CSE 515 Multimedia and Web Databases

Phase #1

(Due Sept 10th 2023, midnight)

Description: In this project, you will experiment with

- image features,
- vector models, and
- similarity/distance measures

Important: This project phase will be performed individually by each group member and each group member will submit their own deliverable . This is, nevertheless, a group activity; meaning that you can collaborate and exchange ideas within your group and you will get group grades.

PROJECT TASKS:

- **Task 0:** In this phase of the project, we will use
  - python as the programming environment,
  - pytorch as the deep learning library,
  - torchvision as the visual information extraction package, and
  - you may also need numpy and scipy for array manipulation and other mathematical operations.

  Familiarize yourself with these tools and languages. In particular, download and familiarize yourselves with the following pre-trained neural architecture and data set available through the torchvision package:

  - *pre-trained neural architecture:* ResNet50 (with default weights)
  - *data set:* Caltech 101

  In this phase, you are free to store the data however you wish: you can use a relational database (such as MySql), a no-SQL database (such as MongoDB), or create your own file/data structures.

- **Task 1:** Implement a program which, given an image ID and one of the following feature models, visualizes the image and then extracts and prints (in a human readable form) the corresponding feature descriptors:
  - *Color moments, CM10x10*: Resize image to 300×100, partition the image into 10x10 grid, for each grid cell compute three color moments (mean, standard deviation, and skewness) for each channel in the RGB color space, and combine these color moments to obtain a unified $10 \times 10 \times 3 \times 3 = 900$ dimensional feature descriptor. See

    `https://en.wikipedia.org/wiki/Color_moments`
  - *Histograms of oriented gradients, HOG*: Map the image to gray scale, resize image to 300×100, partition the image into 10x10 grid, compute 9-bin (signed) magnitude-weighted gradient histogram (each bin corresponding to 40 degrees) for each grid cell, and combine these histograms into a $10 \times 10 \times 9 = 900$ dimensional feature descriptor. You can use $\langle -1, 0, 1 \rangle$ and $\langle -1, 0, 1 \rangle^T$ masks to obtain dI/dx and dI/dy for each pixel position in the grid cell.

https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf
https://filebox.ece.vt.edu/~jbhuang/teaching/ece5554-4554/fa16/lectures/Lecture_23_ObjectDetection2.pdf

  - *ResNet-AvgPool-1024*: Resize image to 224×224; attach a hook to the output of the "avgpool" layer of the ResNet pre-trained architecture to obtain 2048 dimensional vector, reduce the number of dimensions of the vector to 1024 by averaging two consecutive entries in the vector.
  - *ResNet-Layer3-1024*: Resize image to 224×224; attach a hook to the output of "layer3" layer of the ResNet pre-trained architecture to obtain $1024 \times 14 \times 14$ dimensional tensor, convert this tensor to a 1024 dimensional vector by averaging each 14×14 slice.
  - *ResNet-FC-1000*: Resize image to 224×224; attach a hook to the output of "fc" layer of the ResNet pre-trained architecture to obtain a 1000 dimensional tensor.

  Note: Some of the images in the CalTech101 data set do not have three channels – you can skip those images.

- **Task 2:** Implement a program which extracts and stores feature descriptors for all the images in the data set.
- **Task 3:** Implement a program which, given an image ID and a value "k", returns and visualizes the most similar k images based on each of the visual model -you will select the appropriate distance/similarity measure for each feature model. For each match, also list the corresponding distance/similarity score.

Deliverables:

- Your code (properly commented) and a README file.
- Your outputs for the provided sample inputs.
- A short report describing your work and the results.

Please place your code in a directory titled "Code", the outputs to a directory called "Outputs", and your report in a directory called "Report"; zip or tar all off them together and submit it through the digital dropbox.