Assignment-2
Template Tracking

Deadline : **October 20, 2021**

Video tracking is the process of locating a moving object (or multiple objects) over time using a moving camera. This assignment will use traditional non-learning-based object tracking techniques to track specific objects in the given test sequences. Such a system needs no prior training and even may not have seen an object prior to tracking.

The dataset consists of different test video sequences. Each video sequence consists of-
1. Framewise images.
2. Ground truth file that consists of bounding boxes per frame of the format (x, y, h, w) where (x,y) are the coordinates of the leftmost corner of the object and (h,w) are the height and width of the object.

You can consider the ground truth for the first frame as the object template for tracking the object in future frames.

Note - There may be multiple objects present in the video that are visually similar to the object of interest (that we are tracking). The tracking algorithm that you would develop should not get distracted by such objects.

Assuming a sequence consists of N frames, evaluation of tracking will be done based on the mean IOU across all the remaining N-1 frames.

$$mIOU \ = \ \frac{1}{N-1} \frac{Area\ of\ overlap\ of\ prediction\ with\ the\ bounding\ box}{Area\ of\ union\ of\ prediction\ and\ bounding\ box}$$

You need to report mIOU quantitatively and qualitatively report examples as asked in each part for all the parts below.

Part 1 Appearance-based Techniques [1 point]
    a. Implement template tracking techniques using block-matching techniques (appearance based) sum of squared distance (SSD) and normalised cross correlation (NCC) (refer to the relevant course lecture).
    b. Analyze problems associated with appearance-based matching techniques.

c. Show failure examples using a few images for the best performing appearance-based method and explain it.

Part 2 Lucas Kanade Algorithm [2 points]
a. Implement the Lucas Kande algorithm and test it on the provided datasets.
b. Compare your results visually with part 1.
c. See if you were able to resolve a few of the failure cases of part 1.

You can use the translation implementation from opencv but you should additionally implement the affine and projective transformation model of your own.

Part 3 Pyramid based Lucas Kanade Algorithm [3 points]
a. Implement pyramid-based Lucas Kanade algorithm using a multiscale Gaussian pyramid. OpenCV contains routines to generate an image pyramid which you can use.
b. Experimentally find the optimal number of pyramid levels, and other hyper-parameters to improve part 2.
c. Compare results visually to part 2.

Part 4 Live Demo [3 points]
a. Create a framework in which you can live track an object on uploading the template image of the object to be tracked.
b. Use your best performing model.
c. During the demo we will ask you to upload the image of an object and track it live using a bounding box.
d. The model should be able to handle any affine and projective transformation to the object like rotating it via a certain angle. The shape of the bounding box, called rotated-box parameterization in literature, should change as per the transformation.

Report will carry an additional 1 point.