For my results, please open ***main.ipynb*** via Jupyter Notebook. You will see everything neatly laid out. There are 6 functions:

- **exhaustive(image, window)** takes in an image and a window size, and returns all possible windows within the image, allowing for exhaustive search for later detecting the best match of the template region.
- **SSD(image, template)** calculates the sum of squared difference between a window generated by Exhaustive() and our initialized template.
- **CC(image, template)** calculates the cross-correlation between a window generated by Exhaustive() and our initialized template.
- **NCC(image, template)** calculates the normalized cross-correlation between a window generated by Exhaustive() and our initialized template.
- **return_imgs(num, fn, coordinates, box_size)** saves the best template-matched image from the exhaustive results based on the method (or fn) of choice. You can provide a set of coordinates to start your template selection with a window size, or not for the purposes of this assignment.
- **vid(fn)** transforms the images into a video. Only this function used cv2 to append the images together into a video, and nothing else.

All images were saved in the **'RESULTS' folder** in subfolders titled '…_imgs' based on their matching methods, and I stitched the **final videos** together in the folder 'RESULTS/VIDEOS', with the titles 'SSD.mp4' for matching via evaluating the sum of squared difference, 'CC.mp4' for matching via cross-correlation, and 'NCC.mp4' for matching via normalized cross-correlation.

For our last assignment, we studied various methods to track a target across a series of frames provided one template of our choosing. This best-match feature required us to (1) initialize by selecting an object within an image, in this case a template window of the face of a girl, then (2) search and match the best possible same-size window within the next frame to our template based on (3) three matching methods. Further, I went back and (4) made changes to the program so that if there was any occlusion (such as a person coming in the way of the girl), the girl would continue to be tracked despite the partial occlusion. Finally, (5) I converted the images per matching method into videos located in the RESULTS/VIDEOS folder.

To begin we must identify a target. If we're identifying her face, we can build a facial recognition tool, but to keep it simple we were allowed to create a bounding region and use that region as our template. I needed to exhaustively search every region within the next image by the dimensions of our smaller template, and so I searched for each possible in the exhaustive() function and created a dictionary to store them in. Then I created 3 functions as methods used to match one image to another. The idea was to use an equation, such as the sum of squared difference (SSD), and calculate the sum of each exhaustively searched region minus the template, squared, and produce a value. The value with the least difference should in theory be our best and closest match. In the first image, I found one region with a difference of 0, which simply indicated that the template came from this image, and so we had an exact match. Another way to describe the SSD is that is the squared Euclidean distance. The same was done with a cross-correlation (CC) method, which measures the similarity between two images as a function of displacement from one to another. If we look at their equations,

$$SSD: d(u,v) = \sum_{x,y} (f(x,y) - t(x-u, y-v))^2$$

and

$$CC: c(u,v) = f(x,y) * t(x-u, y-v)$$

The pdf shows the same equations in a different format. Unfortunately, our cross-correlation method produced terrible results, but when normalized, we rid ourselves of the problem of having major variations due to brightness of the image and template due to lighting and exposure. In fact, I'm sure no other student would have realized that the instructions didn't ask us to use a Normalized Cross-Correlation (NCC) method, it ACTUALLY asked us to use a Zero-Normalized Cross-Correlation method, because a regular normalized does not subtract the local mean value of intensities. I should get extra credit alone for this knowledge. Just kidding.

$$NCC: N(u,v) = \frac{1}{n} * \sum_{x,y} \frac{1}{\sigma_f \sigma_t} (f(x,y) - \mu_f) - (t(x,y) - \mu_t)$$

Moving on, we exhaustively search each window with each of the 3 methods, and find the least difference calculated from SSD and CC, and greatest value calculated from NCC, and we have our best match! I create a bounding box per image, stitch, and we have our video.

I did the optional extra credit in this assignment as part of the functions. I used different methods in searching for the best match. Once I was constantly updating the template with every frame such that it was more similar to the frame and accounted for changes in dimension, however it failed dramatically when the girl turned around. It did however do a good job of keeping a focus on the girl despite occlusion, but only in those frames. So I tried another method which was to average distance between the original 1st template and the updated template as long as the update template was to some degree similar, and this produced much better results. Finally a 3rd idea was to get rid of thresholds. Originally, based on my 3 methods, if I found a value that did not meet a certain threshold of accuracy, I would get rid of it and thus there were less false positives, but occlusion occurred. This is why I instead chose the best possible match based on my algorithms, which increased the number of false positives, lowered precision, but solved the occlusion problem. As you can see from my videos, occlusion is not a problem.

As for comparing the 3 methods, I found that SSD provided great average results as long as the face was facing the screen, but performed terribly otherwise and had some noise. CC seemed to be much more inaccurate and problematic. Fortunately, NCC was great in that like SSD, it accurately tracked the face when the girl was facing the screen (with less noisiness too), but also to some degree focused on the girl's head region with better matches than SSD especially when the girl had turned around. The reasoning for why, I've discussed at the top of this page. I highly recommend (Zero-)Normalized Cross-Correlation methods.

Overall this was a fun assignment! I attached a comparison below. The rest you can see via the videos.

SSD Result (frame 430):

CC Result (frame 430):

NCC Result (frame 430):