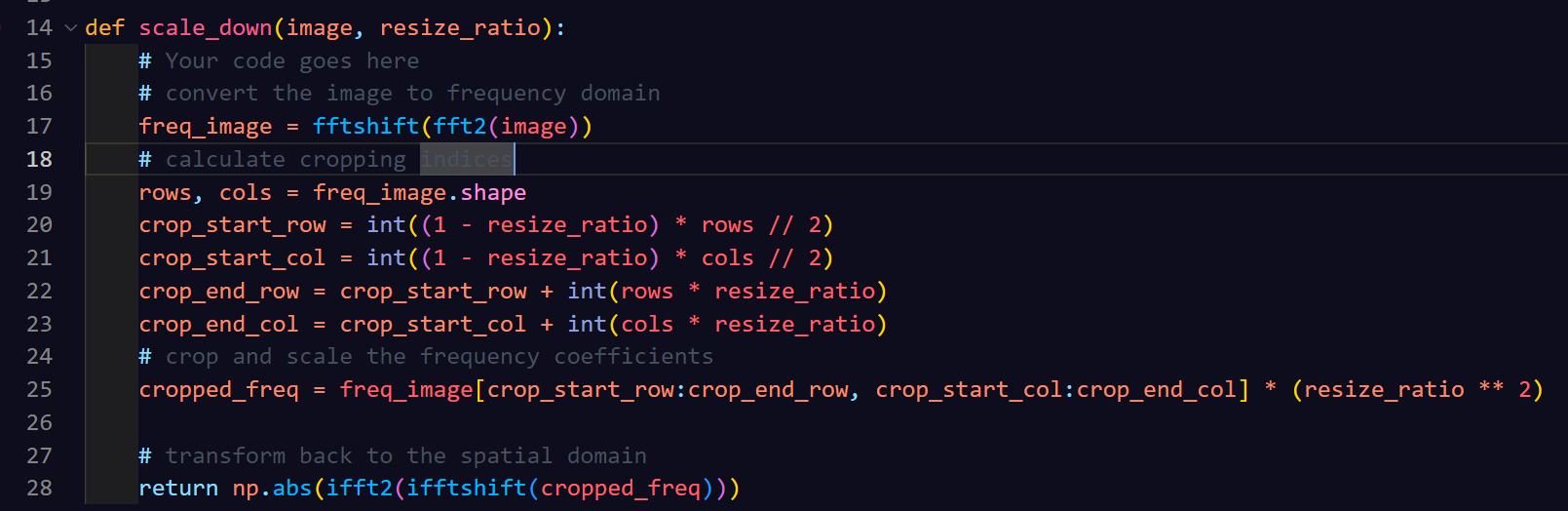
**Names:**

**Raneem Ibraheem (212920896),**

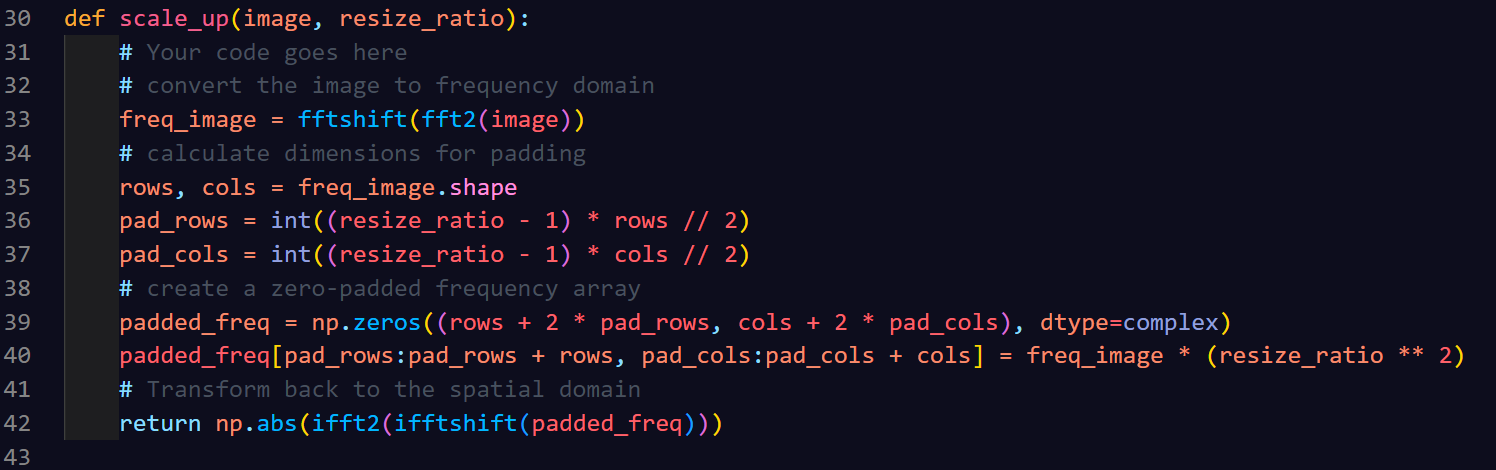
**Selan Abu Saleh (212111439)**

**Problem 1 – Template matching**

1. **First of all, we were asked to implement the scale\_down function that takes an image and a resize\_ratio and scales the image down by that ratio, hence we started by converting the image to the frequency representation, because when we want to scale down the image, the low frequencies that preserve the important information of the picture will be in the center, so we want to group them in the center and then start cropping the image according to the ratio and keep the middle part that we cropped. After preserving the middle part, we use the inverse transformation to go back to the spatial domain and get the scaled down image.**

****

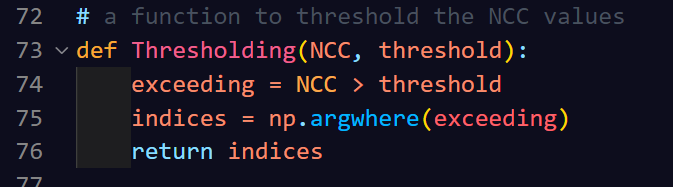
1. **As for the scale\_up image, a similar approach to the scale\_down function is taken, but instead we create an image that is all zeros, and then we put our image that is in the frequency domain in the middle of the zeros image, so what happens is that we get the frequency representation of our original image padded with zeros on all the sides, of course all of these happen in relation to the ratio size.  
   Then we do the inverse transform to go back to the spatial domain and get our scaled up image.**

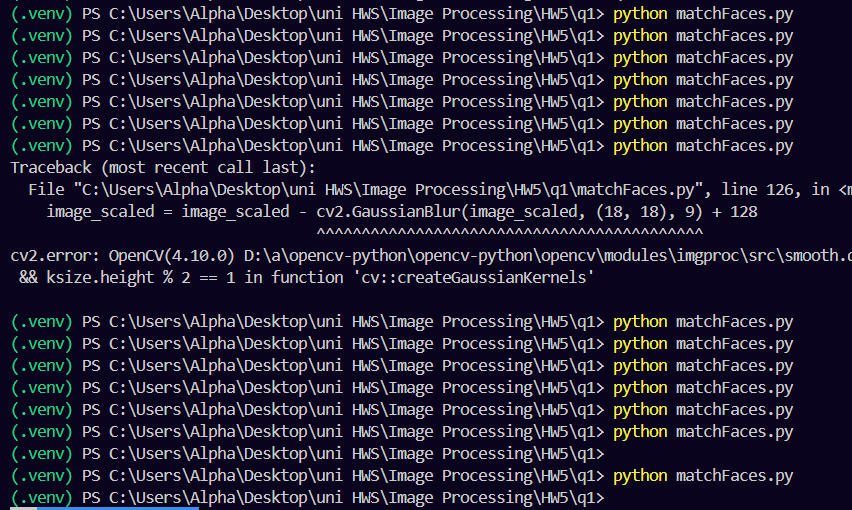
****

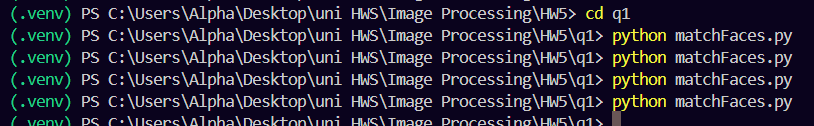
1. **In this section we were requested to implement the NCC function which takes the image and the pattern, and then slide the pattern across every possible position. We calculate the mean and the standard deviation for each overlapping window and normalize both to ensure the scale is similar. Then we use the formula of the NCC value at each position to get the 2d array of the NCC result where each value is the similarity between the pattern and the position in the image.**

****

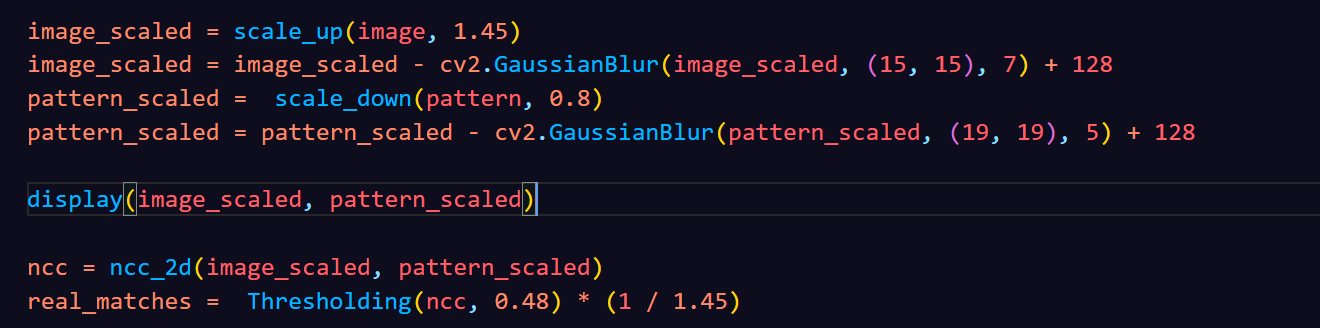
1. **Technically in here we only had to implement the Thresholding function which takes an NCC value and a threshold and filters out the values that don’t exceed the threshold.**

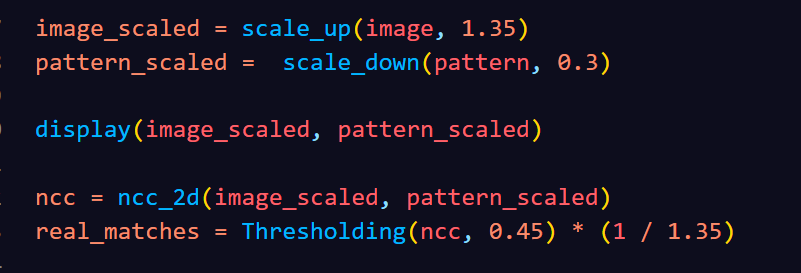
****

1. **For this section we had to scale up the images, and scale down the filters to an appropriate size in order to find the matching pattern, and we had use a gaussian blur for both the first image and the pattern to reduce the noise and smooth the high frequencies, and this helps to get a more accurate pattern matching. As for the second image, blurring wasn’t necessary because the facial features were more defined and blurring in this case might damage those features and reduce the accuracy.  
   The parameters choice in this section was through a trial-and-error process to find the best befitting parameters:  
   **

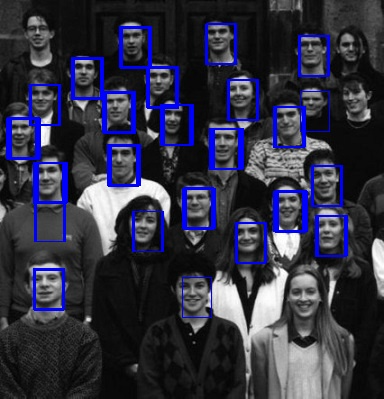
****

**Note: more trials have been conducted and there could have been better parameters to choose, but the current parameters are performing well in this case.**

****

****

**Results for the last section:**

****