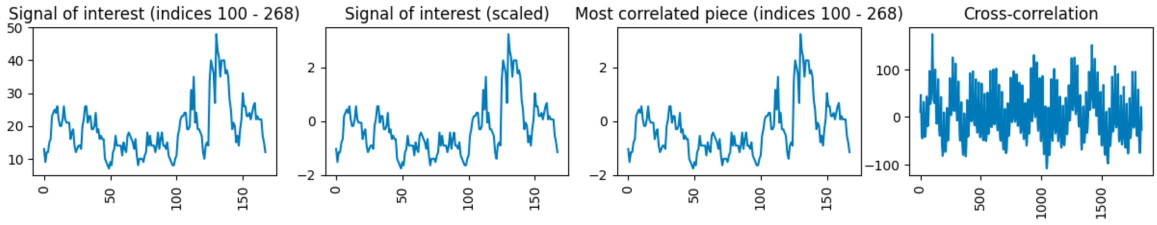
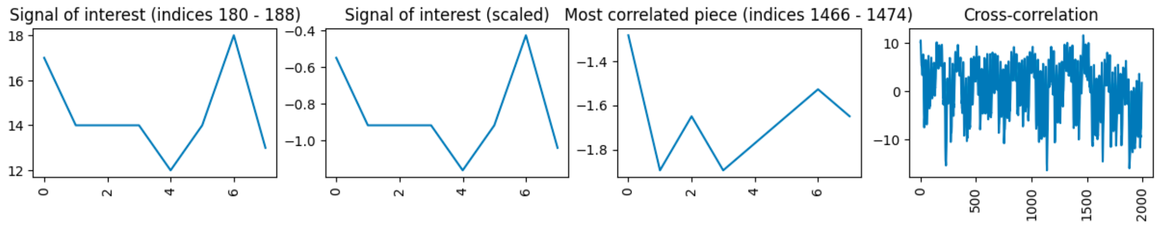
gbsv Mini-Challenge 2 - Wichtigste Resultate

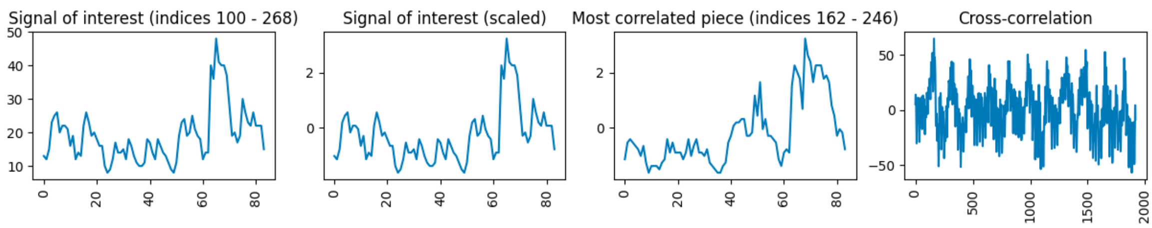
1. Find signal of interest from the original full signal by cross correlation



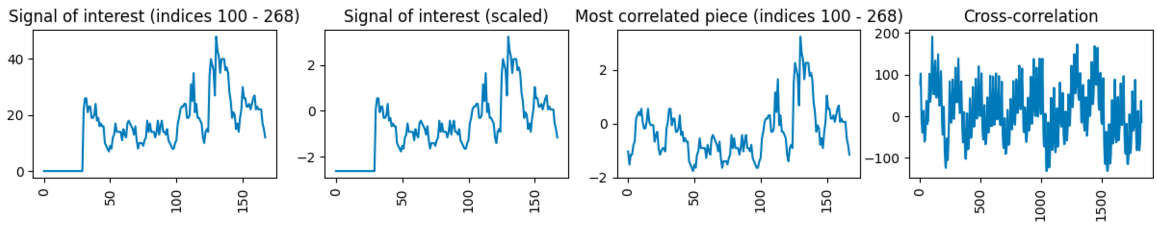
My selected data is a traffic dataset which records the number of vehicles crossing a junction in every hour. By my observation, one cycle is around 168 hours (one week). So, I select a piece of signal with length 168 from the original full signal. By cross correlation, correlations will be calculated between my signal interest and every possible same length signal in the original signal. Larger correlation factor means higher similarity. Here, the most correlated piece is same as my signal of interest. This means cross correlation could find this signal from original data. When I shift the signal of interest at x-axis, it could still be found. Scaling the amplitude of the target signal, it could still be found.

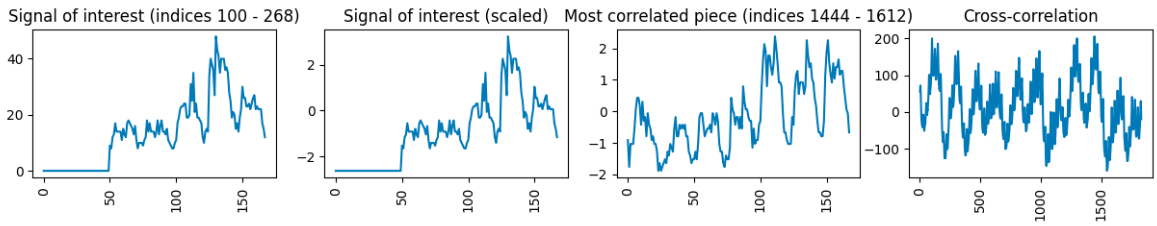


However, when the target signal is very short, it could not be found from original data by cross correlation. This means, the target signal must have a certain minimal length, so that it could be recognized from the full signal.



Reducing the target signal frequency, it also could not be found by cross correlation.

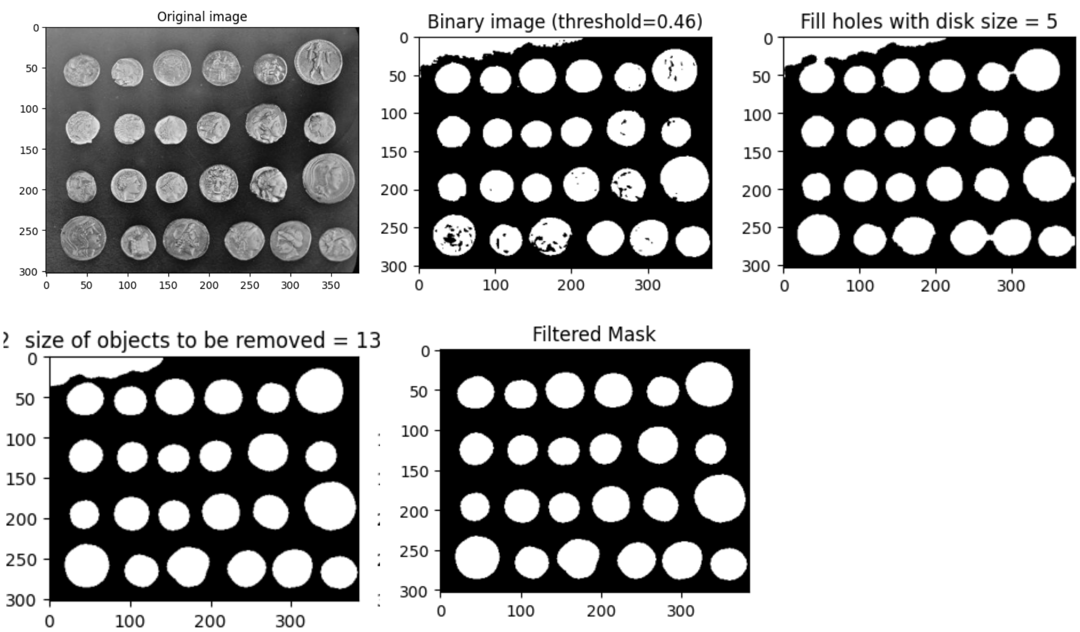




When changing short parts of the signal (upper row), cross-correlation still works finding the target signal. However, when changing a longer part of the signal (row below), the target could not be found anymore.

- time shift: Taking another signal piece with same length but shifted time is usually tolerated. This means it still could be found by cross-correlation.
- signal length changes: Taking longer or slightly shorter signal is tolerated. The signal will still be found. However, when largely reduce the target signal length to a certain level, the signal will not be found from the full signal.
- amplitude scaling: The signal piece with scaled amplitude could still be found by cross correlation.
- frequency changes: It is not toleranted, the signal with changed frequency will not be found.
- Waveform structure changes: When replacing minimal parts of the signal amplitude with 0, usually the signal could still be found. However, when the changes part reach a certain length, the signal could not be recognized from the full signal anymore.

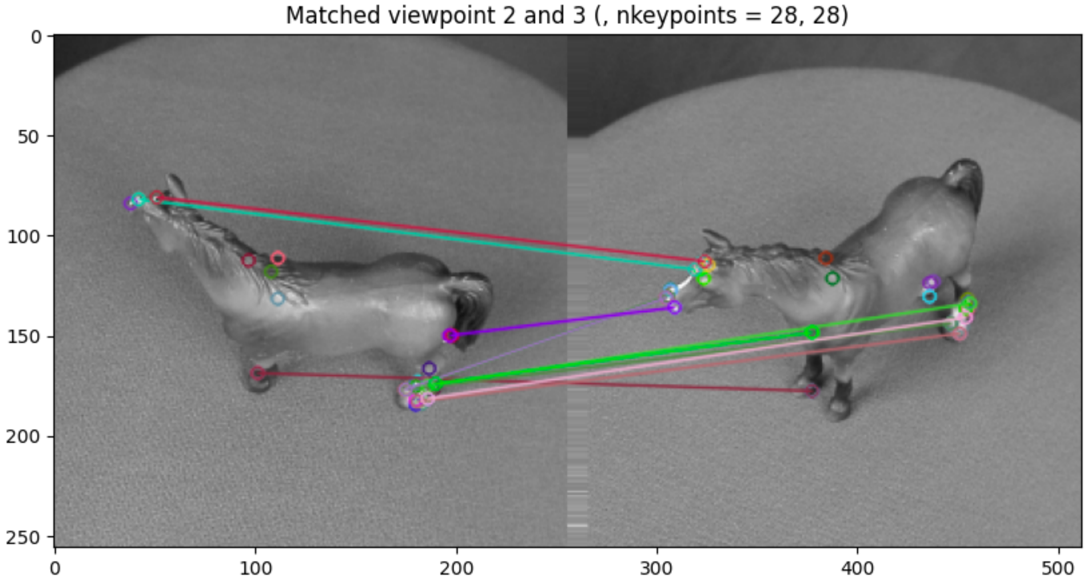
2. Segmentation

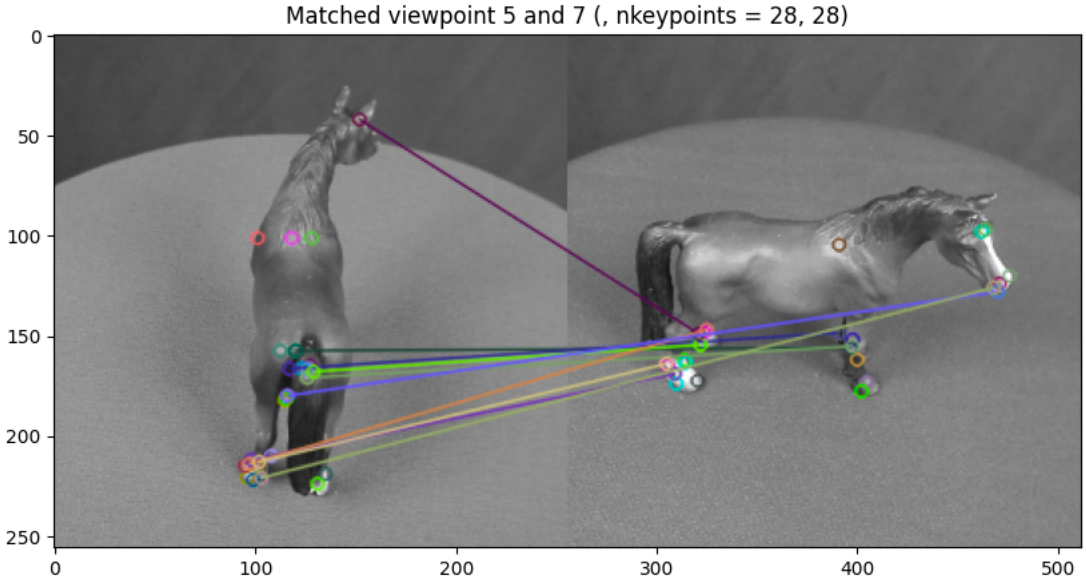


The goal is to segment all coins from the orignal image. Although the coins look brighter than the background, some shadow pixels in the middle of the coins are darker than background. Firstly, converting the image pixel values to binary (1 and 0) whrere pixel value = 1 should represent as many pixels as possible for the coins, pixel value = 0 should represent the background. Set a threshold value, all pixels with value larger than threshold will set to 1, others to 0. Increasing threshold, more pixels will be converted as black, less as white, with threshold of 0.6, the converted image could remain the coin area most accurately. The false labeled background object at the left corner could be removed in later step. Secondly, filling the holes by morphology closing operation. With a large disk size, it will fill large holes, however, the irrelevant space will also be filled, for instance the space between the coins may be filled. With disk size of 5, the operation has the best balance between filling the relevant holes as many as possible and filling the irrelevant holes as little as possible. Next step, the connections between the coins will be removed by morphology open, which will remove small objects with size smaller than the parameter size. With size of 13, the coins have the shape most similar to the original coins. Finally, all individual objects are extracted and plotted. The left upper corner object name is identified and removed.

3. ORB keypoint descriptors to match images

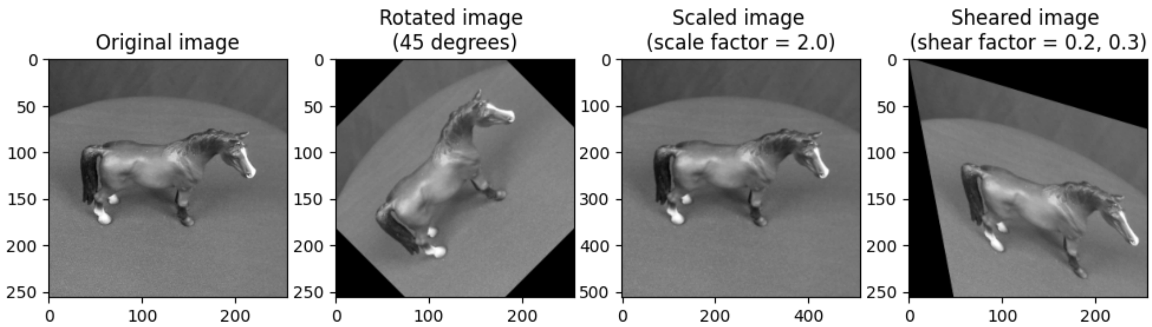
Use Images of one object with eight viewpoints



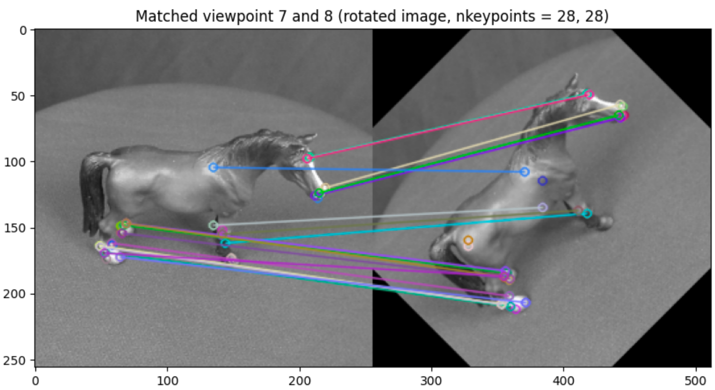


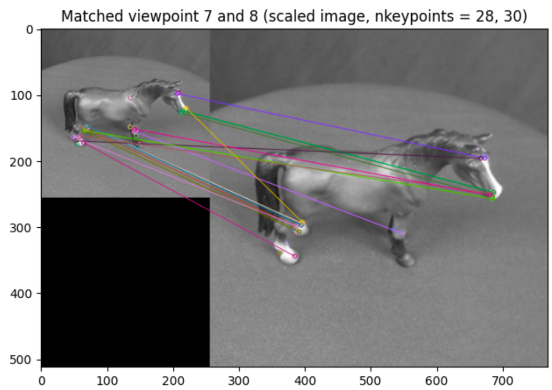
Though the eight viewpoints differences seem not large, most of the images match not good. From all 28 pairs (with various ORB parameters), only the pair of viewpoints 2 & 3, and viewpoints 7 & 8 have good matches. Upper first pair of images is a good match, 2 features on head and 3 features on back foot are correctly matched. The second pair is not a good match, because there is no correct matches. This is because, ORB algorithm detects keypoints based on the intensity of the pixels. When the main detected keypoints (horse head, feet, tail) of two images that are under similar lighting conditions, the algorithm can match two images well. Otherwise (e.g. horse face has very small pixel intensities in viewpoint 5 and larger intensities in viewpoint 7), the images cannot be matched well by pixel intensity based algorithm.

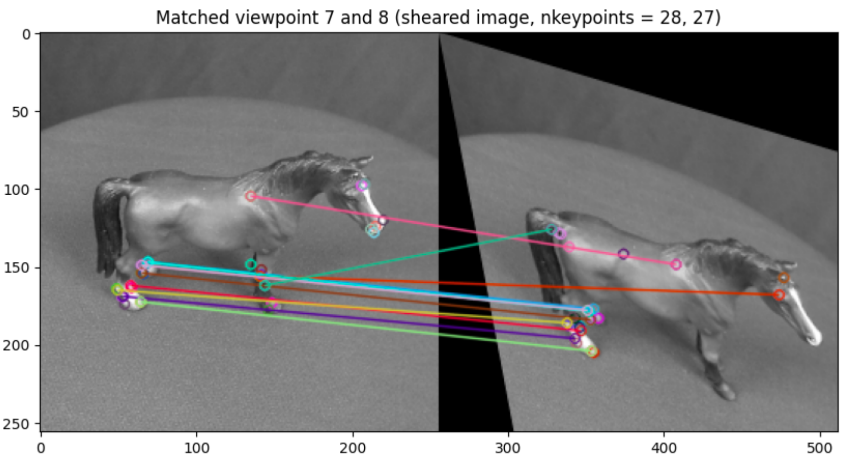
Use transformed images from one original image



I selected image 7, and transformed with rotation, scaling, shearing on both x and y axis. Then match each of them with the original image.







From the results, we could see that the ORB algorithm is very robust to rotation and scaling. This is because ORB detects and matches keypoints invariant to changes in rotation, scale, and lighting conditions. ORB assigns orientations to keypoints, allowing it to detect and match features even when the image is rotated. ORB uses a pyramid structure to handle different scales, making it robust to scaling transformations. It detects features at the most appropriate scale.

ORB can tolerant to shearing, though the performance is not as effectively as to rotating and scaling. Because shearing can distort the local pixel intensities and alter the relationships between keypoints, making feature matching more challenging.

In summary, ORB can effectively detect and match features in images with rotation and scaling transformations. Its performance may largely decrease involving objects with significantly different viewpoints or orientations across images.

Code

The gitlab repository is already shared with susanne.suter@fhnw.ch