convert the simplex into N hyperplanes $\mathcal{H}_1, \ldots, \mathcal{H}_N$. By applying Proposition 3 in my lecture notes, the BC can then be computed from $(\boldsymbol{p}, \boldsymbol{\alpha}_i, \mathcal{H}_i)$, where the hyperplane $\mathcal{H}_i = \{\boldsymbol{x} \mid \boldsymbol{b}_i^T \boldsymbol{x} = h_i\}$ can be parameterized by the normal vector $\boldsymbol{b}_i$ and the inner product constant h_i.

2. **[Hint 2]** If you want to apply Proposition 3 in my lecture notes, you need to perform Principal Component Analysis (PCA) to represent $(\boldsymbol{p}, \boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N)$ on a 2-dimensional affine hull (cf. $N = 3$); otherwise, the above hyperplane representation would be invalid. Note that using PCA does not change the BC because the principal components form an orthonormal basis.

3. **[Hint 3]** Besides the computational speed, the key factor to evaluate your implementation is the percentage of the missing data, upon successful recovery. Specifically, the complete image $\boldsymbol{Y} \in \mathbb{R}^{M \times L}$ has $M = 224$ spectral bands with $L = 10,000$ pixels, so you have $2,240,000$ entries in total. In case that your algorithm input $\mathbf{Y}_\Omega$ contains $|\boldsymbol{\Omega}| = 2,000,000$ entries, then your score will be positively correlated to the percentage $\left(1 - \frac{|\boldsymbol{\Omega}|}{ML}\right) \times 100\% = \frac{2{,}240{,}000 - 2{,}000{,}000}{2{,}240{,}000} \times 100\% \approx 10.71\%$. So, try your best to minimize $|\boldsymbol{\Omega}|$.

4. **[Hint 4]** For typical missing pattern, the computational time for recovering the image of such scale is just around seconds on normal desktop computing facilities. If your algorithm runs too slow, your score will also be limited. So, if your algorithm runs more than 3 minutes, please check your implementation carefully.

5. What you need to do:

   - Implement the image inpainting algorithm introduced in my lecture notes.
   - Generate the data $\mathbf{Y}_\Omega$ based on the aforementioned instructions. *(Very Important)* Please set the missing entries as "0"; otherwise, our verification method may go wrong!
   - Process the image $\mathbf{Y}_\Omega$ using the inpainting algorithm you implemented.

- Give an oral presentation (8 minutes) on December 24, 2020, followed by an Q&A session (2 minutes). *(See below for how to prepare your slides.)*

6. What you need to submit:

   - A "Group_t.pptx" file ($t \in \{1, \ldots, 36\}$ is your team number), including a set of slides organized as follows:
     - (Page 1) A cover page showing what "each" group member has done.
     - (Page 2) Show the computational time T (in seconds), and the percentage of the missing data $\left(1 - \frac{|\boldsymbol{\Omega}|}{ML}\right) \times 100\%$ you truly achieved (upon successful recovery).
     - (Pages 3 to *End*) Detailed discussion. *[e.g., a) list all difficulties you have encountered when implementing the algorithm, b) explain how you solve those difficulties, c) share any novel idea or new algorithm designed by yourself (optional), etc.]*

     If you need more slides for "insightful" discussion, feel free to do so, but keep in mind that you get only "8 minutes" for your oral presentation.
   - A completed "Algorithm_t.m" file (i.e., the MATLAB code that you have implemented).
   - The best incomplete image $\boldsymbol{Y}_\Omega \in \mathbb{R}^{100 \times 100 \times 224}$ you achieved (i.e., the image with smallest cardinality $|\boldsymbol{\Omega}|$, upon successful recovery). Please submit the image as a 3-dimensional MATLAB DATA, named "Data_t.mat".

   Please submit the three files (i.e., "Group_t.pptx", "Algorithm_t.m", and "Data_t.mat") to Jhao-Ting Lin (christtlin870117@gmail.com) and Pang-Yu Lin (brian1997081@gmail.com) by the end of December 17, 2020. Please use "Final Examination EE 5720 (Group t)" as the email title.

7. What we will do after receiving your files:

   - We will identify those "0" entries (missing data) in "Data_t.mat", based on which we will know your definition of the support $\boldsymbol{\Omega}$ (available data). Accordingly, we will verify the percentage $\left(1 - \frac{|\boldsymbol{\Omega}|}{ML}\right) \times$ 100% you provided in page 2 of "Group_t.pptx".

We will also check if those non-zero entries in "Data_t.mat" remain the same as the corresponding entries in the data we provided to the tth group.

- To confirm that your implementation does successfully recover the image $\boldsymbol{Y}$ from its incomplete version $\boldsymbol{Y}_\Omega$, we will directly feed "Data_t.mat" into "Algorithm_t.m" to get the algorithm output denoted by $\widehat{\boldsymbol{Y}}$ (note that you have to convert the algorithm output as a $100 \times 100 \times 224$ tensor accordingly to the form of data we provided).

  Then, we will compute the error as the Frobenius norm $\|\boldsymbol{Y} - \widehat{\boldsymbol{Y}}\|_F$ to see if the error is zero (it is acceptable if the error is very small (e.g., 10^{-12}) due to the inevitable numerical errors).
- We will check to confirm that your implementation does not use any information other than $\boldsymbol{Y}_\Omega$.
- We will run all the implementations "Algorithm_t.m", for all $t \in \{1, \ldots, 36\}$, on the same computer facility for fair comparison of the computational time T.
- We will crosscheck the codes submitted from all the 36 groups to judge whether you have implemented the algorithm by yourself.

8. If you achieve (some of) the following goals, you will be graded with high scores:

   - Propose a *new* idea for image inpainting. You need to *clearly* and *concisely* illustrate your idea during your oral presentation. A graphical illustration (without math details) would also be good.
   - Implement your new idea with successful experimental results to prove that your idea is implementable. If you opt to do so, you need to submit your implementation to TAs as well.

   If you achieve any of these goals, please explicitly mention it during the oral presentation.

9. Plagiarism is not allowed, and if found final exam gets 0%.

10. You must hand in all the three files by the deadline, so that we have sufficient time to evaluate your works. Your score will be multiplied by a factor of 0.9 per day passing the deadline.

11. If you have any question, please let us know by email or just come to Lab 92A53 or Lab 92931 (EE Building) for a discussion.