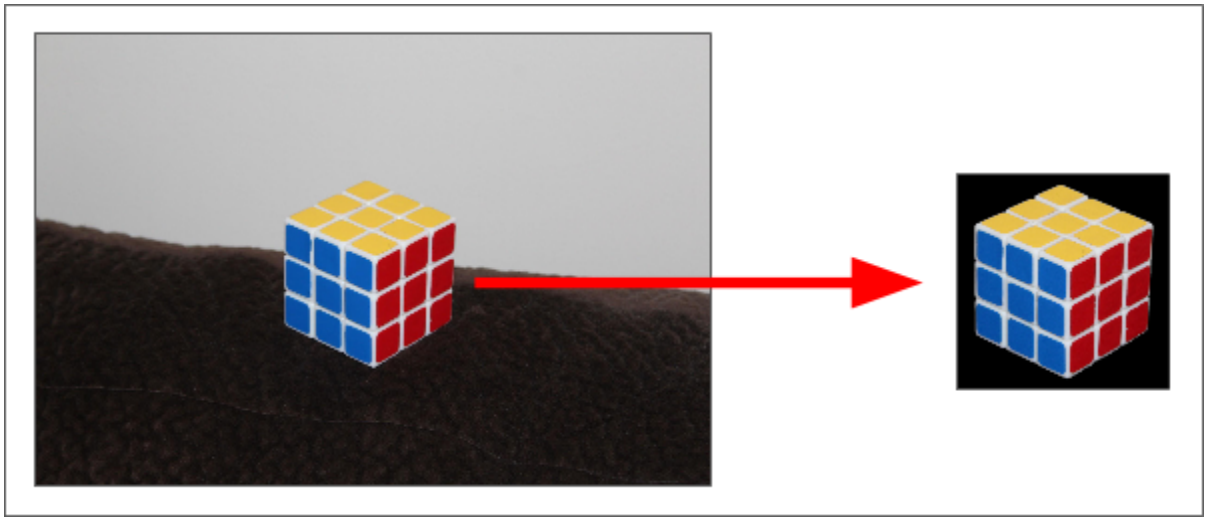
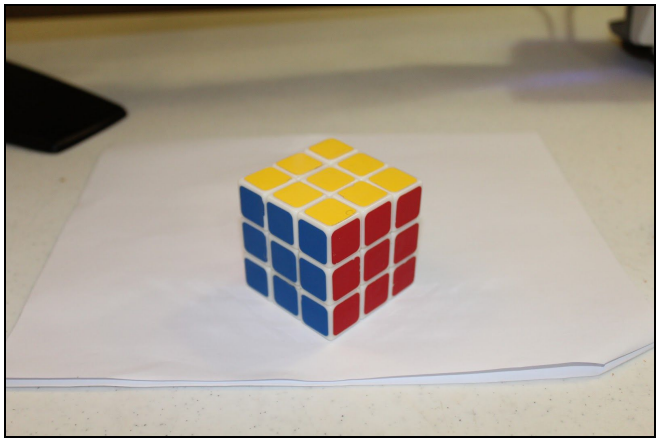
Jones, Jonathan

Homework 5

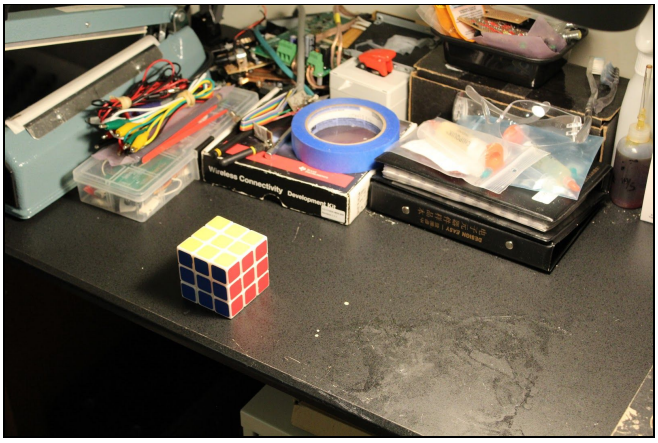
1) Input Images



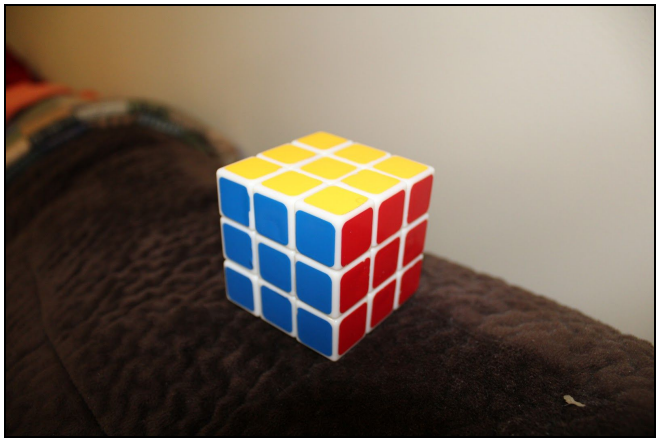
The cropped template image.



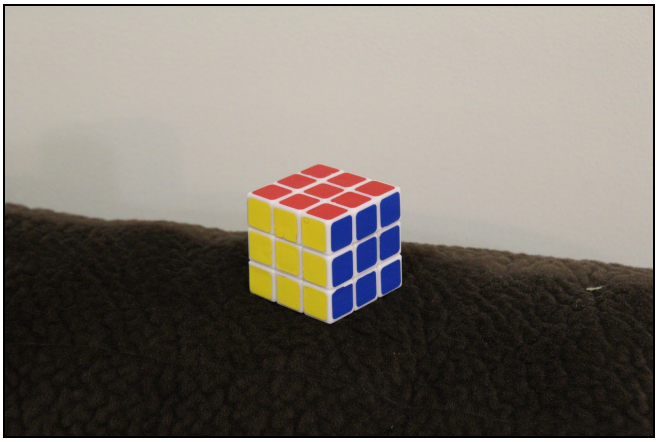
The sample image.



The lighting image.



The scaled image.



The rotation image.

2) OpenCV Implementation

An `ORB` object is first created, then the keypoints and descriptors for each image are computed using `detectAndCompute()`.

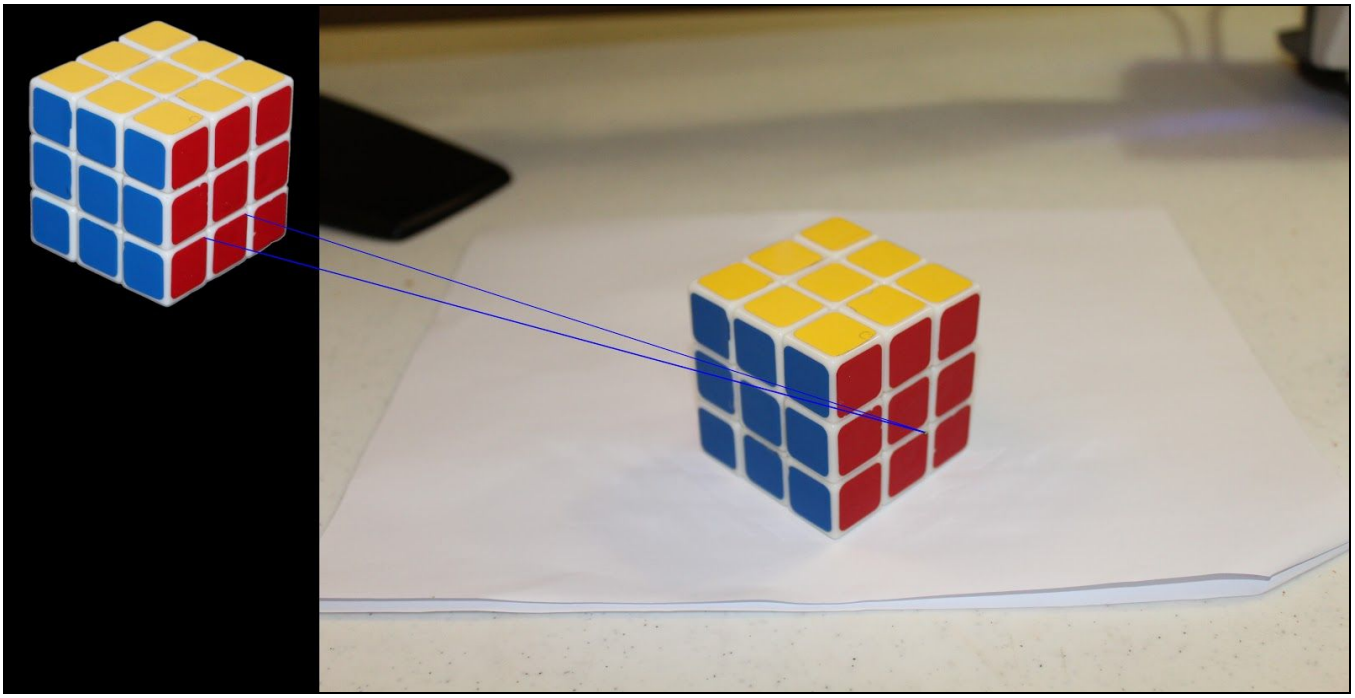
Now that we have keypoints and descriptors for both images, a `BFMatcher` object is created using the `NORM_HAMMING` normType. I decided leave crossCheck set to false when creating this object. The reason for doing this is so I could perform the ratio test against the matches returned from `knnMatch()`. More information about the ratio test can be found in section 7.1 of David Lowe's *Distinctive Image Features from Scale-Invariant Keypoints*.

Since we are interested in only the top `10` matches, the matches are checked with it's nearest neighbor using an initially small ratio. If less than `20` good matches are found, the ratio threshold value is incremented by `0.01`, and the matches are checked again. This is repeated until at least `20` ratio-filtered matches are found.

Lastly, the list of matches is sorted according to it's distance, and first `10` elements in the list are returned.

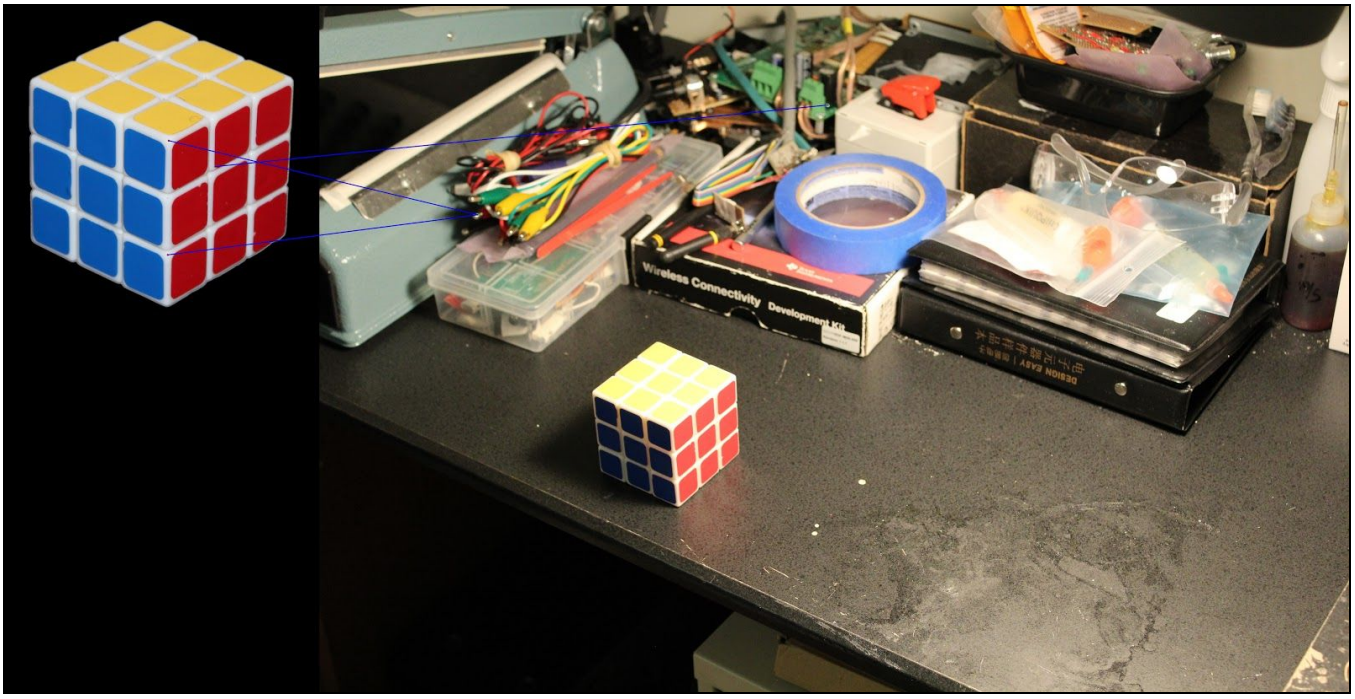
3) Results

Sample



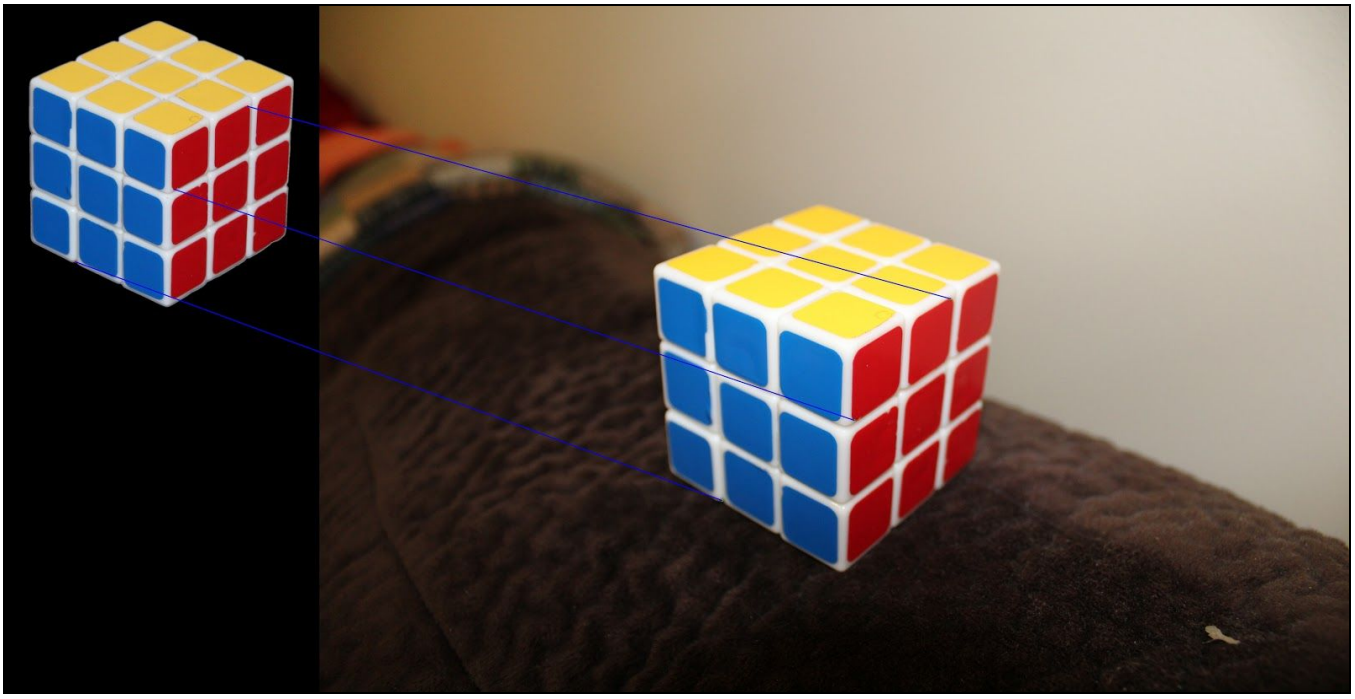
- *How many features does your algorithm get correctly?*
  - 4/10
- *Why do you think it gets some of the errors it gets?*
  - The picture was taken at a higher angle with respect to the rubik's cube. The number of areas that are idential in color & shape makes it difficult to distinguish one red square apart from another red square. For example, 6 points were 1 red block away from being correct matches.
- *Did you try taking the picture multiple times to try and improve results? What did you do to make this work?*
  - Yes, multiple pictures were taken. The picture with the best (most uniform) lighting & angle was used.

Lighting



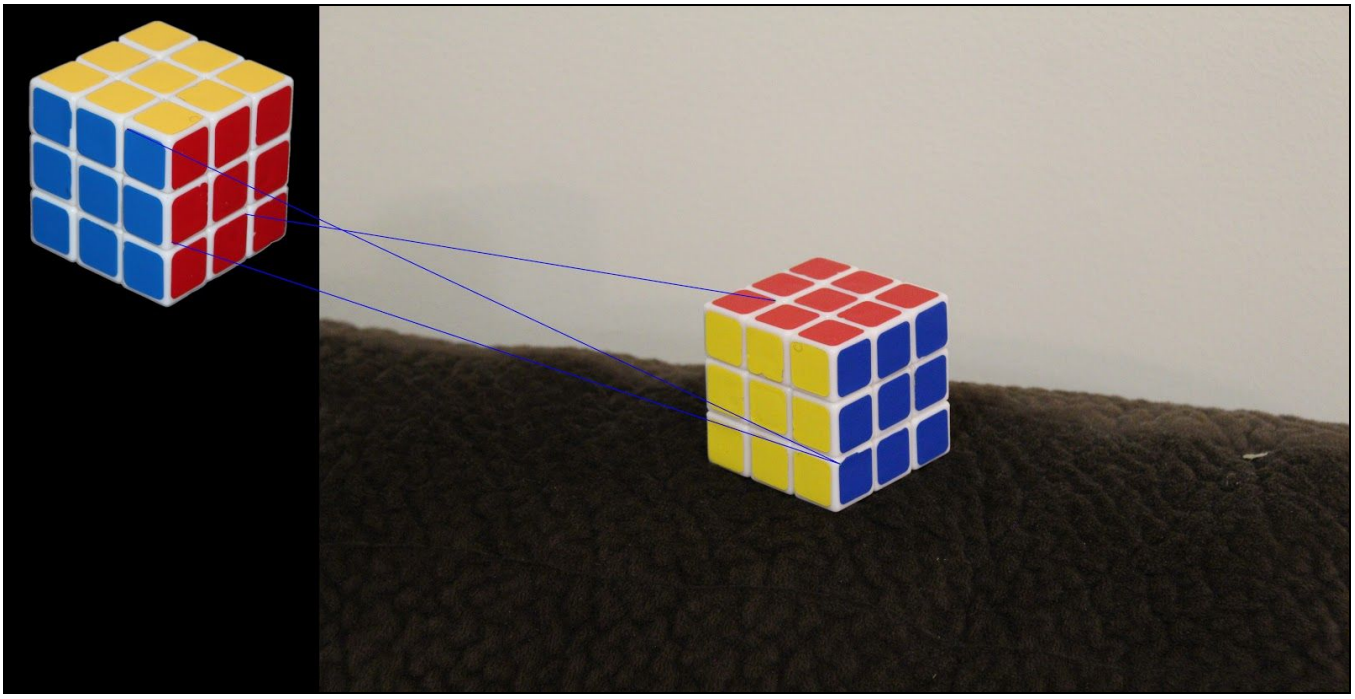
- *How many features does your algorithm get correctly?*
  - `0/10`
- *Why do you think it gets some of the errors it gets?*
  - There are many objects in the picture other than just the rubik's cube. The lighting across the cube also varies from one side to the other (brighter on yellow & red sides, darker on blue side).
- *Did you try taking the picture multiple times to try and improve results? What did you do to make this work?*
  - Multiple pictures were taken, but they were all similar to what's shown above. None provided good results.

Scale



- *How many features does your algorithm get correctly?*
  - `10/10`
- *Why do you think it gets some of the errors it gets?*
  - No errors.
- *Did you try taking the picture multiple times to try and improve results? What did you do to make this work?*
  - Yes, with/without flash to match the lighting on the cube. Pictured above is with the camera flash.

Rotation



- *How many features does your algorithm get correctly?*
  - `0/10`
- *Why do you think it gets some of the errors it gets?*
  - The number of areas that are idential in color & shape makes it difficult to distinguish one yellow square apart from another yellow square. For example, 5 points were really close, but they were 1 block off on the rubik's cube.
- *Did you try taking the picture multiple times to try and improve results? What did you do to make this work?*
  - Multiple pictures were taken, but they all had similar results.

All source files are hosted on GitHub @ https://github.com/jjones646/immatch