Sabanci University, FENS
CS419 Digital Image and Video Analysis
2024-2025 Fall semester
Assignment 4 - texture classification

I was about to give you the same assignment as last year on shape description, but then I changed my mind, as you deserve better. Let's spice it up a bit. Our topic is instead texture classification with both rotation and illumination variations. The outex12 dataset is a classic from many years ago acquired from an industrial environment. It contains 24 types (classes) of color textures.

Dataset link: https://drive.google.com/file/d/1C0AV6ebxqj7aqmlda_RP64xUysPBPbZK

Each of the 20 samples per class has been acquired at nine orientations (0, 5, 10, 15, 30, 45, 60, 75 and 90 degrees). The training set consists of 20 x 24 = 480 samples, all acquired under 2856K incandescent light at zero degrees orientation.

There are however two testing sets. Each of them has 9 x 20 x 24 = 4320 samples.

The first one (TL84) consists of samples acquired at all nine orientations, under 4000K fluorescent light. And the second set contains samples acquired under 2300K horizon sunlight.

The objective is to design a descriptor function that you'll use in order to train a model with its feature vectors, so that it'll be able to generalize to the different illumination conditions and orientations of the test sets.

Let's break it down to steps.

a) You'll use a global shape descriptor D to convert each sample into a numerical feature vector. You can use any approach you want (convolutional networks, vision transformers included). However you must test **at least** the following:

- Local binary patterns (any variant will do) **(10 points)**
- Gray level co-occurrence matrices **(10 points)**
- Fourier transform based texture descriptors **(10 points)**
- Gabor filterbanks **(10 points)**
- Combinations of the above **(10 points)**

b) For every test sample Y, you'll calculate the distance between Y's feature vector and the feature vectors of every training sample X_i, to determine the training sample that is closest to Y. You'll assign the class of the closest training sample to Y. And once you have repeated this for every testing sample you'll calculate the ratio of correctly classified test samples and report it (i.e. you’ll perform nearest neighbour classification). **(20 points)**

c) You’ll repeat all of your experiments with 4 distinct distance functions: Euclidean, Manhattan, Chi-squared, and Mahalanobis. This way you’ll be able to assess if/and by how much the distance function choice affects performance **(20 points)**.

d) You’ll investigate the effect on performance of counter-measures (if any) that you’ll take against illumination variations. **(10 points)**

e) You’ll compile all your findings into a well formatted table and report the performance of each run (separate scores for each test set). Let’s see which features are the best and how high you can get on average across the 2 test sets.

Instructions

1. Integrity: Plagiarism is strongly prohibited and may lead to failure of this course.

2. Questions: Contact the TA for any questions you might have.

3. Write-up: Please submit your answers as a zip file containing the documented **python notebook** of your implementations (**as ipynb files**) and a single **pdf file** type-set with LateX containing your answers to the various questions. Do not submit scans or photographs of handwritten documents, or pdfs prepared in word/libreffice, they will not be accepted for evaluation.

4. Collaboration: You can work in groups, however each student must submit their own work.

5. You are free to use any software library.

Good luck.