CIS581 Project 4 Checkpoint
Richard Arietta & Danny Rerucha

- **Who you are working with (or if you are working solo)**

  For this project, Danny Rerucha and I (Richard Arietta) are working as a team.

- **Which option have you selected?**

  We have selected to implement Option 3. As of right now, we are utilizing MATLAB's built-in vision toolbox. The toolbox includes an object class known as the CascadeObjectDetector. This feature detector implements the Viola-Jones algorithm and is pre-trained for several basic and relevant facial feature classification models, such as 'Face', 'Nose', 'LeftEye', 'RightEye', and 'Mouth'. The results so far are highly variable, with some instances returning perfectly and some instances giving us many false negatives and missed positives. We are hoping to get some more powerful and reliable performance out of this class, otherwise we will have to consider switching to a different third party package.

- **Overview of you approach**

  Currently, the approach I have implemented functions as follows: The algorithm detects a face in the reference image (a photo of my own face), and within that face detects the center points of each major feature. The algorithm then takes an input "destination" image to which my face will be mapped, and detects the same features. For each image, the corresponding features are mapped from my face to the destination and a projective homography is built using the same approach as in Project 3 (panoramic image stitching).

  We are not yet sure how to handle the convex hull culling and blending of the reference image onto the destination image, but when the facial feature detector class functions at a high level this homography seems to provide a pretty good mapping onto the input (see example below).

- **Any preliminary results**

  Here is a not-yet-blended result image of my face mapped onto the first image in the easy input set.