

Input Images:

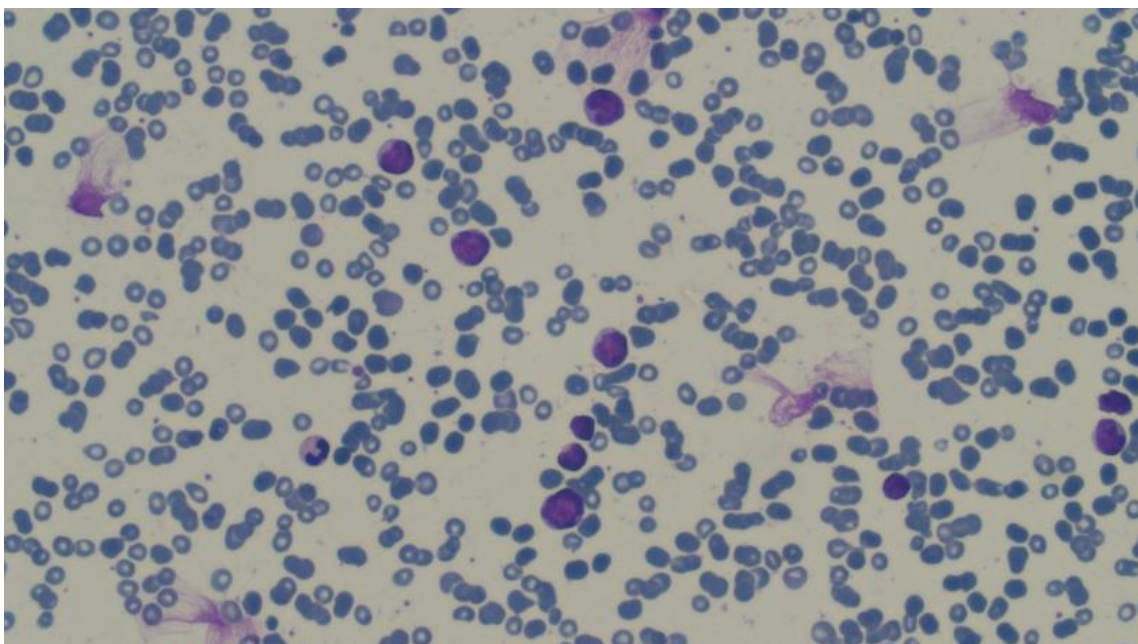


Image 1

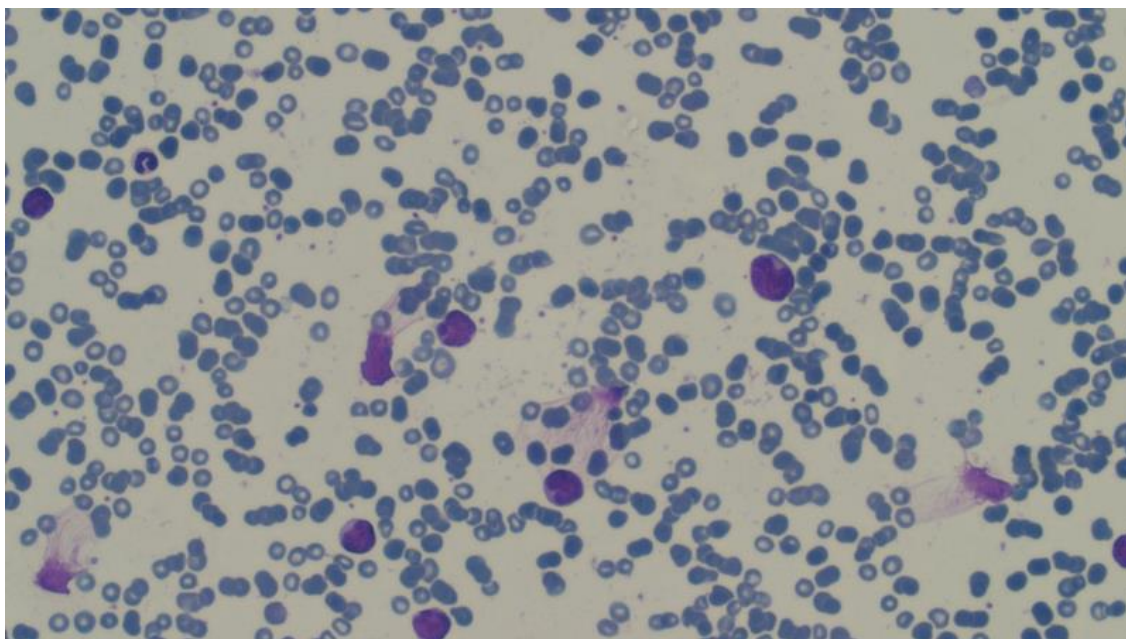
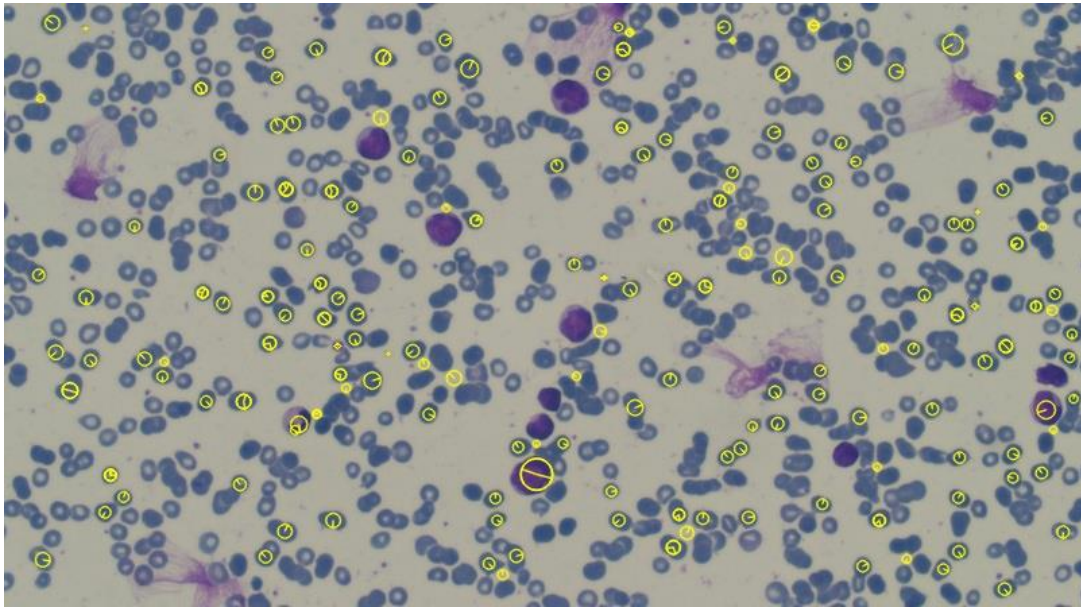
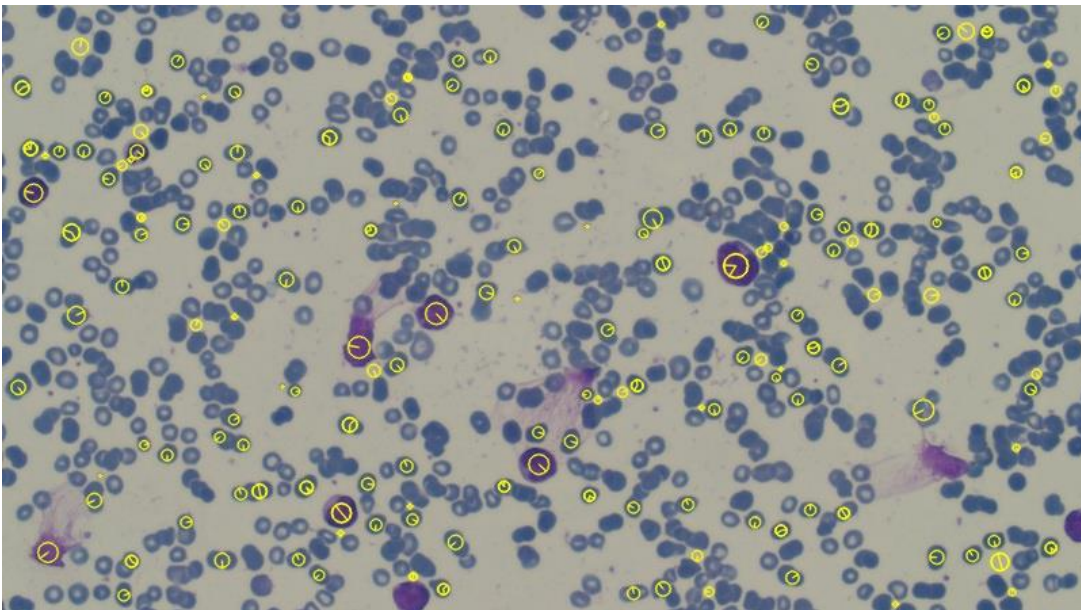


Image 2

Local Features:



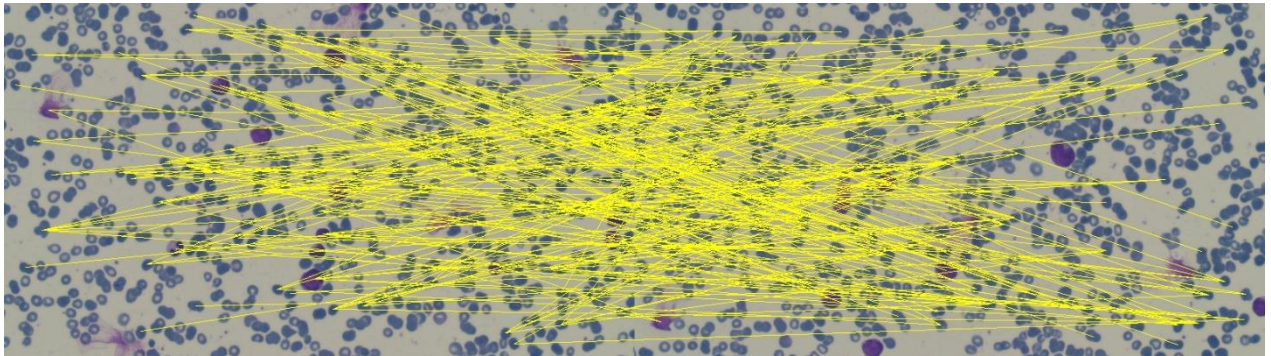
Feature Points (Image 1)



Feature Points (Image 2)

For both the pictures we get the above calculated feature points respectively as marked on the pictures with the **SIFT** command for extracting features and descriptors.

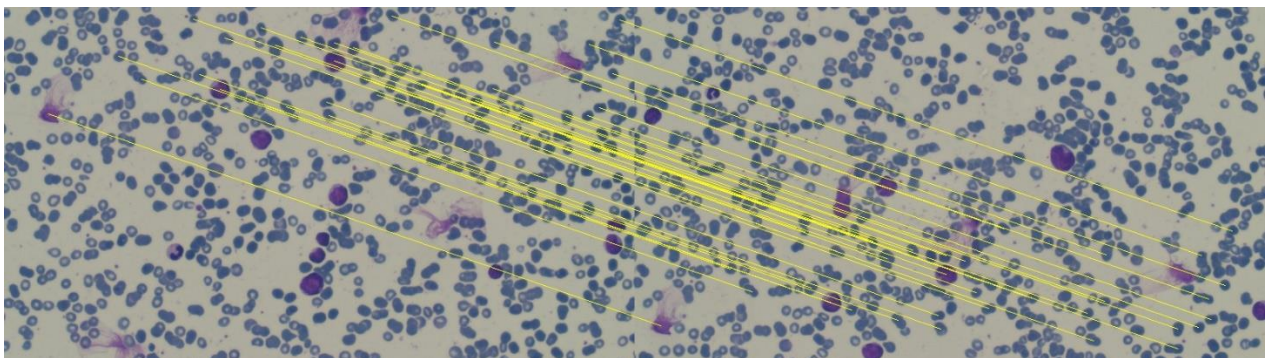
Euclidean Distance calculation:



Matching Feature Descriptors

We find the correspondence between these feature descriptors from the last part using Euclidean distance. *scipy.spatial's distance* was used for this purpose. Both images were stitched, and lines drawn across them. We matched approximately 200 points in both images.

RANSAC to remove outliers:



Best Matches

From the matches of the last part, we have removed the outliers with RANSAC. The above shown output is after this process. It represents the inliers of our data. The result is a stitched image like the last part but only showing the key points that matter.

Panorama Image:

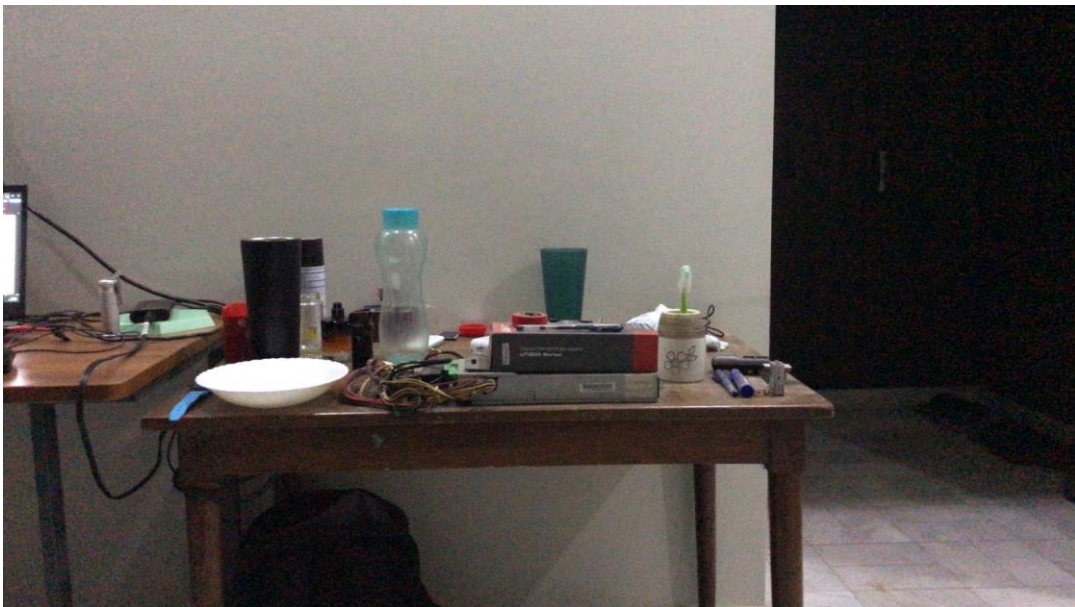
Unfortunately, we were unsuccessful in computing a panorama.

Testing Algorithm on Natural Images

Input Images:

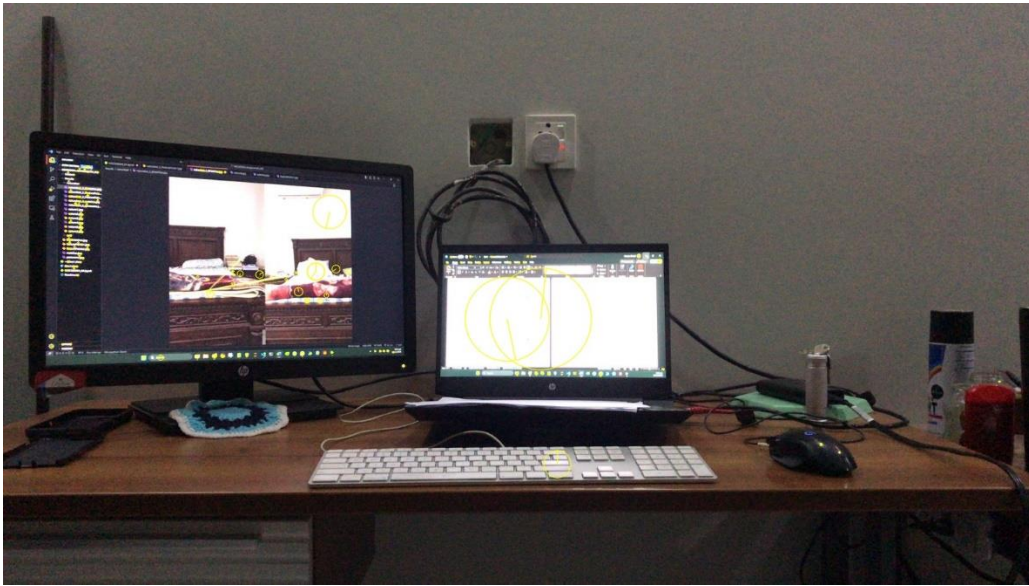


A graduate student's natural habitat 1

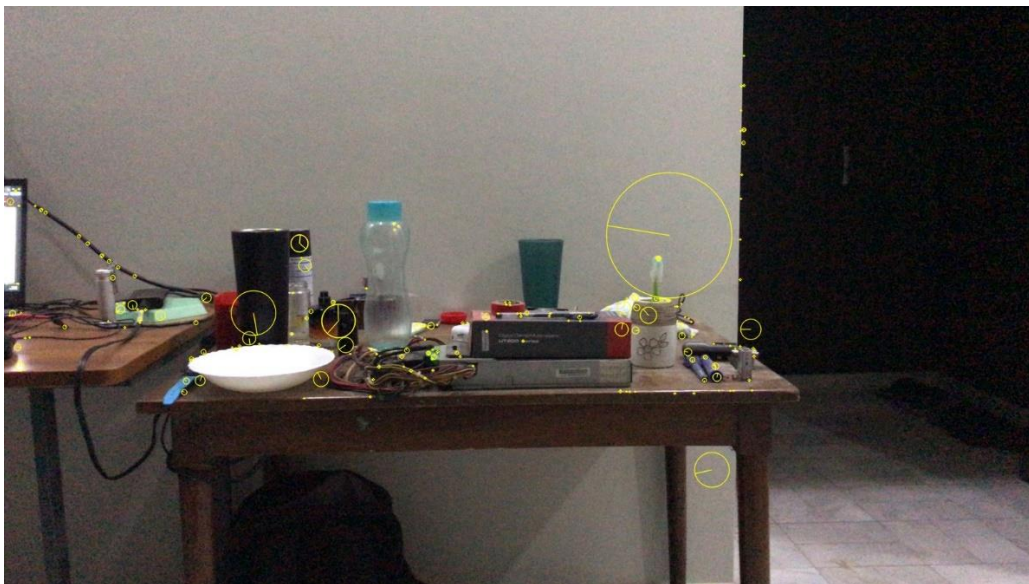


A graduate student's natural habitat 2

Local Features:

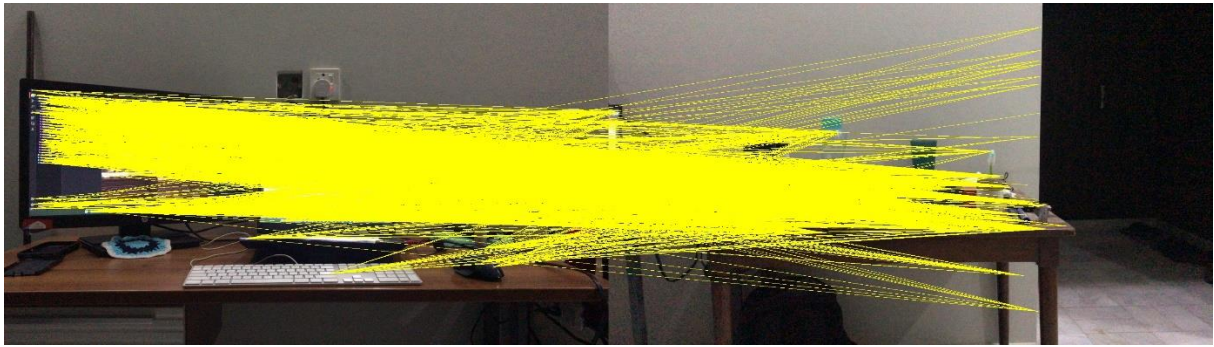


Local Features Image 1



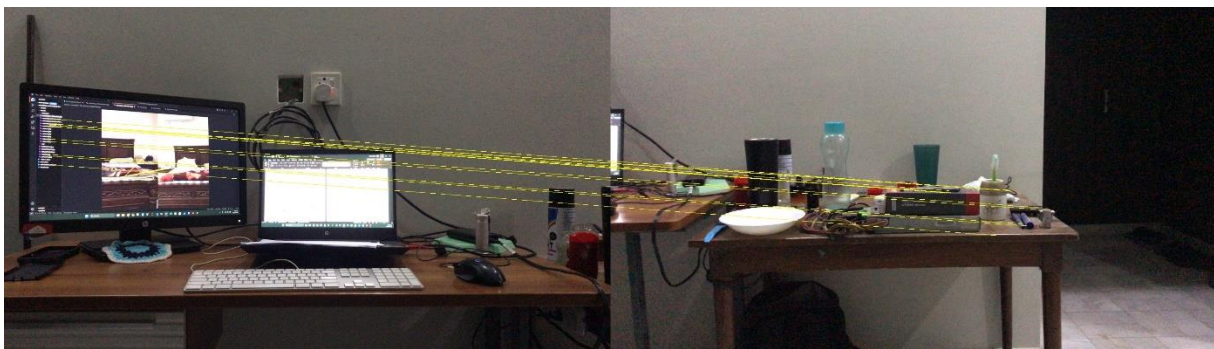
Local Features Image 2

Euclidean Distance calculation:



Matching Feature Descriptors

RANSAC to remove outliers:



Best Matches

Panorama:

Unfortunately, we were unsuccessful in computing a panorama.

Issues and solutions:

Our panorama code returned an empty image with a certain color from the first image's array. Despite repeated efforts to debug the issue we were unable to compute a panorama for either set of images, even though our code logic is apparently correct.

