Armin Karimi – 2080603 Lab5

1) The idea for this task this to use erode and then K-means

First of all in erodee function we convert image to grayscale, then using median blur the image for decrease the noises (on gray scale image). Now we use thereshold (15px) to separate black pixels from others. For next step we use erod to eliminate small or thin object(noises) in image and for the last step use median to blur again to remove other noises.

Then we pass the result of this function to k_means function. We apply K-means with k = 2 to reach 2 label(track OR asphalt).

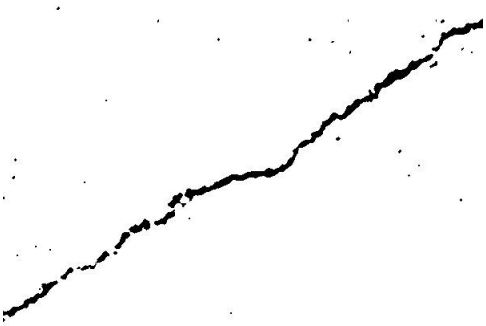
Left result (black and white images) are the outputs if we only apply erode – Right result(blue and red) are output when first erode and then apply Kmeans.

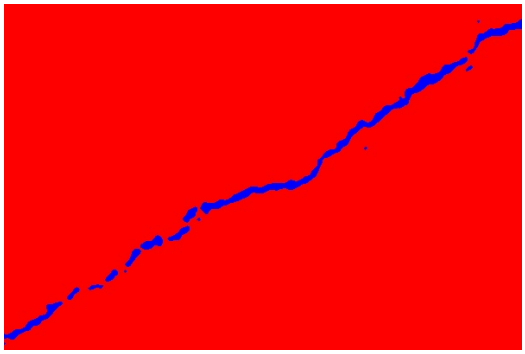
2) For this task , I use two different method. For Solution 1 I use “findContours” which is in opencv c++ to detect all the area in the image. Then I use a loop on evety area in which “findContours” function provided and find 2 the biggest area (sky and asphalt are the biggest area in the image which the question want from me). And finaly apply a mask which sky and asphat(biggest area of the “findContours” out put will be their orginal color and rest of the image will be black).

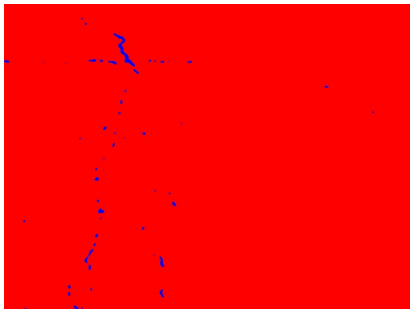
For second approach I use K-means. First we determine the termination condition with this code (“TermCriteria criteria(cv::TermCriteria::EPS + cv::TermCriteria::COUNT, 10, 1.0);”) then use k-means (“kmeans(data, K, labels, criteria, attempts, flags, centers);) and set number of cluster, termination condition , image and output image. For final step we set a color for each cluster(sky : green – asphalt: blue and everything else red).

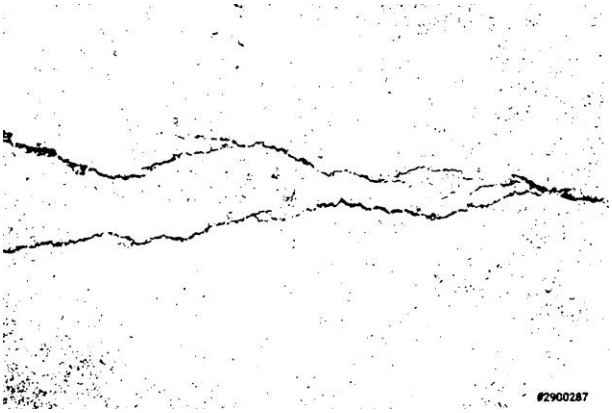
3) The main idea for this question is to create an orange mask and use it to detect only T-shirt of the robots. For this purpose we set a lower and upper bound for this color (orange) to create a mask. Using the previous lab the we had use to detect the lower and upper bound of the orange color in this image. At the end we apply the mask on the image and the output will be only orange color (for the T-shirt) and black for other images. We use “bitwise_and” and “bitwise_not” for this.

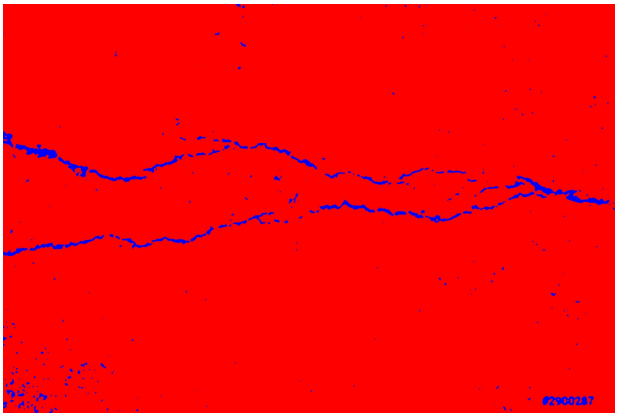
Task1)



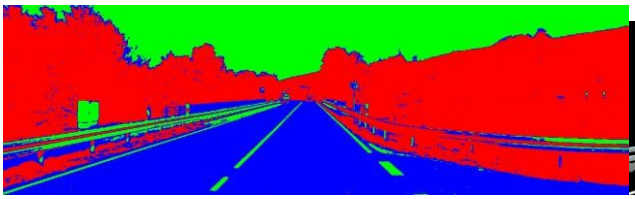








Task2)



Task3)