Student: Arzu Fatullayeva

Week: 01.06.23 – 08.06.2023

During this week I have tried to implement ORB for mammogram images to see fist results how it works on them, and detect further improvement steps.

The first version of code uses the ORB (Oriented FAST and Rotated BRIEF) algorithm for keypoint detection in mammogram images.

The main contributions of ORB are as follows:

- Fast and accurate orientation component: ORB improves upon the FAST keypoint detector by adding a component that efficiently computes the orientation of keypoints. This enhancement enhances ORB's robustness to image transformations, such as rotation.
- Efficient computation of oriented BRIEF features: ORB utilizes the BRIEF descriptor, which efficiently represents local image patches. It extends BRIEF by incorporating orientation information obtained from the previous step, allowing it to compute oriented BRIEF features that capture both intensity patterns and keypoints' orientations.
- ORB analyzes the variance and correlation of the oriented BRIEF features to ensure their distinctiveness and robustness. By considering statistical properties, ORB can select more informative features and discard less reliable ones, resulting in improved performance.
- ORB employs a learning method to decorrelate the BRIEF features, particularly under rotational transformations. This technique improves ORB's performance in nearest-neighbor applications, which are critical for matching keypoints across different images.

Here's a breakdown of how the code detects keypoints:

The training image, which is a mammogram, is loaded using cv2.imread() and converted to RGB color space using cv2.cvtColor(). Then, it is further converted to grayscale using cv2.cvtColor() again. This step is performed to simplify the image and make it suitable for keypoint detection.

The ORB (Oriented FAST and Rotated BRIEF) algorithm is instantiated using cv2.ORB_create(). ORB is a fusion of the FAST keypoint detector and the BRIEF descriptor. It is designed to be fast and efficient for real-time applications.

Keypoints and descriptors are computed for the training image using orb.detectAndCompute(training_gray, None). The detectAndCompute() function detects keypoints in the grayscale image and computes descriptors for those keypoints. The keypoints represent distinctive and salient points in the image, while the descriptors encode information about the local image patch around each keypoint.

The keypoints are visualized on the training image using cv2.drawKeypoints(). This function draws circles at the location of each keypoint on the image.

The same steps are then applied to the test image, which is a transformed version of the training image. The test image is downsampled and rotated to simulate scale invariance and rotational invariance.

Finally, the keypoint matching step is performed using the Brute Force Matcher (cv2.BFMatcher). The ORB descriptors of the training image and the test image are compared, and the matches are sorted based on their distances. The best matching points are then visualized using cv2.drawMatches().

It's important to note that the effectiveness of keypoint detection and matching in mammogram images can vary depending on the specific characteristics of the images and the nature of the abnormalities being detected. Further customization and fine-tuning of the algorithm parameters may be required to achieve optimal results for mammogram analysis.