Computer Vision - Homework 4

Ruchi Manikrao Dhore - W1652116

Monday 13th February, 2023

1 Introduction

This report provides an overview of Lucas-Kanade optical flow estimation method with sparse and dense optical flow. Additionally, we explored different image segmentation algorithms to separate image into coherent objects. In clustering algorithms, we examined the K-means clustering algorithm. As well, we studied Means-shift algorithm which is a clustering-based segmentation algorithm.

2 Lucas-Kanade Optical Flow

2.1 Input

As an input, 5 videos of varied lengths were taken from an freely available open source platform. To examine the effects, videos with varying number of objects and movements were included. Some videos with single object were considered whereas on the other hand videos with multiple objects were taken. Figure 1 shows the original set of the video.

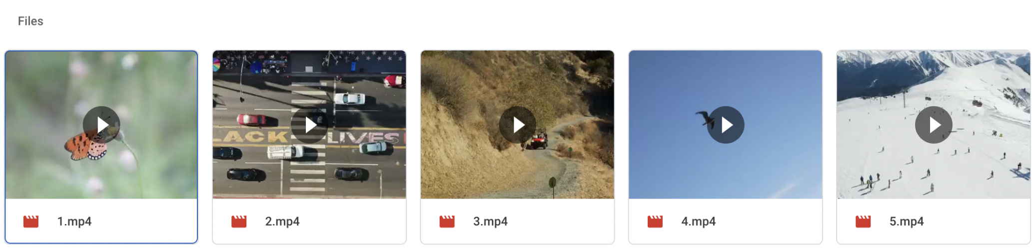

Figure 1: Video dataset

2.2 Optical Flow

The apparent movement of brightness patterns in an image as captured by a camera is known as optical flow. The calculation of optical flow involves analyzing the changes in intensity and location of pixels between frames, and using this information to estimate the velocity of the movement. Optical flow algorithms are useful for a variety of tasks in computer vision, such as tracking objects, estimating the speed of moving objects, and improving the accuracy of structure from motion algorithms. There are two types of Optical Flow, Sparse and Dense.

2.3 Lucas-Kanade Algorithm

The optical flow is frequently calculated using the Lucas-Kanade approach. It is a gradient-based method that uses the brightness constancy constraint to estimate the motion of pixels in an image sequence. The algorithm starts by assuming that the flow between two consecutive frames is constant within a small region around each pixel. It then calculates the derivatives of the intensity of the pixels in the current and previous frames and uses this information to estimate the flow vectors for each pixel.

2.4 Sparse Optical Flow

In Sparse Optical Flow, it computes the motion vector for the specific set of objects. Some preprocessing is essential to extract features from the image. We won't have the motion data for pixels in this optical flow. Following figures shows start, intermediate and final frames.

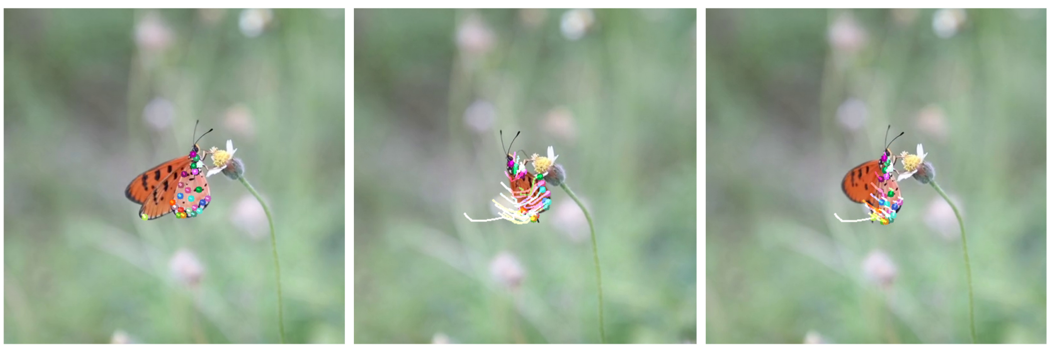

Figure 2: Output 1

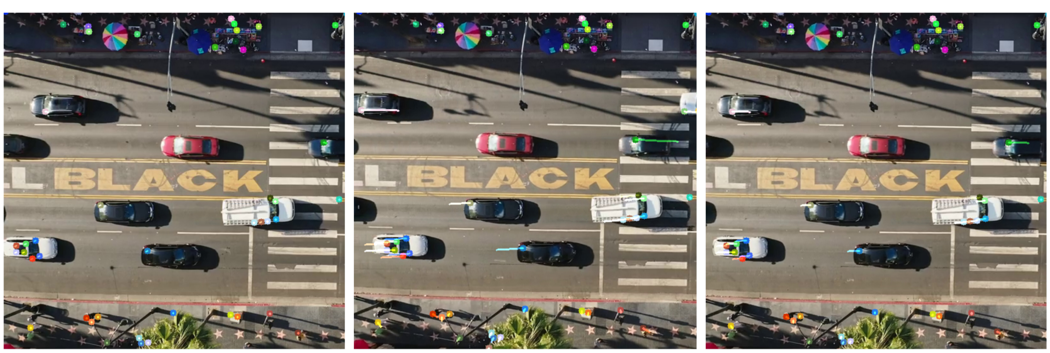

Figure 3: Output 2

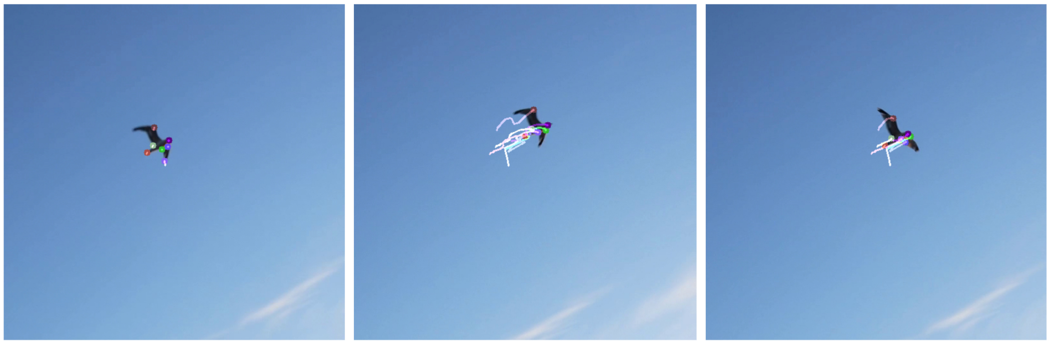

Figure 4: Output 3

2.5 Dense Optical Flow

The restriction of missing motion data for pixels is uplifted in Dense Optical Flow. It calculates a motion vector for every pixel in the image. Following figures shows start, intermediate and final frames.

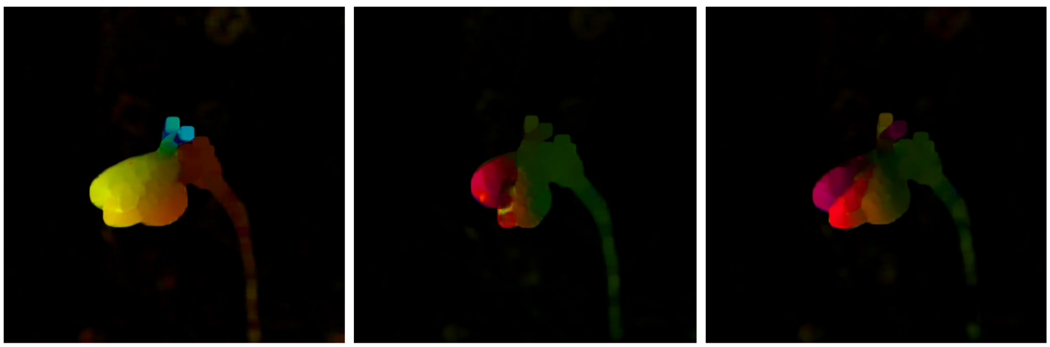

Figure 5: Output 1

Figure 6: Output 2

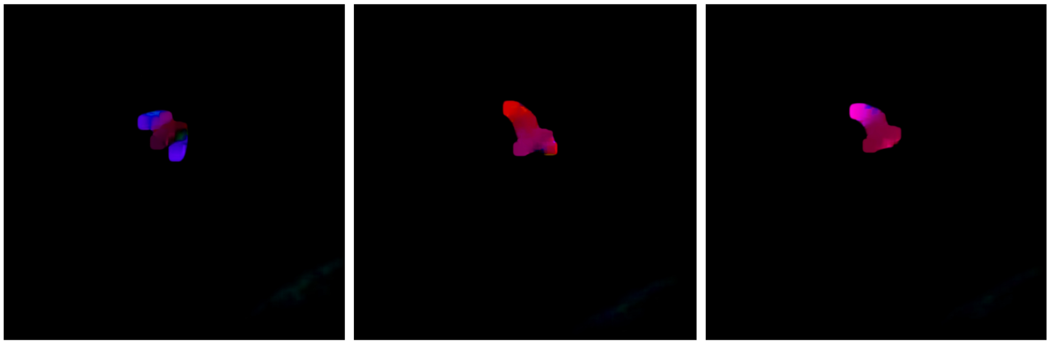

Figure 7: Output 3

2.6 Parameters and Observations

The OpenCV implementation of the Lucas-Kanade optical flow algorithm provides several parameters that can affect the performance and accuracy of the optical flow estimation. These parameters include:
1. Window size: The size of the window around each pixel that is used to calculate the optical flow. Increasing the window size will increase the robustness of the algorithm to noise and fast motions, but may also increase the computation time.
2. Pyramid scales: The number of pyramid levels to be used for optical flow calculation. Increasing the number of pyramid scales can increase the accuracy of the optical flow estimation, especially for large motions. However, this will also increase the computation time.
3. Max level: The maximum pyramid level to be used for optical flow calculation. Increasing the maximum level can increase the accuracy of the optical flow, but may also increase the computation time.
4. Criteria: The criteria for the termination of the optical flow calculation. The criteria can be set based on the maximum number of iterations or the maximum time taken to converge.
5. Flags: Various flags that can be set to control the behavior of the optical flow calculation, such as the use of a initial flow estimate, the use of a sparse optical flow, or the use of a dense optical flow.

In general, the choice of the parameters depends on the specific requirements of the application and the trade-off between accuracy, computation time, and robustness to noise and fast motions. For example, for real-time applications, it may be necessary to use a smaller window size or fewer pyramid scales to reduce the computation time. On the other hand, for applications where high accuracy is necessary, it may be necessary to use a larger window size or more pyramid scales to improve the robustness to noise and fast motions.

Sparse optical flow and dense optical flow are two approaches for estimating the motion of pixels in an image sequence.

Sparse optical flow focuses on tracking only a few selected keypoints in an image, such as corners or edges, and estimates the flow vectors for these points only. However, the sparse representation of the flow may not provide a complete picture of the motion in an image, especially for complex and fast motions.

Dense optical flow, on the other hand, estimates the flow vectors for all the pixels in an image. The dense representation provides a complete picture of the motion in an image.

However, the computation time for dense optical flow is much higher compared to sparse optical flow, and it may not be suitable for real-time applications.

3 Image Segmentation

3.1 Input

As an input, 10 images were taken from each of the 10 classes from the CIFAR-10 dataset. The CIFAR-10 provides tiny images of the dimensions 32x32 and image classes like airplane, automobile, bird, cat etc. Figure 2 shows the original set of the images.

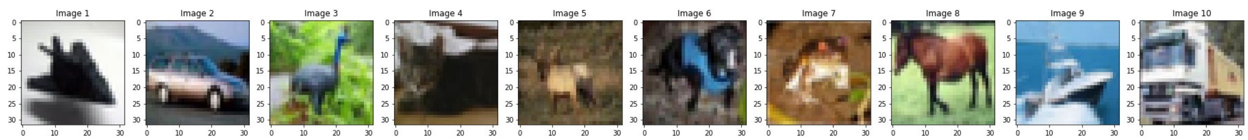

Figure 8: Image dataset

3.2 K-means Clustering

K-Means clustering is an algorithm that divides a set of data points into a fixed number of clusters, where each point belongs to the cluster with the closest mean. Following figure shows the output on applying K-means clustering:

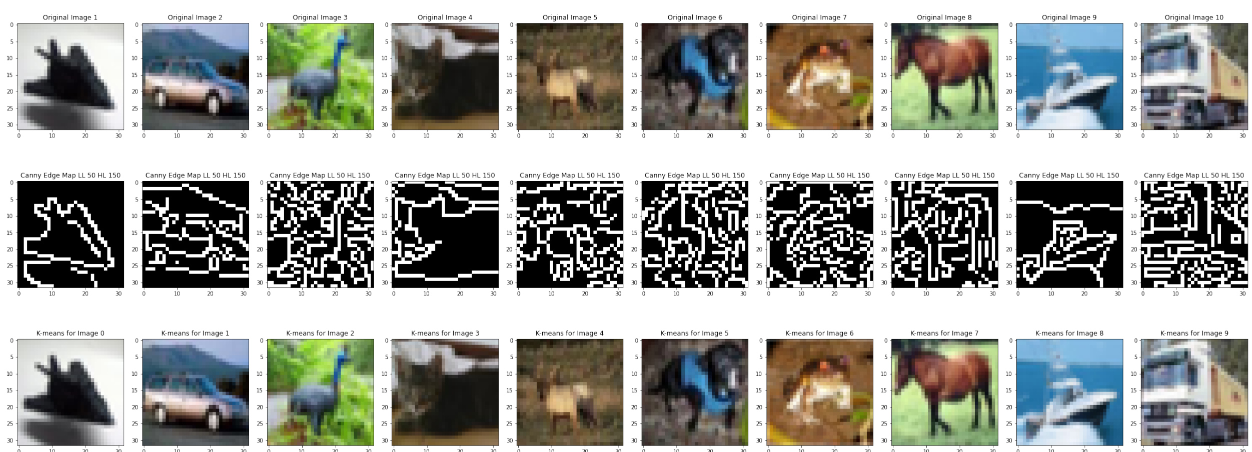

Figure 9: K-means clustering output

3.3 Means-shift Algorithm

Mean-Shift clustering is a non-parametric algorithm that does not require the number of clusters to be pre-defined. Mean-Shift iteratively moves a window over the data and shifts it towards the direction of the highest density. Following figure shows the output on applying Means-shift algorithm with different values of window radius and color window radius:

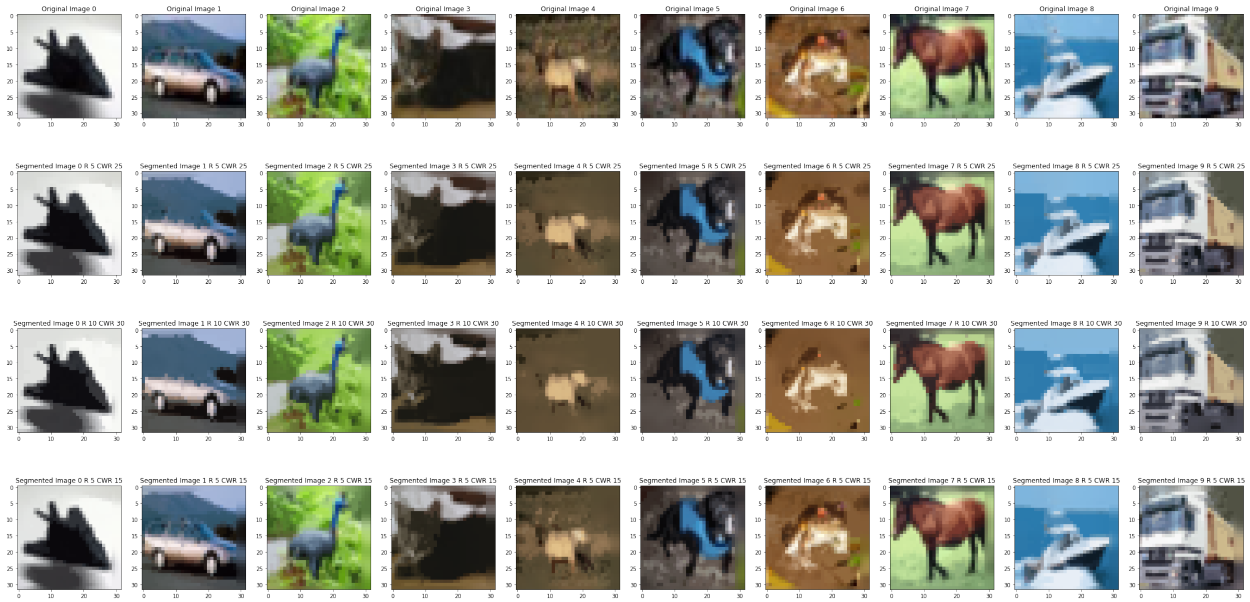

Figure 10: Means-shift algorithm output

3.4 Observations

The choice between K-Means and Mean-Shift clustering algorithms for images depends on the specific requirements of the image analysis task. Both algorithms are unsupervised clustering methods that can be used to group pixels with similar properties in an image.

K-Means is computationally efficient and can be used to cluster images with large datasets. It is useful for tasks such as image segmentation, texture analysis, and feature extraction. However, K-Means requires the number of clusters to be pre-defined, and may not perform well when the data is highly dimensional.

Mean-Shift clustering, on the other hand, is useful for tasks such as object tracking, feature tracking, and image segmentation. Mean-Shift is computationally more expensive than K-Means and may not be suitable for large datasets.

In conclusion, for tasks such as texture analysis, feature extraction, or clustering images with large datasets, K-Means is a good choice. For tasks such as object tracking, feature tracking, or image segmentation, where the number of clusters is not known in advance and where the data may not be highly dimensional, Mean-Shift can be a good choice.

4 Conclusion

With the help of this assignment, we explored two optical flows in Lucas Kanade, Sparese optical flow and Dense optical flow. We delved into different image segmentation algorithms. In K-means clustering, we applied Canny edge detection algorithm and then implied the algorithm to show segmentation. Similarly, in the Means-shift algorithm, we tried different set of values for window radius and color window radius to study effects.