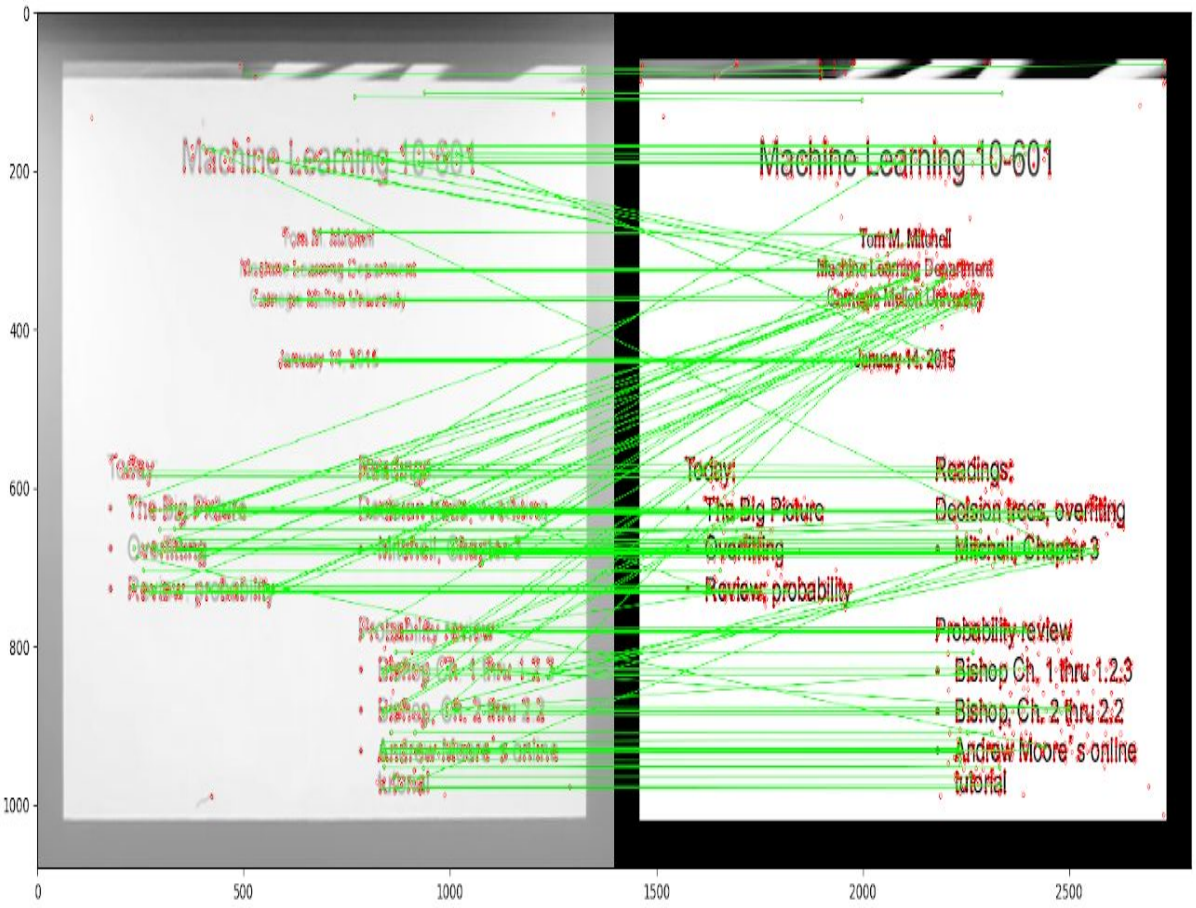
DSAA

PROJECT

By - 2071209 & 20171210



In this picture I've shown a mapping between the matching points of the two pictures according the technique I've used.

Method Used:

We have used opencv to solve the problem , along with that numpy and matplotlib have also been used. What we did initially was use brute force matcher. What it does is, it takes one descriptor of one feature in the first set and then this feature is matched with the rest of the features taken in second set using some distance calculation and of all the features the closest one is returned.Now, the features taken and computed using a technique SIFT(Scale Invariant Feature Transform) , which extract keypoints and compute its descriptors. This technique was discovered in 2004.

Now for each frame I've taken the number of non matching points with a slide and from them the one with the least count is the matched part.

To do better than the Brute Force method we switched to FLANN which stands for Fast Library for Approximate Nearest Neighbors. It contains a collection of algorithms optimized for fast nearest neighbor search in large datasets and for high dimensional features. It works more faster than BFMatcher for large datasets. We use it for the calculation of the matching points instead of brute force.

In the SearchParams. It specifies the number of times the trees will be recursed for matching features. Higher values gives better precision, but also takes more time. To change the value we just have to change the value of checks in the line -> `search_params = dict(checks=100).`

Over to all frames :

We loop over all the frames for one slide, calculating for each frame how much of it's features match with each of the slides and we store the number of the particular frame with it's matching slide and write it into the <roll num.txt> file.

The complexity of this method is O(N*M) , where N is the number of slides and M is the number of frames.