ECE763 Computer Vision: Models, Learning and Inference (including Deep Learning)

Standard Project 01

(Grades: 100 points in total)

Depart. of ECE, NC State University

Instructor: Tianfu (Matt) Wu

NC STATE UNIVERSITY Electrical and Computer Engineering

Discussions on Projects

- If applicable, identify problem(s) in your own field that have images/videos generated and to be analyzed, and collect dataset(s) (e.g., publicly available ones, or from your own research)
  - Formulate the problem(s): e.g., input-output mapping, target platforms (edge device or workstation/server), evaluation metrics
  - Find a few references in top-tier conferences/journals
  - Our goal: to have a submission ready by the end of this class, through customized project 1-3 design
- If no specific domains of interest,
  - Check https://paperswithcode.com/
  - Find problem(s) that interest you
  - Survey SOTA methods therein
  - Our goal: to have a submission to top-tier CV conference ready by the end of this class, through customized project 1-3 design
- If both do not work for you, **start working on this one!**
  - We will have standard project assignments.

Grading Policy

- **Homework**: 7% x 5 = 35%
- **Projects**: 60% + [5% bonus]
  - Project 1 (10%) & 2 (20%): **work individually [not required, see the thrust 1 and 2 in the next slide)**
  - Project 3 (course project): 30%, **a group (<= 3 members) is allowed [not required, see the thrust 1 and 2 in the next slide)**
    - You can propose your own projects and work on it once agreed by both of you and me.
    - Requirement: final write-up (15%) and self-contained reproducible code (15%)
    - Bonus points (5%): for top-3 winner of the final project
- **Attendance***:* 5% (via in-class moodle quiz, 10min before and 15min after class, from the 2nd week)
- **Late policy**
  - 5 free late days – use them in your ways (counted using 0.5 unit, <=6 hours as 0.5 late day, otherwise 1 later day)
  - Afterwards, 25% off per day late
  - Not accepted after 3 late days per HW and Project 1 & 2
  - Does not apply to Project 3 (no late assignments allowed).
  - We will **NOT** accept any replacement of submission after deadline, even if you can show the time stamp of the replacement is earlier than the deadline. So, please double-check if you submit correct files.
- **Other policies**: please see details in the syllabus.

Instructions and Notes

- *How to submit your solutions*: put your report (word or pdf) and results images (.png) if had in a folder named [your_unityid]_hw01 (e.g., twu19_hw01), and then compress it as a zip file (e.g., twu19_hw01.zip). Submit the zip file through **moodle**.

- *If you miss the deadline and still have unused late days, please send your zip file to TAs and me.*

- **Important Note:** We will **NOT** accept any replacement of submission after deadline ([+late days you use]), even if you can show the time stamp of the replacement is earlier than the deadline. So, **please double-check whether you submit correct files**.

Project 01

- **Objectives:** Face image classification using Gaussian model, Mixture of Gaussian model, t-distribution, Mixture of t-distribution, Factor Analysis and Mixture of Factor Analyzer

- **Importance of working on Project 01 individually**: Getting hands-on experience is important, which benefits you the most if you play with it individually and independently.

Project 01-Data Preparation

- **<u>Data Preparation.</u>** Prepare training and testing data. Go to https://github.com/betars/Face-Resources and select one of the provided 17 face datasets which has face bounding boxes annotated. Download the dataset (note that some of the 17 datasets might need registration to download and you can skip those to save time). Extract $n = 1000$ training images for face and non-face respectively, and $m = 100$ testing images for face and non-face respectively, both at 20×20 resolution (resizing accordingly, you can also try 10×10 and 60×60 to see how robust and efficient your codes are). Make sure training face images and testing face images are separate, that is no face testing images are from the same person in the training set of face images. And, non-face images should be cropped randomly from background in the provided images in the dataset you selected.
  - *Organized the extracted image patches Write a self-contained data i/o module in python or matlab or c/c++.*

Project 01-Data Preparation

- Why doing this step? To get familiar with loading datasets, extracting image patches based on provided annotations. The datasets usually provide some toolkit for manipulating image and annotation which you can re-use and modify. This is the first step in almost of computer vision problems.
- You can try to use more training and testing images at your will.
- You can also try to use smaller or bigger patches so you need less or more computing power and memory footprint, depending on your laptop or workstation.
- You might need to extract more images initially, then manually prune those to make your dataset less messy. E.g., containing relatively clean faces, frontal and profile, without different types of occlusions (e.g., big glasses, mask, etc).

Project 01-Task

- With your own face dataset created, you can train your models and test the performance. **For each model, report results as follows.**
  - *Visualize the estimated mean(s) and covariance matrix for face and non-face respectively*; Use RGB images, but you are welcome to try on other things such as gray images, gray images with histogram equalized, and HSI color space, etc.
  - *Evaluate the learned model on the testing images using 0.5 as threshold for the posterior. Compute false positive rate (#negatives being classified as faces / #total negatives), and false negative rate (#positives being classified as non-face / #total positives), and the misclassification rate ((#false-positives + #false-negative)/#total testing images)*
  - Here, the threshold 0.5 means
    - $p(w=1|x) = \frac{p(x|w=1)p(w=1)}{p(x|w=1)p(w=1)+p(x|w=0)p(w=0)} > 0.5$
    - First assume $p(w=1) = p(w=0) = 0.5$ (balanced two-class). Then, it means,
    - $p(x|w=1) > p(x|w=0)$
    - And, to eliminate the computational cost (numeric instability), we resort to compute,
    - $\log p(x|w=1) > \log p(x|w=0)$
    -
- *Plot the ROC curve where x-axis represents false positive rate and y-axis true positive rate (i.e, 1-false negative rate). To plot the curve, you change the threshold for posterior from $+\infty$ (use maximum posterior across testing images in practice, then all being classified as non-faces) to $-\infty$ (use minimum posterior across testing images in practice, then all being classified as faces) with for example1000 steps. (ref: https://en.wikipedia.org/wiki/Receiver_operating_characteristic)*

Project 01-Task

- Hint:

- Think about how to modularize your codes in a nice way. For example, you will only need one ROCplot functions, and you can write a common EM algorithm. It is important to learn how to connect understanding of mathematical derivation to actual efficient implementation.

Project 01 - Models (Required)

- **<u>Model 1 (20 points).</u>** Learn single Gaussian model using training images and report your results as stated above.
- **<u>Model 2 (40 points).</u>** Learn Mixture of Gaussian model using training images and report your results as stated above. You can tune the number of components (e.g., based on cross validation strategy).
- **<u>Model 3 (20 points).</u>** Learn t-distribution model using training images and report your results as stated above.
- **<u>Model 4 (20 points).</u>** Learn factor analyzer using training images and report your results as stated above. You can tune the dimensionality of the subspace.

Project 01 - Models (Optional)

- *Optional [10 bonus points per model]:*
  - ***Model 5.*** *Learn Mixture of t-distribution model using training images and report your results as stated above. You can tune the number of components (e.g., based on cross validation strategy).*
  - ***Model 6.*** *Learn Mixture of t factor analyzer model using training images and report your results as stated above. You can tune the dimensionality of the subspace and the number of components (e.g., based on cross validation strategy).*