AMERICAN UNIVERSITY OF ARMENIA

College of Science and Engineering

CS 260 Image Processing

Task 0

Step 1: Created a GitHub repository.

Step 2: Chosen images 151-155 from the FEI Face Database and stored them in \originalimages_part4 folder.

Task 1

Step 1: Chosen frontal images from the dataset (151-11, 152-11, 153-11, 154-11, 155-11).

Step 2: Performed plugin AllLayers on them.

Step 3: After trying several filters to produce connected and smoothened decided to perform Median filter with 10px radius for all the images.

Step 4: Performed plugin AllLayers on them. Saved file 'Task1.tif' as a result of the action.

Task 2

Step 1: Chosen xxx-11, xxx-12, xxx-13 frontal images and performed Median filter with 10px radius for them as it was done for Task1. The preprocessed images are saved in '\Task2\1 Preprocessed_images' folder.

Step 2: Created a macro (\Task2\Task2_Macro.ijm) and performed all the steps described in the task description with it.

Step 3: Saved centroid coordinates for each image as 'Centroid(xxx-yy).csv' in folder '\Task2\2 Task2_results'.

Step 3: Saved resulting images of detected central region as 'Result(xxx-y).png'. As a result, only two of the central regions over 15 images couldn't be detected correctly.

Maksim Khachatryan

Task 3

Step 1: With the aim of comparing the results at the end, chosen the same images that were chosen for the previous task, those are xxx-11, xxx-12, xxx-13 frontal images and performed Median filter with 10px radius for them as it was done for Task1. The preprocessed images are saved in '\Task3\1 Preprocessed_images' folder.

Step 2: Created a macro (\Task3\ Task3_Macro.ijm) and performed all the steps described in the task description with it. To minimize or remove insignificant details from the resulting images, additionally performed some dilate and erode filters before calling command 'analyze particles' within the macro.

Step 3: Saved resulting images in '\Task3\2 Task3_results' folder. As a result, the central regions of all the images were correctly detected. So, we can assume that the Task 3 algorithm may work better than the Task 2 algorithm. Although this hypothesis needs to be tested and validated in the future based on significantly huge dataset.

Task 4

Step 1: Chosen all xxx-01 to xxx-10 images representing different rotations of the same image xxx. Performed Median filter with 10px radius for them as it was done for Task1. The preprocessed images are saved in '\Task4\1 Preprocessed_images' folder.

Step 2: Created a macro (\Task2\Task4_Macro.ijm) with analog of Task3_Macro and performed all the steps described. To minimize or remove insignificant details from the resulting images, additionally performed some dilate and erode filters before calling command 'analyze particles' within the macro.

Step 3: Saved resulting images in '\Task4\2 Task4_results' folder. As a result, we see that the central regions of few images couldn't be detected, and the reason is the fill holes command couldn't be work because of existing some partially fully holes within the binary layers.

Step 4: Observing all the obtained images, we see that the nested ellipses gradually change their position relative to the center of the large ellipses of layer 1 depending on the rotation of the picture.

Task 5

Step 1: Chosen all the images and performed Median filter with 10px radius for them as it was done for Task1. The preprocessed images are saved in '\Task5\1 Preprocessed_images' folder.

Step 2: Created a macro (\Task5\ Task5M1_Macro.ijm) for the first method and performed all the steps described in the method 1 description with it. To minimize or remove insignificant details from the resulting images, additionally performed some dilate and erode filters before skeletonization within the macro.

Step 3: Saved resulting images of the first method in '\Task5\2 Method1' folder.

Step 4: Created a macro (\Task5\ Task5M2_Macro.ijm) for the first method and performed all the steps described in the method 1 description with it.

Step 5: Saved resulting images of the first method in '\Task5\2 Method2' folder.

Step 6: Observing all the obtained images for both methods we see that both have given good enough results for the xxx-13 and better for xxx-14 images. Most probably the dose of the light of the image affects the effectiveness of these methods. Maybe it would be better to perform histogram matching algorithm to make all other images to have the same color distribution as images xxx-14 before performing these methods on them.

Task 6

Step 1: Chosen all the images and performed Median filter with 10px radius for them as it was done for Task1. The preprocessed images are saved in '\Task6\0 Preprocessed_images' folder.

Step 2: Written a custom PlugInFilter '\Task6\LR_Image.java' for reducing image resolution 5 times.

Step 3: Obtained low resolution images for all preprocessed images using 'LR_Image' plugin. The results are saved in '\Task6\1 LR_Images' folder.

Step 4: Performed Task2_Macro on low resolution images. The results are saved in '\Task6\2 PreviousTasksOnLR\Task2_results' folder. As a result, only the centroids of 12 over 70 images could be detected. Because of low resolution some of the holes having originally narrow borders became partially fully holes and the fill holes command didn't affect. That is why most of the central regions couldt be detected.

Step 5: Performed Task3_Macro on low resolution images. The results are saved in '\Task6\2 PreviousTasksOnLR\Task3_results' folder. As a result, central nested ellipsoids of only 2 over 15 images couldn't be detected. Same algorithm was performed for the images participating in Task 4, saving the results in '\Task6\2 PreviousTasksOnLR\Task4_results' folder. And the result is same. We can assume that Task 3 can be performed for low resolution images approximately with the same accuracy as for high resolution images.

Step 6: Performed Task5M1_Macro and Task5M2_Macro on low resolution images. The results are saved in folders '\Task6\2 PreviousTasksOnLR\Task5_Results\Method1' and '\Task6\2 PreviousTasksOnLR\Task5_Results\Method2' accordingly. The results for high resolution images were better than for low resolution images, but for images xxx_14 the algorithm gives pretty close results compared with high resolution results.