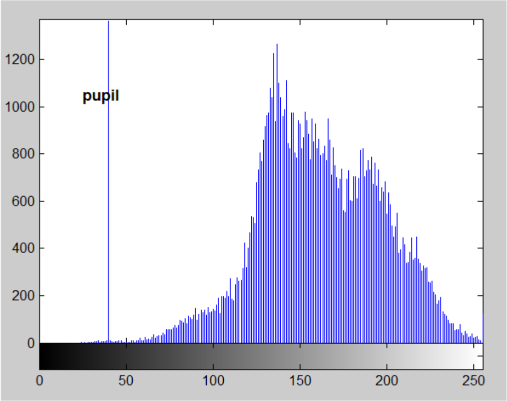
Hints for Individual Project

Hi all,

I provide some hints for this individual project below.

1. **IrisLocalization.m/py**: detecting pupil and outer boundary of iris. Refer to steps 1~3 of section 3.2.1 in Ma's paper, For pupil detection of CASIA ver1 dataset, you can use a single threshold to segment pupil, e.g. 50,



For the outer boundary of iris detection, please refer to slide #46 of "DSI_week6_IrisRecognition_2-21-2018.pdf". You can use a smaller image to speed up Hough transform for detecting a circle. For the non-circular pupil cases, the fitted circle doesn't match the exact shape of the pupil which is ok for this project.

2. **IrisNormalization.m/py**: mapping the iris from Cartesian coordinates to polar coordinates. Please refer to slide#10 of "DSI_week7_IrisRecognition_2-28-2018.pdf". Ma and Daugman's normalization methods are same except using different variables name. In Ma's paper (equation 2 on page 5), because pupil and iris are usually non-concentric, (Xp(theta),Yp(theta)) is calculated based on the pupil center. (Xi(theta),Yi(theta)) is calculated based on the iris center.

3. **ImageEnhancement.m/py**: enhancing the normalized iris. Refer to 3.2.3 of Ma's paper. For the background illumination: "the mean of each 16x16 small block constitutes a coarse estimate of the background illumination", i.e., you will get a 4x32 size image. Then you expand the 4x32 image to a 64x512 image, e.g., you can easily do it by using a matlab function imresize to expand your image, i.e., imresize(meanImg,16,'bicubic'). You can also try to skip the estimation of the background illumination, i.e., directly does enhancement by using histogram equalization on the normalized iris image, if you think the enhancement result does not benefit from the background illumination, which is optional in some iris recognition systems.

4. **FeatureExtraction.m/py**: filtering the iris and extracting features. Refer to sections 3.3.1 & 3.3.2. For the equation 4 on page 7, it is a spatial domain convolution (review “DSI_week4_BasicKnowledge_2-7-2018”). You can use existing function to conduct the filtering such as matlab function conv2(img,filter,'same'). The size of the filter can be 8x8, for example.
5. **IrisMatching.m/py**: using Fisher linear discriminant for dimension reduction and nearest center classifier for classification. Refer to sections 3.3.2 and 3.4. The dimensionality of the reduced feature vector is 200 (the original feature vector is 1,536), which is used for Table 3. Fig. 10 shows the recognition results using features of different dimensionality.
6. PerformanceEvaluation.m/py: calculating the CRR for the identification mode, which will output Tables 3 & 10; calculating ROC curve for verification mode, which will output Table 4 and Fig. 13. (refer to Ma’s paper). CRR and ROC may be different from Ma’s paper because we have only a sub-dataset, i.e., 756 iris images from 108 eyes (7 images per each eye).

There is no requirement for detecting eyelashes and eyelids etc for this project. Just follow the exact same procedure as described in Ma’s paper, e.g., on page 7, “In our experiments, we find that the upper portion of a normalized iris image (corresponding to regions closer to the pupil) provides the most useful texture information for recognition (see Fig. 7). In addition, eyelids and eyelashes rarely occlude this section. So, we extract features only in this section (called region of interest, ROI) shown as the region above the dotted line in Fig. 7.”

Please don’t forget to provide a readme file when you submit your project (refer to IndividualProject_2-28-2018.pdf).

If you still have questions, please try to find answers on Piazza, or Google it, or send us an email, or post your new questions on piazza. We suggest posting questions on Piazza so that other students can share the answer.

Good luck!

Thank you very much!

Best,

Xiaofu