**Limitiations of a specific counting pipeline**

The accuracy of the pedestrian detection algorithm may be affected by factors such as variations in lighting conditions, occlusions, and the position and orientation of the pedestrians in the frame.

The tracking algorithm may struggle to maintain accurate tracking of pedestrians if they move too quickly or if there are multiple pedestrians close together, resulting in potential errors in the count.

The field of view of the camera may be limited, meaning that pedestrians who are outside of the camera's field of view will not be counted.

The pipeline may be sensitive to changes in the environment, such as the addition or removal of objects in the scene, which could impact the accuracy of the pedestrian count.

**Discuss each step for the localisation of Scale Invariant Feature Transform (SIFT) interest points.**

Scale Invariant Feature Transform (SIFT) is a feature detection algorithm used to identify and extract unique, stable, and distinctive features in an image. These features, known as SIFT interest points, are used in tasks such as image matching, object recognition, and 3D reconstruction. Here are the steps involved in the localisation of SIFT interest points:

Scale-space extrema detection: The first step in SIFT is to build a scale-space representation of the image by applying a Difference of Gaussians (DoG) filter at multiple scales and orientations. This results in a set of scale-space extrema, which are the points in the image where the DoG response is a local maximum or minimum.

Keypoint localisation: The scale-space extrema are then refined to accurately localise the SIFT interest points. This involves fitting a 3D quadratic function to the scale-space extrema to determine the precise location and scale of the keypoint.

Orientation assignment: SIFT interest points are associated with a dominant orientation, which is used to provide invariance to image rotation. To determine the dominant orientation, SIFT computes the gradient orientation histogram of the image region around each keypoint. The dominant orientation is then assigned to the keypoint as the orientation with the highest peak in the histogram.

Keypoint descriptor computation: Once the keypoints have been localised and oriented, a descriptor is computed for each keypoint by sampling the gradient orientation and magnitude at a set of locations in the image region surrounding the keypoint. The descriptor is represented as a histogram of gradient orientations, which is used to describe the appearance of the keypoint and provide invariance to image transformations such as scaling and rotation.

Some potential limitations of the SIFT algorithm include:

**SIFT**[**[3]**](https://en.wikipedia.org/wiki/Scale-invariant_feature_transform#cite_note-patent-3)**can robustly identify objects even among clutter and under partial occlusion, because the SIFT feature descriptor is invariant to**[**uniform scaling**](https://en.wikipedia.org/wiki/Scaling_(geometry))**,**[**orientation**](https://en.wikipedia.org/wiki/Orientation_(geometry))**, illumination changes, and partially invariant to**[**affine distortion**](https://en.wikipedia.org/wiki/Affine_transformation)**.**[**[1]**](https://en.wikipedia.org/wiki/Scale-invariant_feature_transform#cite_note-Lowe1999-1)

**Sift is computationally expensive so not good on devices with low battery!**

Despite the fact that SIFT keypoints are robust against several factors (rotation, scaling etc.), they are not robust against strong variations in illumination.

SIFT is computationally intensive, which can make it slow to run on large datasets or in real-time applications.

SIFT is sensitive to image noise, which can affect the accuracy and stability of the detected features.

**Counting pedestrians pipeline**

A computer vision pipeline for counting the number of pedestrians crossing the street during a 30-minute interval could involve the following steps:

1. Image acquisition: The camera captures images at regular intervals, such as every second, and stores them in memory.
2. Image pre-processing: The images are pre-processed to improve the quality and reduce noise. This can include techniques such as histogram equalization, denoising, and resizing.
3. Background Subtraction: The pre-processed images are then subtracted with a static background image of the scene taken at different time or by using running average of few frames. This will give us the moving objects in the scene.
4. Blob Detection: The moving objects image obtained in the last step is then processed using Blob detection techniques such as Laplacian of Gaussian, Difference of Gaussian or by using simple thresholding techniques.
5. Object tracking: The detected blobs are then tracked across multiple frames to ensure that each individual pedestrian is only counted once. This can be done using techniques such as Kalman filtering or optical flow.
6. Counting: Finally, the number of pedestrians crossing the street is counted based on the number of unique individuals tracked by the object tracking step.

In terms of choices for this pipeline, background subtraction can be done using running average of few frames or by using a static background image, for blob detection Laplacian of Gaussian and Difference of Gaussian are commonly used, for object tracking Kalman filter or optical flow based algorithms have been traditionally used.

The pipeline's limitations could be:

* In case of heavy occlusion or shadows, the background subtraction may not be able to detect the pedestrian accurately.
* In case of a large crowd, it may be difficult to track individual pedestrians, leading to an overcount or undercount.
* The pipeline is not robust to changes in lighting conditions, weather, or camera angle, which could significantly impact the performance of the background subtraction and blob detection algorithms.
* The pipeline may not be able to account for pedestrians who are obscured by other objects or who are outside of the field of view of the camera.
* One of the major limitation would be the size of the dataset used for training, if the dataset doesn't include images with a similar scenario as the one we are trying to detect, the pipeline may not perform well.
* With this pipeline, it's not possible to detect the direction of pedestrian.

Overall, while this pipeline can provide an estimate of the number of pedestrians crossing the street, it is important to note that it may not be completely accurate and further improvements could be made to make the pipeline more robust to changes in lighting conditions, weather, and camera angles.

**Invariance and repeatability of interest points**

Invariance refers to the property of a feature or algorithm to remain unchanged or have consistent output when presented with transformed input. In the context of computer vision, this often refers to the ability of local features or interest points to remain consistent and identifiable despite changes in viewpoint, lighting conditions, or image translation. Repeatability, on the other hand, refers to the ability of a feature or algorithm to consistently detect and identify the same feature or interest point across multiple images of the same scene. Together, these properties are important for many computer vision tasks, such as object recognition, image alignment, and structure from motion.

**Repeatability:** given two images of the same object or scene, taken under different viewing conditions, a high percentage of the features detected on the scene part visible in both images should be found in both images