

## Task 2. Computer vision. Sentinel-2 image matching - Demo

In this task, you will work on the algorithm (or model) for matching satellite images. For the dataset creation, you can download Sentinel-2 images from the official source here or use our [dataset from Kaggle](#).

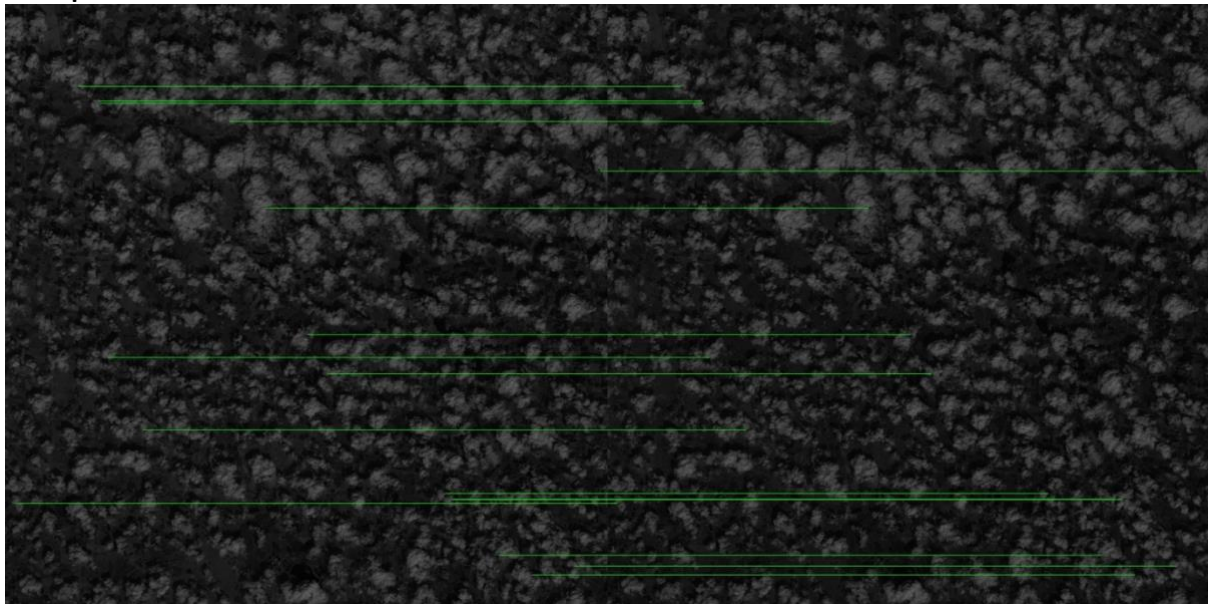
We will visualize a maximum of 20 matches for the purposes of this demonstration.

Input images:

T36UYA\_20190805T083601\_B07.jp2

T36UYA\_20190805T083601\_B06.jp2

Output:



The matching appears to be effective, particularly when comparing different bands of the same image. It is logical for corresponding keypoints to align in both bands. However, it is important to emphasize that the current method does not provide an exact evaluation of performance.

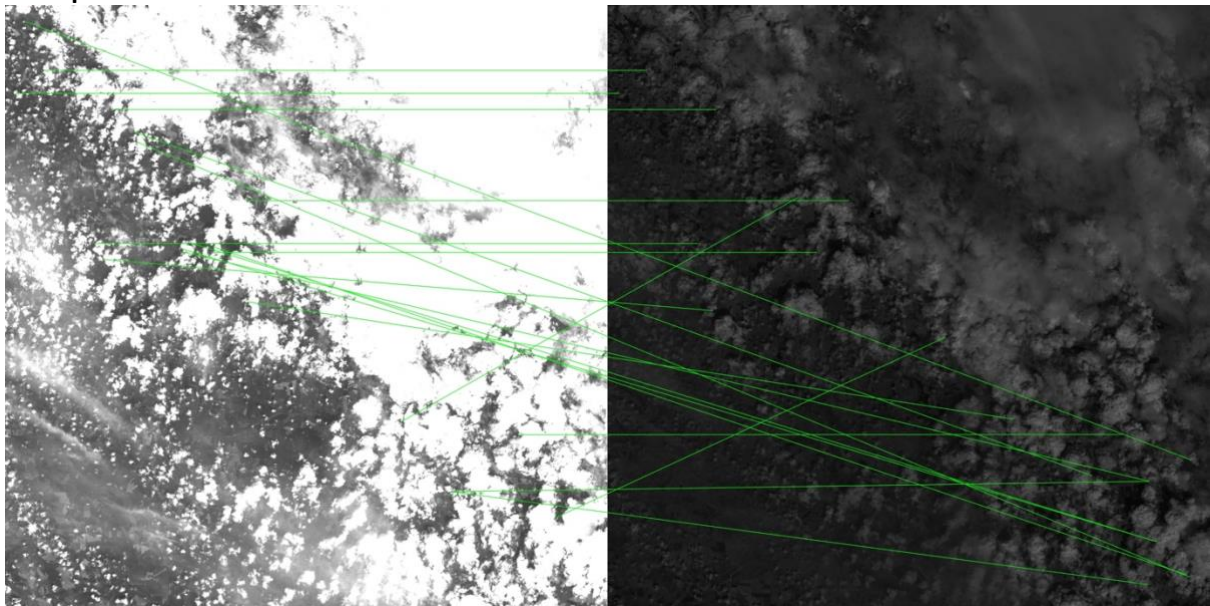
We will visualize a maximum of 20 matches for the purposes of this demonstration.

Input images:

T36UYA\_20190726T083611\_TCI.jp2

T36UYA\_20190726T083611\_B8A.jp2

Output:



The proposed solution employs keypoint detection and image matching functionalities from OpenCV:

- Keypoint detection and descriptor extraction are carried out using the `.detectAndCompute`` function of ORB.
- Matching is performed using the brute-force matcher (BFMatcher in OpenCV), leveraging Hamming distance to determine the closest descriptors.

While this represents a fundamental approach, it is essential to acknowledge that this simplistic solution may not deliver optimal outcomes when applied to images captured during different seasons.