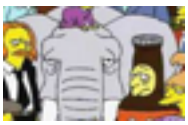
Comments 1: Particle Filter

Question) Discuss what your state vector is and why you define this state vector.
Answer)
The state vector, `particles`, in the modified code represents the state information of each particle in the particle filter. It is a 4-dimensional vector that includes the X and Y positions, as well as the X and Y velocities of the particles.
The X and Y positions are randomly generated integers between 0 and `imgW-1` (inclusive) and 0 and `imgH-1` (inclusive) respectively. These positions indicate the coordinates of the particles on the image plane.
The X velocity is obtained by sampling from a normal distribution with a mean of 3 and standard deviation of 3, while the Y velocity is obtained from a normal distribution with a mean of 0 and standard deviation of 3.
By using this state vector, the algorithm captures the relevant information about the particles' positions and velocities, enabling the prediction and update steps of the tracking Algorithm.

Question)
Do you assign the prediction noise? If so, what your prediction noise is and how do you define it?
Answer)
Prediction noise is assigned in the provided code to account for uncertainties in the system. The prediction noise is incorporated for both position and velocity components of the particles.
For position prediction noise, a 2D Gaussian distribution is used with a mean of 0 and a standard deviation specified by the variable posNoise. The gaussian2D function generates random values from this distribution.
For velocity prediction noise, a 1D Gaussian distribution is used with a mean of 0 and a standard deviation specified by the variable velNoise. The gaussian1D function generates random values from this distribution.
In the code, the prediction noise is added to the predicted position and velocity components of each particle using the np.dot function and element-wise addition.
Therefore, yes, prediction noise is assigned in the provided code, and it is defined by the values of posNoise and velNoise.

Comments 2: NCC Template Matching

Closest Match	Resultant Image (Image Enlarged)
1st	
2nd	
5th	
10th	
100th	
500th	

Discussion : The 1st,2nd ,5th and 10th closest match have shown almost similar patterns. The object presented in the template have similar match in the search image as well. The resultant image achieved from the 100th match has shifted towards left and the object in the right is slightly cropped but still the elephant is captured perfectly. The 500th match has shown slightly different result and patterns but the color, texture remained the same in all results. The resultant image has cropped the top most object.
From all the results we can assess that the top level matched templates have higher level of similarity to the template compared to lower-ranked templates. The first 4 results seems satisfactory and it shows the efficacy of the implemented matching approach. While from the last results (500th match), it indicates area for improvement and additional refinements in the matching algorithms.