

## MP7 - Template-Matching Based Target Tracking

Logan Boswell

### Overview

This MP is focused on implementing template-matching based target tracking. In order to implement this technique, some python functions were written to select a target area as a template and perform tracking on the area.

### Template-Matching Based Target Tracking

The first step in my template-matching based object tracking function is to load and sort all image frames from the input folder. I then manually select a bounding box in the first frame using OpenCV's ROI selector, which represents the initial location of the target (the girl's head). This region is converted to grayscale and used as the initial template for tracking. For each subsequent frame, I convert the frame to grayscale and perform a local exhaustive search within a specified radius around the previous target location. For every candidate patch in the search window, I compute a similarity score against the template using one of three metrics: Sum of Squared Differences (SSD), Cross-Correlation (CC), or Normalized Cross-Correlation (NCC). The location with the best score is selected as the new target position. A bounding box is drawn at that location and saved as an output image. Optionally, each frame with the drawn box is also written to an MP4 video file. The result is a list of bounding box positions across all frames, which represents the tracked path of the target.

### Results

I tested the tracking function with the sum of squared difference, cross-correlation, and normalized cross-correlation matching methods and obtained the videos (shown as gifs below and included in submission).



Figure 1: Test With Cross-Correlation Matching (My Worst Results)



Figure 2: Test With Normalized Cross-Correlation Matching



Figure 3: Test With Sum of Squared Difference Matching