Arzu Fatullayeva

arzu.fatullayeva@gwmail.gwu.edu

**Baku 2023**

Report 4  
Course: Guided Research I  
Project title : Finding informative regions in grayscale mammogram images.

**COMPUTER SCIENCE AND DATA ANALYTICS**

**Logo

Description automatically generated**

****

# Detailed application of ORB to mammogram images.

## This report will provide a comprehensive and detailed explanation of how ORB processes an image, addressing the feedback received in response to reports 2 and 3.

## Before applying ORB

Before starting to apply ORB for mammogram images, difficult negative case images were removed. Excluding difficult negative cases during data cleaning is a common practice in machine learning and data analysis.

Difficult negative cases refer to those instances that are particularly challenging for the model to classify correctly as negatives. These cases may exhibit characteristics or patterns that are similar to positive cases, leading to potential confusion for the model during training or evaluation. In some situations, including difficult negative cases in the dataset could cause the model to perform poorly or result in unintended biases.

By excluding difficult negative cases, the goal is to create a more balanced and representative dataset that allows the model to focus on learning from the clear, well-defined negative instances. This approach can improve the model's generalization and performance on new, unseen data.

# ORB

In simpler terms, the ORB (Oriented FAST and Rotated BRIEF) algorithm is a feature detection technique that performs on par with the popular SIFT (Scale-Invariant Feature Transform) method, while surpassing SURF (Speeded-Up Robust Features) in terms of speed. ORB combines two well-known techniques: FAST keypoint detector and BRIEF descriptor, both known for their good performance and efficiency.

The key contributions of ORB are as follows:

1. Fast and Accurate Orientation Component: ORB enhances the FAST keypoint detector by adding a component that efficiently determines the orientation of the keypoints.

2. Efficient Computation of Oriented BRIEF Features: ORB efficiently computes the BRIEF features with orientation information for the keypoints.

3. Analysis of Variance and Correlation: ORB analyzes the variance and correlation of the oriented BRIEF features, allowing for better performance in various applications that require feature matching.

4. Learning Method for Decorrelation: ORB employs a learning method to decorrelate the BRIEF features under rotational invariance, which improves performance in tasks involving nearest-neighbor matching.

Overall, ORB is a powerful feature detection technique that maintains accuracy comparable to SIFT, outperforms SURF in terms of speed, and is particularly attractive due to its efficient and cost-effective nature.

## Implementation



Fig.1 ORB implementation

The code snippet provided on Fig. 1 is using the OpenCV library in Python to perform keypoint detection using the ORB (Oriented FAST and Rotated BRIEF) feature detector and descriptor.

How does ORB work here?

FAST (Features from Accelerated Segment Test) is an image processing technique that analyzes the brightness of a central pixel "p" by comparing it to the brightness of 16 surrounding pixels forming a small circle around "p." These surrounding pixels are divided into three classes: those that are lighter than "p," those that are darker than "p," and those that have a similar brightness to "p."

To identify keypoints in the image, FAST examines whether more than 8 pixels in the circle are either brighter or darker than the central pixel "p." If this condition is met, "p" is selected as a keypoint. These keypoints are crucial as they provide valuable information about the locations of edges in the image, helping to identify important features for further processing and analysis.

The ORB (Oriented FAST and Rotated BRIEF) algorithm addresses the limitations of FAST features by incorporating a multiscale approach. It achieves this by creating an image pyramid, which is a representation of the original image at multiple scales or resolutions. Each level in the pyramid contains a downsampled version of the image compared to the previous level.

With the image pyramid in place, ORB utilizes the FAST algorithm at each level to detect keypoints. By doing so, it effectively identifies keypoints at different scales. This means that ORB is partially scale invariant, as it can detect the same features across different scales of the image.

By incorporating the multiscale image pyramid and leveraging the fast keypoint detection technique, ORB enhances its ability to capture features and edges in an image at varying levels of detail, making it a robust and versatile feature detection algorithm for image processing tasks.

Once ORB has identified keypoints in the image using the FAST algorithm, it proceeds to assign an orientation to each keypoint. This orientation helps to determine the direction in which the keypoint is facing, such as left or right.

To calculate the orientation, ORB employs a technique called intensity centroid. This method assumes that in a corner-like structure, the intensity is shifted or offset from the center of the keypoint. By analyzing the intensity values around the keypoint, ORB identifies the centroid of these intensity variations. This centroid can be viewed as a vector that points towards the direction of the corner's intensity shift.

Using the intensity centroid vector, ORB is able to impute the orientation of the keypoint, representing whether it is facing towards the left or right direction. This orientation information enhances the descriptor's ability to capture the local features accurately and facilitates more robust matching and recognition of keypoints across different images.

After the FAST algorithm has located keypoints across different scales, the BRIEF (Binary Robust Independent Elementary Features) algorithm comes into play. BRIEF takes these keypoints and converts them into binary feature vectors, which serve as concise representations of the keypoints and collectively describe the object or scene.

The binary feature vector, also known as the binary feature descriptor, is a compact representation containing only 1s and 0s. For each keypoint, BRIEF generates a feature vector, typically consisting of 128 to 512 bits. Each bit in the vector represents a specific comparison between pixel intensities in the region surrounding the keypoint.

These binary feature vectors efficiently capture the distinctive characteristics of the keypoints. They enable a streamlined and memory-efficient way to represent the object or scene, making it easier to compare and match keypoints in different images and perform various computer vision tasks efficiently.

To address the issue of rotation invariance, ORB introduces a modified version of BRIEF called rBRIEF (Rotation-aware BRIEF). While BRIEF is fast and efficient in generating binary feature vectors for keypoints, it lacks rotational invariance, meaning it may not handle variations in the object's orientation well.

Incorporating rBRIEF, ORB aims to preserve the speed advantages of BRIEF while enhancing its capabilities to handle rotations. The rBRIEF algorithm considers the orientation of each keypoint and adjusts the binary feature descriptor accordingly. By taking into account the keypoints' orientations, ORB ensures that the resulting feature vectors remain consistent even when the object or scene undergoes rotations.

Below Fig.2 and Fig.3 shows the result of ORB for 10 and 500 number of keypoints.

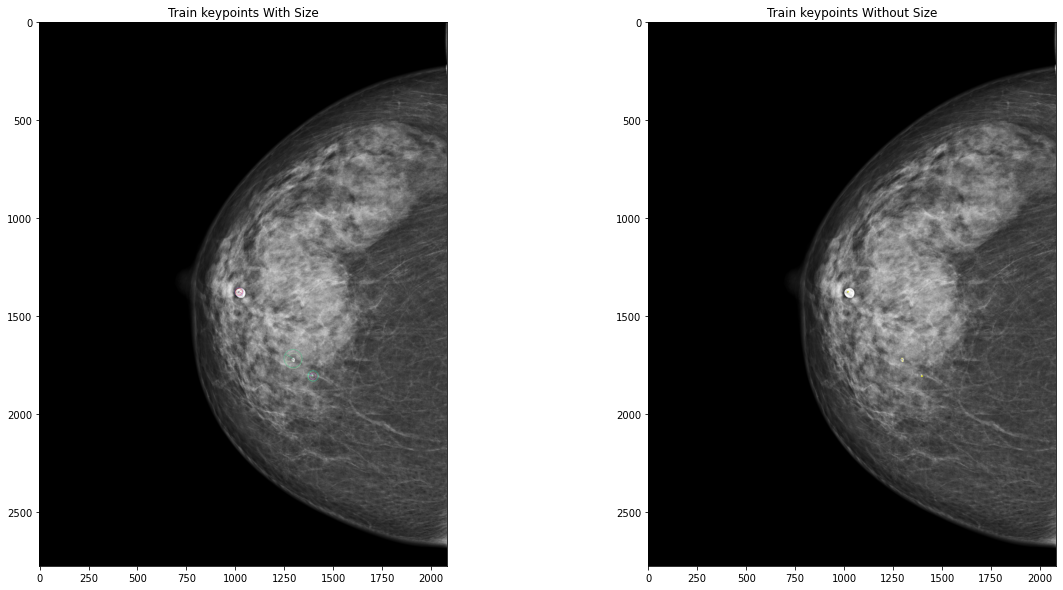


Fig.2 Implementation of ORB for the number of keypoints=10

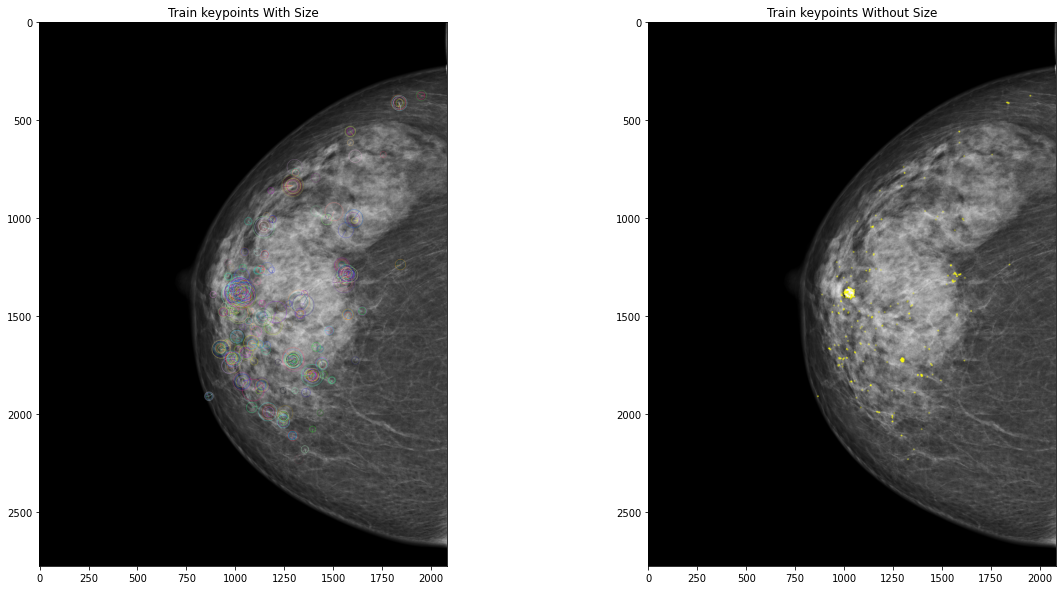


Fig.3 Implementation of ORB for the number of keypoints=500

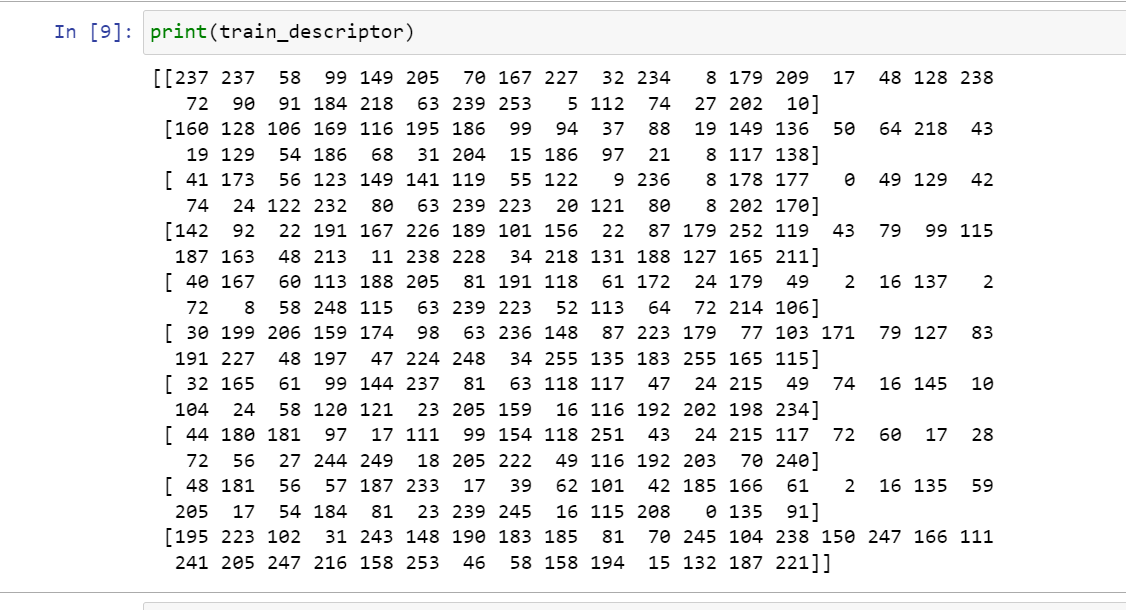


Fig. 4 Descriptors for 10 features.

Descriptors obtained from ORB used as features for ML models. The descriptor is derived from the rBRIEF (Rotation-aware BRIEF) algorithm and is used to describe the local image content around a detected keypoint.

For each image we have for example 10 features, each feature array of 32 (10,32) as on Fig.4.