“Computer Vision”

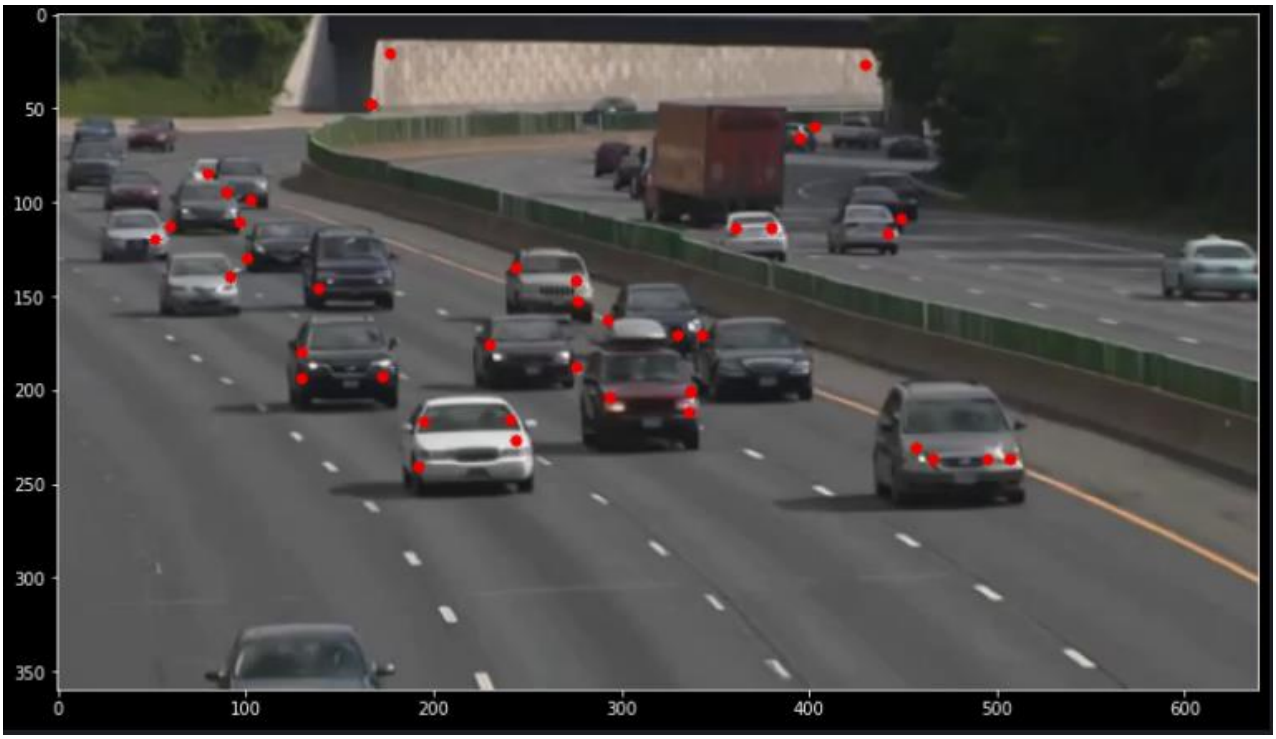
Shervin Halat

98131018

Homework 5

1.

Detected corners of the first frame using Shi-Tomasi method:



Following parameters were considered:

```
def find_corners(frame, maxCorners = 40, qualityLevel = 0.01, minDistance = 10, blockSize = 19):
    feature_params = dict(maxCorners = maxCorners,
                          qualityLevel = qualityLevel,
                          minDistance = minDistance,
                          blockSize = blockSize)
    corners = cv2.goodFeaturesToTrack(frame,mask=None, **feature_params)
    return corners
```

Parameters of cv.goodFeaturesToTrack function:

image(1st parameter):

single-channel image which is the image for which we are to detect corners.

maxCorners(2nd parameter):

Determines the maximum number of corners to return. If there are more corners that are found, the strongest of them will be returned. Zero or negative values imply that no limit on the maximum is set and all detected corners are returned.

qualityLevel(3rd parameter):

Parameter characterizing the minimal accepted quality of image corners. The parameter value is multiplied by the best corner quality measure, which is the minimal eigenvalue

minDistance(4th parameter):

Determines minimum possible Euclidean distance between the returned corners. By increasing this parameter, sparser and less number of corners will be detected.

mask(5th parameter):

Determines which regions of the input image we are interested in detecting its corners.

blockSize(6th parameter):

Size of the block over which neighborhood of each pixel will be considered in order to compute derivatives of the algorithm. Decreasing this parameter may result in detection of more fine corners.

useHarrisDetector(7th parameter):

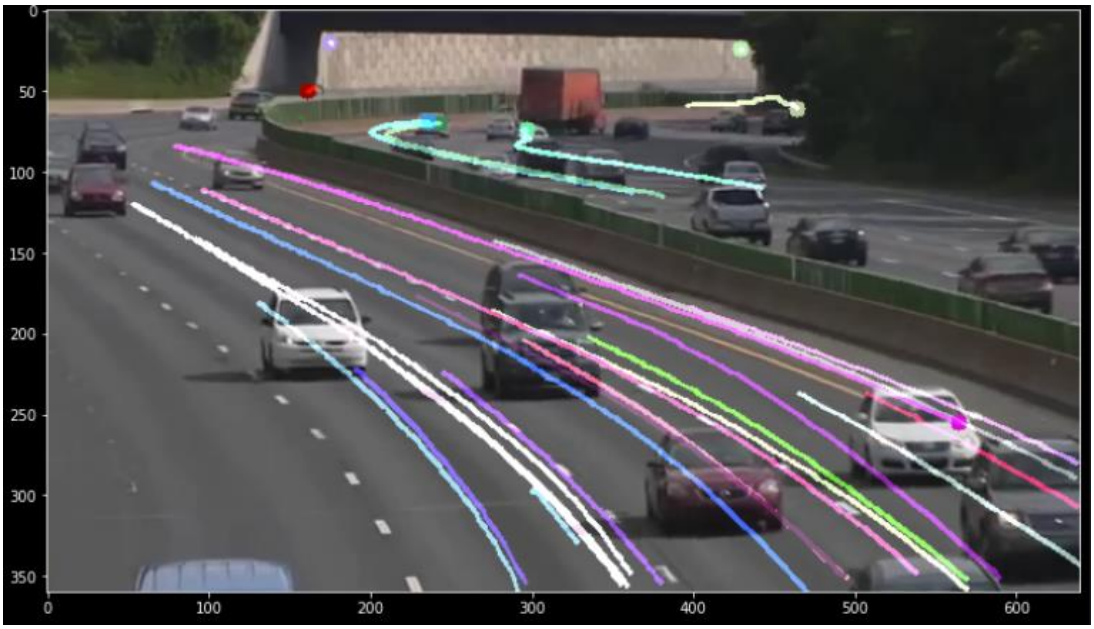
Parameter indicating whether to use a Harris detector or cornerMinEigenValue.

k(8th parameter):

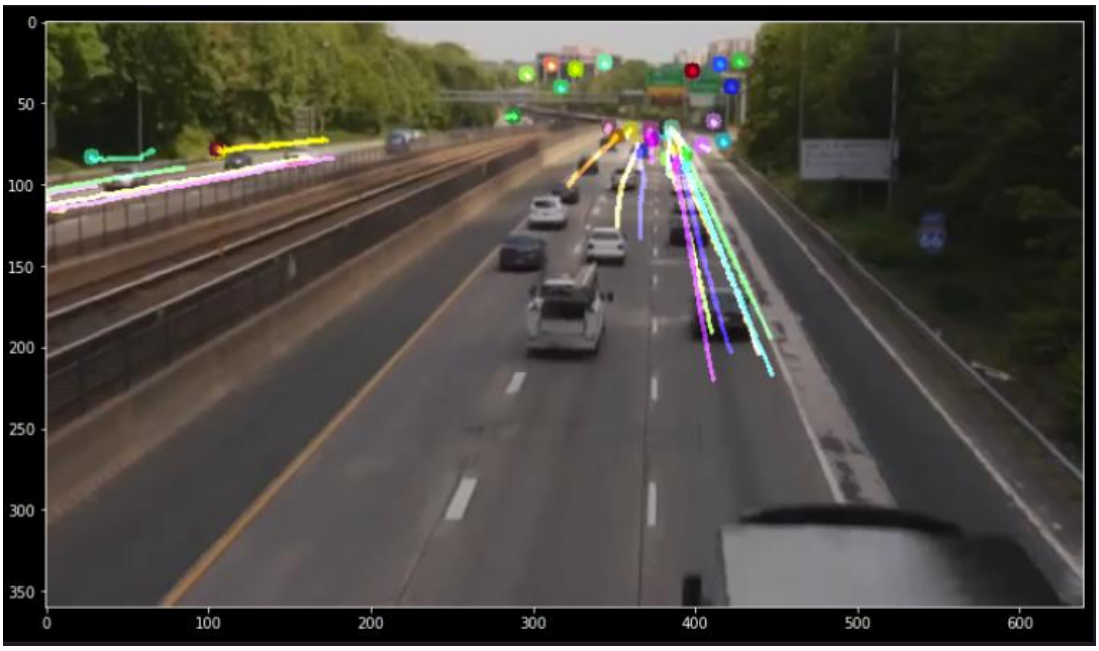
Free parameter of the Harris detector.

2.

Optical flow for the entire first part of the video:



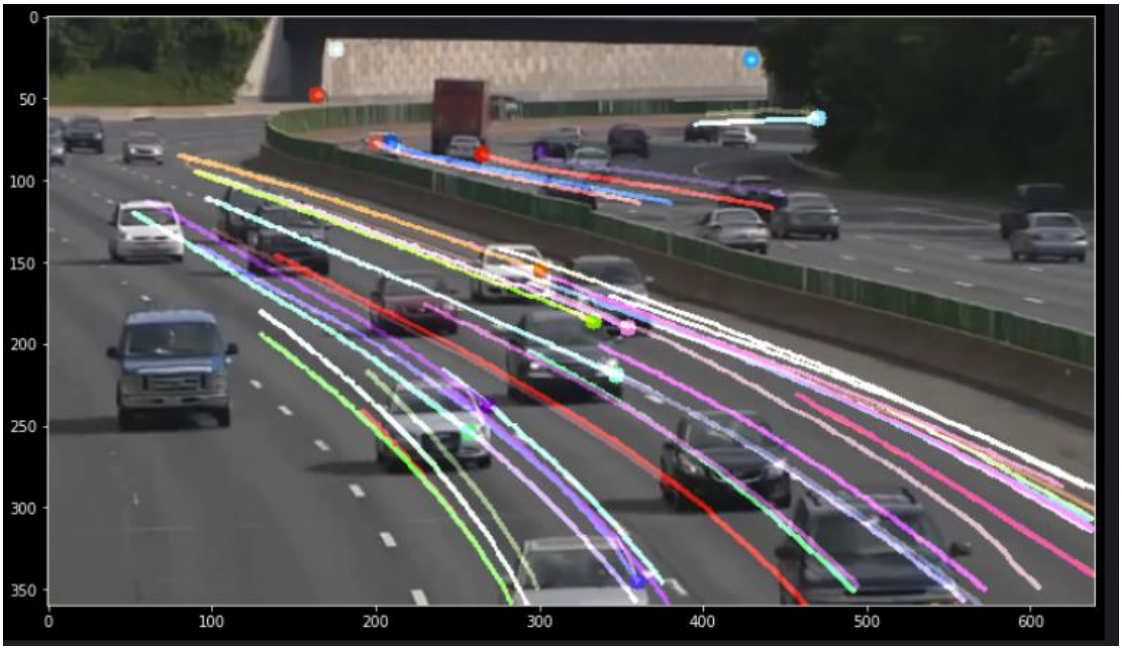
Optical flow for the entire second part of the video:



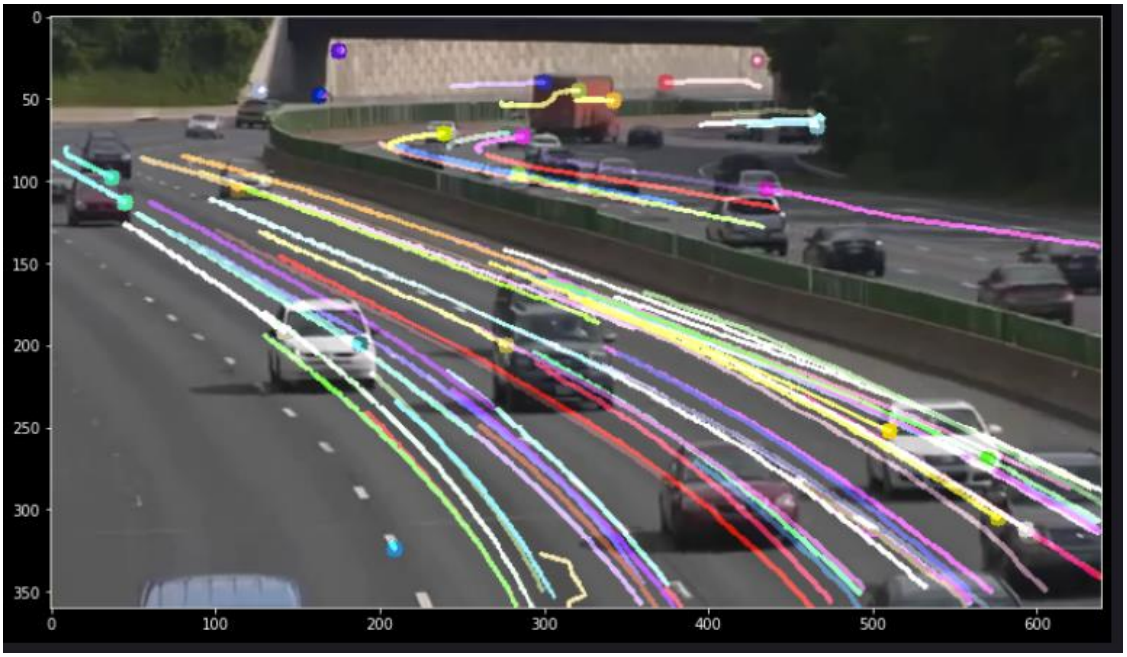
Then, by separating the first part of the video into two parts, and considering optical flow of newly added corner points after an

approximate five seconds, the following two images were obtained:

Optical flow after 5 seconds:



Final optical flow of the first part of the video:



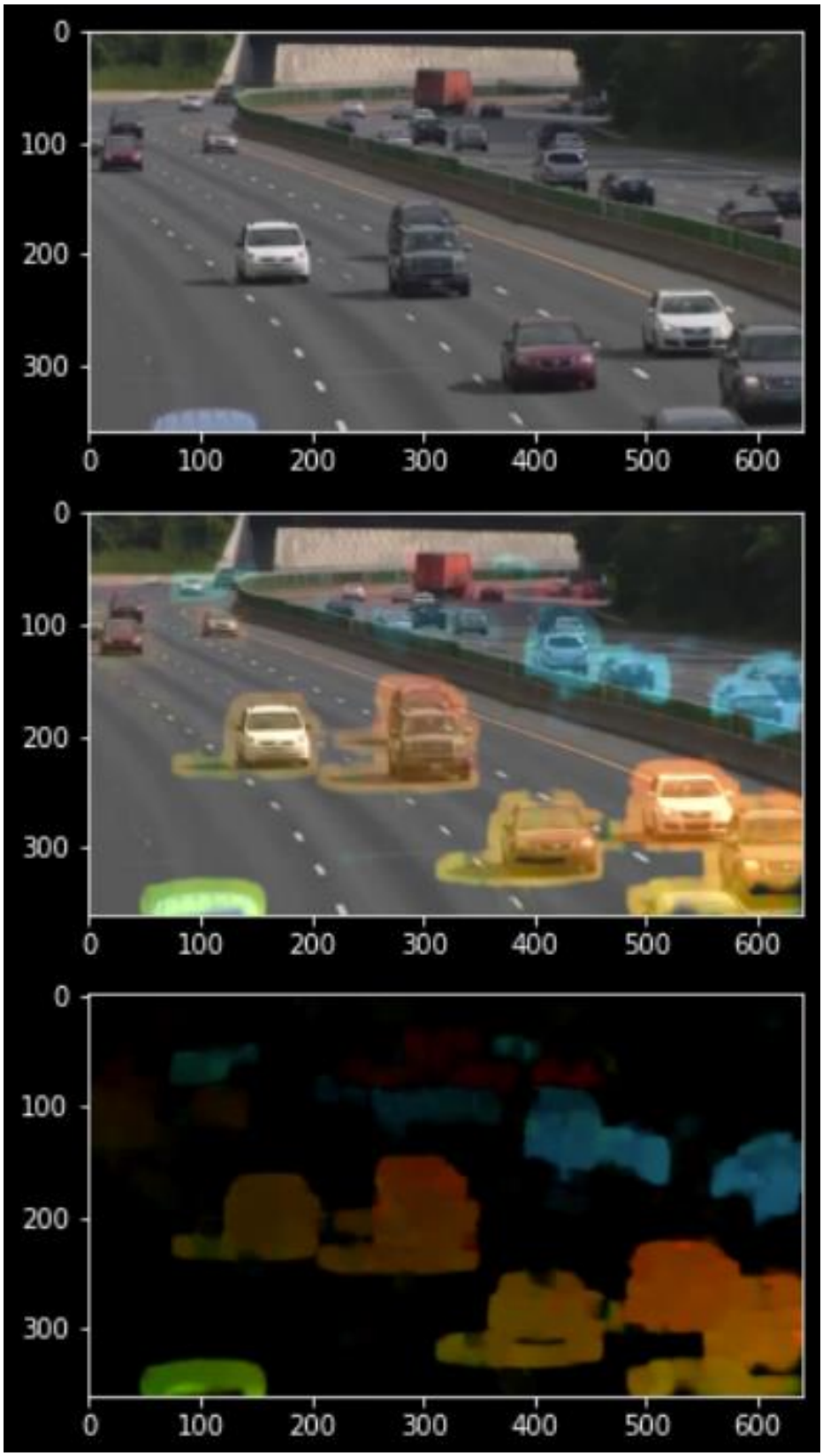
3.

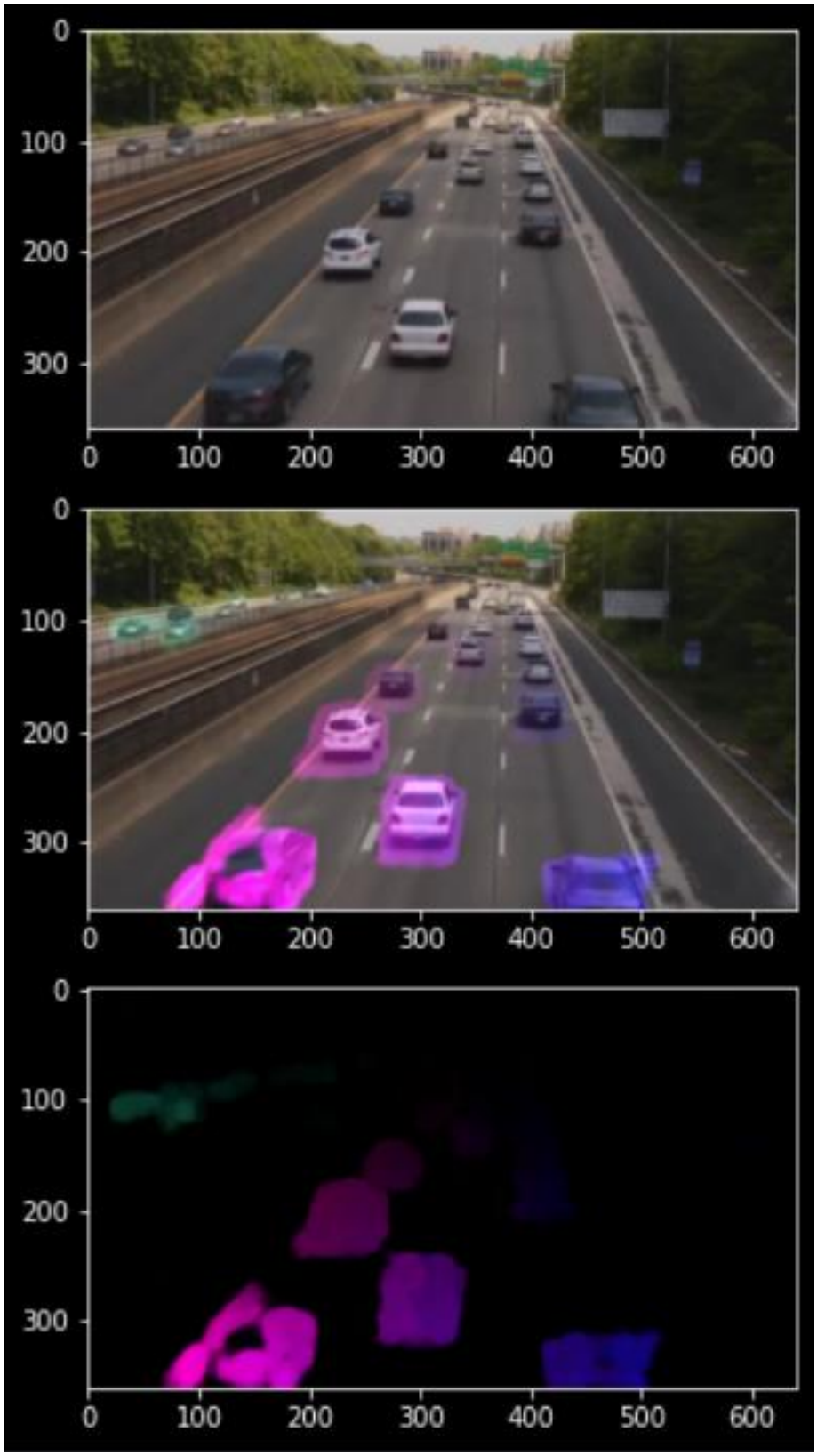
Generally, there are two main approaches for computing optical flow, first computing optical flow for all pixels or second computing optical flow for only detected corners. Lukas-kanade method comes under second approach. This method solves the basic optical flow equations by considering nearby pixels. Therefore, since this method is purely a local method, it cannot provide sensible flow information in the interior of uniform regions of the image; therefore, it only considers corners of the image.

4.

Gunner-Farneback is a dense optical flow method which considers all pixels and their intensity changes between two consecutive frames as against Lucas-Kanade method which was a sparse optical flow method which ignores many of the pixels due to its limitations. Although Gunner-Farneback is more time consuming algorithm, it has much higher accuracy compared to Lucas-Kanade.

Two examples of the mentioned method are as follows:


0
100
200
300
0
100
200
300
400
500
600


0
100
200
300
0
100
200
300
400
500
600