// Loading a sequence of images

File.openSequence("/Users/vmanukyan/Desktop/HW2/dataset/");

// Removing small bright noise spots from all images in the stack

//Cleaning up the images by removing small, bright noise spots, improving the quality of subsequent processing

run("Remove Outliers...", "radius=2 threshold=50 which=Bright stack");

// Smooth the images to reduce noise, due multiple tests, 3 was the better choice

// Making images more smooth, emphasizing larger, more significant structures

run("Gaussian Blur...", "sigma=3 stack");

// Further reduce noise and smooth the images using a median filter

// To further reduce noise, especially impulsive noise, and smooth the images without significantly blurring the edges

run("Median...", "radius=4 stack");

// Running a custom plugin from moodle Homeworks folder

run("Compile and Run...", "compile=/Applications/ImageJ.app/plugins/HSV\_Threshold.java");

// Convert the images to 8-bit grayscale, simplifying subsequent processing

run("8-bit");

// Convert the grayscale images to black and white

// The parts of the face(which we are interested in) are white and the background is black

run("Make Binary", "calculate");

setOption("BlackBackground", true);

// Refining and sharpening the edges of objects in the binary images, making features more distinct

run("Erode", "stack");

// Selecting the oval and choosing the person with the largest face

setTool("oval");

// Define an oval selection in the image for masking or region isolation

makeOval(197, 105, 246, 245);

makeOval(197, 90, 246, 260);

// Invert the selection, selecting the area outside the drawn ovals

run("Make Inverse");

// Set the background color to black for the subsequent clearing step

setBackgroundColor(0, 0, 0);

// Clear (fill with background color) the non-selected area in the current image slice

run("Clear", "slice");

As a result, if we compare the oval coordinates and height/widht inside the Ovals Algorithm 11 with Ovals Manual 11, we would notice that the values are pretty much the same for all the images with the comparably larger difference of Top Left Coordinate Y

Also, the task2\_result contains the result of the 3rd problem of bonus task