Detailed Approach and Analysis

A. Image Segmentation

The first step involves segmenting each image in the stack to detect objects across image.

- **Input**: Read a multi-frame TIFF file representing a sequence of images (stack) as given in the assignment.
- **Converting to Greyscale**: Each frame is converted to grayscale to reduce computational complexity.
- **Thresholding**: Otsu's method is used for adaptive thresholding, which helps separate objects from the background.
- **Morphological Filtering**: To improve segmentation quality, we remove small noise using morphological operations.
- **Labeling Segmented Objects**: After thresholding, identify individual objects within each frame, assigning them unique labels.

B. Object Tracking Across Frames

Linking objects across frames, is essential for understanding their movement and transformation over a timeframe.

- **Object Properties**: First we extract properties such as centroid position and area/size for each segmented object.
- **Using Tracking Algorithm**: We implemented a nearest-neighbor approach, where each object in the current frame is matched with the closest object in the previous frame.
    - **Distance Based**: Currently we have used an Distance based approach. maximum allowable movement distance between frames which had been defined in `constants.py`
    - **Size Based**: For better accuracy we can use size based combined with distance based threshold to ensures that objects with considerable size variations aren't mistakenly matched.
- **Assigning Object IDs**: If a new object is detected within the distance threshold, it is assigned an existing ID. If not, a new ID is created, which is further used for the appearance of new objects in later frames.

C. Exporting Tracking Data

The final step involves exporting the collected tracking data as a CSV file.

- **Data Structure**: We create a following columns for analysis,
    - `Object Index`: Unique index for each object.
    - `Object ID`: The ID assigned to each object for consistent tracking across frames.
    - `Position`: Centroid coordinates (x,y) for each object in each frame.
    - `Size`: The area of each object, useful for understanding transformations.

- **Frame No.:** The frame number, allowing tracking of objects across time.
- **Object Index**: This is a unique identifier for each entry in the tracking data
- **Object ID**: This ID is consistent for each object across frames, allowing us to track the movement and transformations of specific objects over time.

Analysis:

From the labelled images generated in the output_images folder we can see that in totall 5 unique objects have been detected in all the frames. 1. Small Arrow at the top of frame (orange color) 2. Smiley circle which covers the all the above object (purple color) 3. Left eye circle (yellow color) 4. Right eye circle (pink color) 5. Smile rectangle (green colour)

And After analysis the tracking.csv file which represents the position and size of five distinct objects detected and tracked across three frames in an image stack. Each row in the CSV file captures an object's unique index, ID, position, size, and the frame number in which it was observed.

1. Position Analysis (Centroid Coordinates)

- **Position Format**: Each object's position is recorded as a centroid with `(y, x)` coordinates. This is helpful for tracking the object's movement within the image space.
- **Object Movement**:
  - **Object ID 1** shows a steady movement from `(26.17, 25.90)` in Frame 0 to `(41.17, 40.90)` by Frame 2, indicating it moved roughly diagonally across frames.
  - **Object ID 2** moved along the x-axis from `255.60` to `270.60` across frames while remaining relatively stable in the y-axis.
  - **Object ID 5** shows a significant shift from `(325.59, 257.54)` in Frame 0 to `(361.04, 272.28)` in Frame 2, suggesting a consistent movement which we can verify from the stack images as well,

These coordinate changes can indicate the direction and speed of movement for each object, which is essential for applications like motion tracking or behavioral analysis.

2. Size Analysis

- **Object Size**: Each object has a `Size` attribute measured in pixel area, which is consistent across frames for most objects but shows variations in some cases.

- **Consistency Across Frames**:

- **Object ID 1** maintains a size of 743 pixels across all frames, suggesting it's likely static in terms of shape and size.
- **Object ID 5**, however, changes from 4638 in Frame 0 to 3449 in Frame 1, and then to 4627 in Frame 2.

Size changes can indicate physical transformations or interactions with other objects or background elements.

Object ID	Initial Position	Final Position	Movement Observations	Size Stability
1	(26.17, 25.90)	(41.17, 40.90)	Diagonal movement, steady shift	Stable
2	(255.60, 255.80)	(270.60, 270.80)	Diagonal movement, steady shift	Stable
3	(183.06, 192.15)	(198.05, 207.15)	Gradual movement	Stable
4	(183.10, 322.30)	(198.10, 337.30)	Gradual movement	Stable
5	(325.59, 257.54)	(361.04, 272.28)	Significant movement, x and y-axis	Variable across frames