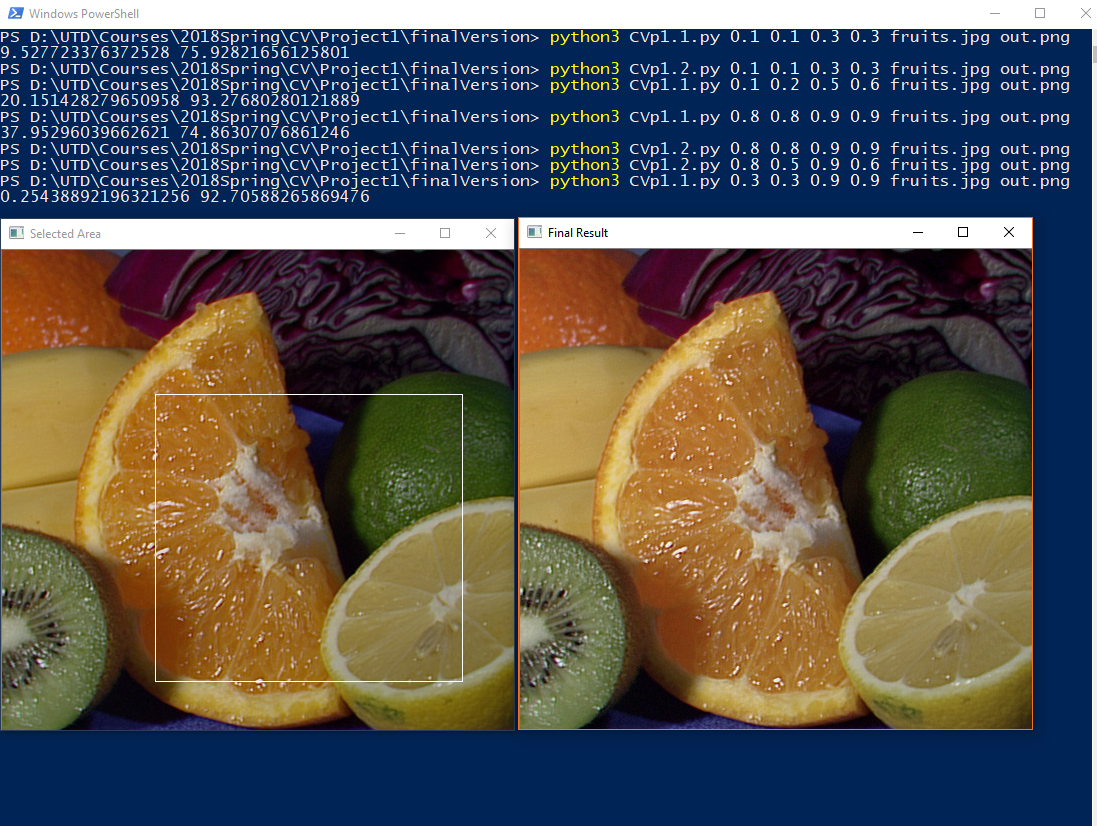
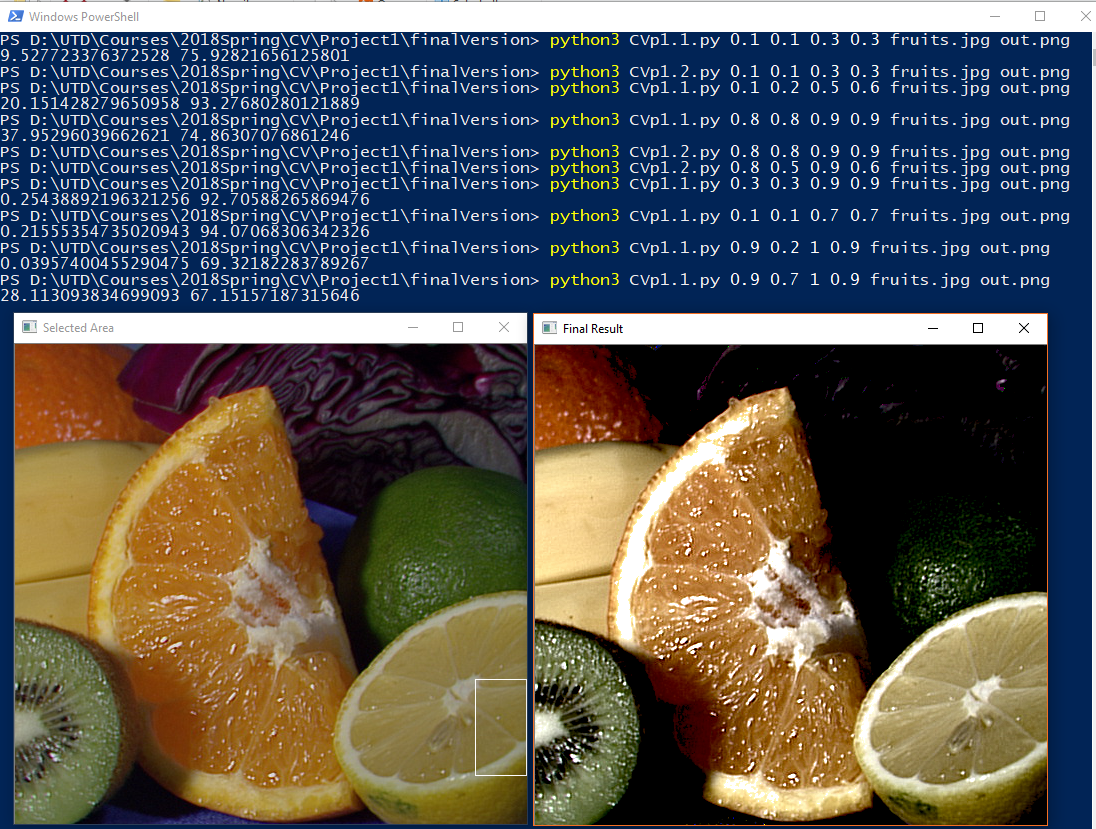
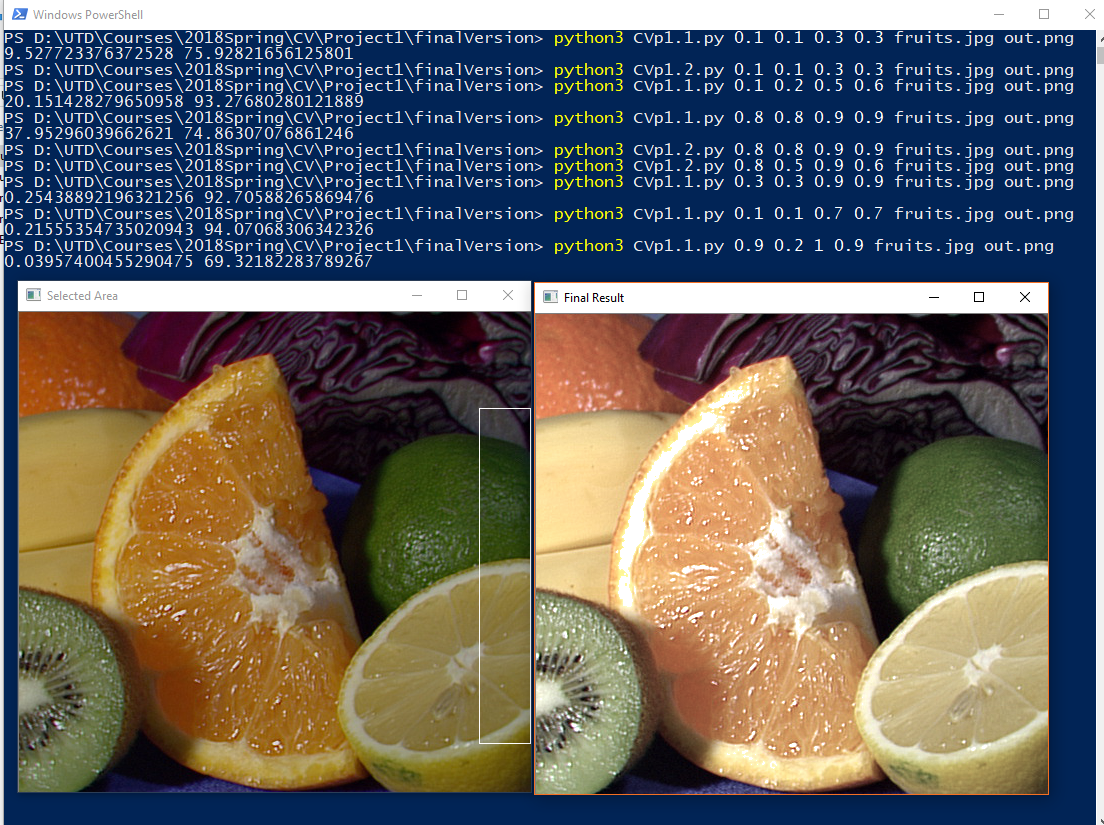
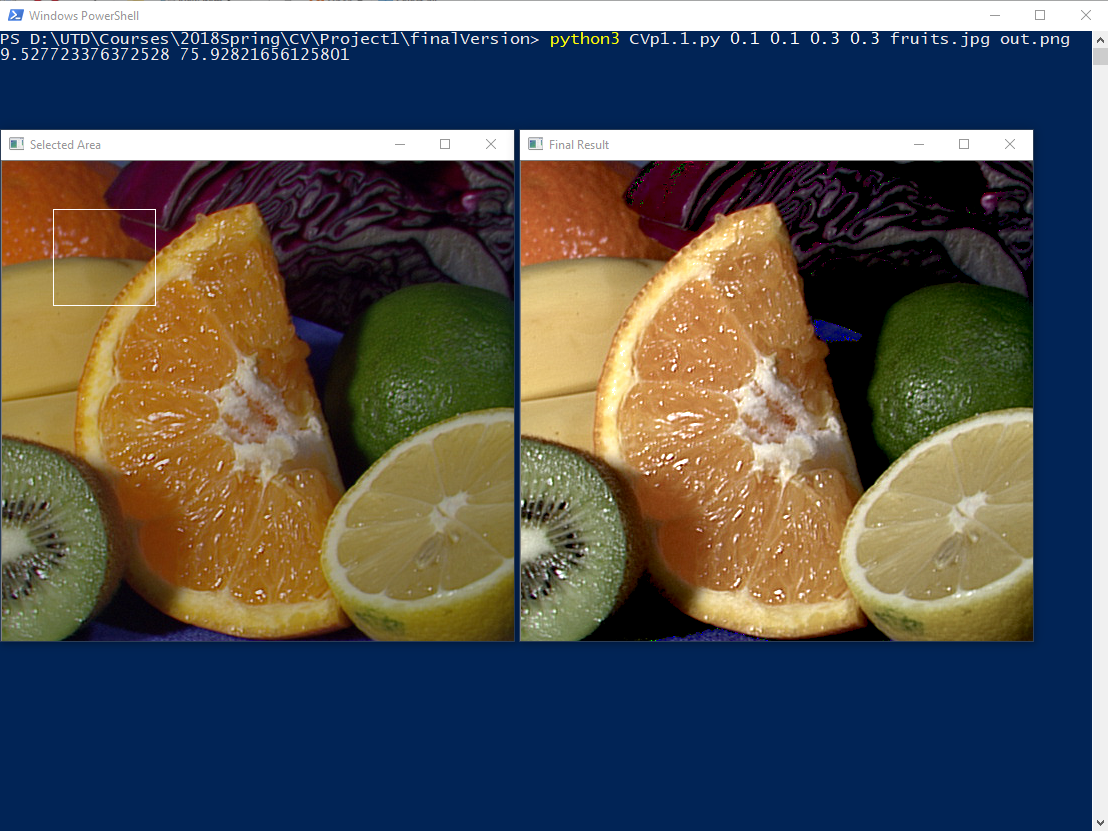
For project1, when transform XYZ to Luv, it may meet XYZ = [0, 0, 0], this will lead to d = 0, which means it will make u’ or v’ divide by 0. To avoid this situation, I chose to directly return a float type array [0, 0, 0].

Kind of same situation, when transform Luv to XYZ, the L value may equal to 0 which means will cause divide by 0 situation, so I chose to directly return a float type array [0, 0, 0].

When I select small area of the image as the calculate data set, it makes the image “bad”. The larger area I select, the better image I have. Examples are as followed:





Based on this situation, one possible reason I can guess is: when we select larger area, there are more pixels contained in that area which means we have a large range of L values than the small area. So we change less pixels’ L directly to 0 or 100 than large area. Then the result will be better.