Task 2. Computer Vision. Sentinel-2 Image Matching

Dataset Creation

All the operations for the dataset creation and loading on Google Drive are done in *"Image Matching Dataset notebook."*

The data was downloaded from Kaggle, from the link in the *"Test_Task_Internship.pdf"* file.

I have decided to use 3 images for testing purposes with 3 different weather conditions:

- *Image with a little bit of clouds and without snow.*
- *Image with a little bit of clouds and with snow.*
- *Image with a lot of clouds and without snow.*

You can notice that there are two types of images:

- **TCI (True Color Image) -** The main type for algorithm testing.
- **NDVI (Normalized Difference Vegetation Index) -** The type that I didn't use eventually.

The reason I even considered NDVI is that I was curious if there were any changes between the TCI color and any other color for image matching algorithms. I found out that **TCI** is better for keypoint detection tasks, so I decided not to include **NDVI** in the notebook with algorithms.

After selecting 3 images with different weather conditions, they were uploaded to Google Drive.

Methods for Image Matching

All the algorithms and inferencing are in the *"Sentinel-2 Image Matching"* notebook.

I used 3 algorithms and compared their outputs.

FLANN-Based Matcher

- **Feature Detection and Description:**
  First, the algorithm extracts keypoints and descriptors from both images using **SIFT** (Scale-Invariant Feature Transform), a feature detection algorithm. The keypoints are the interesting points or corners in the image, and the descriptors are vector representations of these keypoints.
- **Matching Descriptors:**
  FLANN (Fast Library for Approximate Nearest Neighbors) is used to match the

descriptors from both images. FLANN employs **kd-trees** to search for the nearest neighbors of each descriptor from the first image in the second image's descriptor set.

- **K-Nearest Neighbors Matching:**
  The `knnMatch` function finds the two nearest neighbors (**k=2**) for each descriptor in the second image, allowing comparison between the best and second-best matches.
- **Ratio Test:**
  A ratio test (Lowe's ratio test) is applied to filter out ambiguous or poor matches. A match is considered good if the distance of the first match is less than 70% of the second match.
- **Drawing Matches:**
  Good matches are visualized by drawing lines between matching keypoints in both images. Only those that pass the ratio test are shown, and the number of good matches is printed.

LightGlue

LightGlue is a recent approach, an evolution of **SuperGlue**, designed to match image keypoints efficiently using a neural network-based architecture.

- **Feature Extraction:**
  LightGlue relies on learned feature extractors like **SuperPoint** or **DISK** to detect keypoints and extract descriptors from the images. In my code, I choose between **SuperPoint** or **DISK** as the feature extractor depending on the configuration.
- **Transformer-Based Matching:**
  After extracting features, LightGlue uses a transformer-based model to find correspondences between the keypoints of two images.
- **Pruning and Refining Matches:**
  LightGlue includes mechanisms for **pruning** (removing less confident matches) based on the network's confidence in each match.
- **Homography Estimation (Optional):**
  The configuration includes parameters for estimating the **homography** between the two images using **RANSAC**. Homography refines the geometric relationship between matched keypoints by eliminating outliers.
- **Visualization:**
  In the `visualize_lightglue` function, matched keypoints are visualized between the two images. The confidence levels for each matched keypoint are also shown.

ALIKED with LightGlue

ALIKED (Adaptively Learned Invariant Keypoint Detector and Descriptor) is an improvement on **ALIKE**, introducing the **Sparse Deformable Descriptor Head (SDDH)** for more efficient descriptor extraction.

- **Learning-Based Keypoint Detection:**
  ALIKED uses a deep neural network to detect keypoints that are invariant to scale, rotation, and illumination changes.
- **Learning-Based Descriptor Extraction:**
  Once keypoints are detected, ALIKED generates descriptors that describe the local image regions around these keypoints.
- **Feature Matching:**
  ALIKED and LightGlue leverage the same **LightGlue matcher** for the keypoint correspondence step.

Results Analysis

Let's analyze the results from the three different methods that I have used for this task and understand why they perform differently across the two image matching scenarios: one with snow and the other with clouds.

1. FLANN-Based Matcher Results:

- *No Snow & No Clouds vs Snow & No Clouds*: 7 matched keypoints
- *No Snow & No Clouds vs No Snow & Clouds*: 42 matched keypoints

Analysis:

- **Low Number of Matches:**
  The FLANN-Based Matcher is based on **SIFT** for keypoint detection and descriptor extraction, which is a handcrafted method. **SIFT** extracts keypoints based on gradient changes in the image, which can be sensitive to significant changes in the scene, such as snow or clouds. Since snow drastically alters the surface appearance, there are very few shared keypoints between the two images (only *7 matches*).
- **Better Performance with Clouds:**
  The higher number of matches (*42*) in the second case can be explained by the fact that clouds, while obscuring parts of the image, do not entirely change the underlying features of the landscape. Therefore, **SIFT** can find more matching keypoints between the two images because some features remain visible even with clouds.

2. LightGlue Results:

- *No Snow & No Clouds vs Snow & No Clouds*: 199 matched keypoints
- *No Snow & No Clouds vs No Snow & Clouds*: 121 matched keypoints

Analysis:

- **Improved Performance:**
  LightGlue performs significantly better than **FLANN**, with *199 matched keypoints* in the

snow scenario and *121 matches* in the cloud scenario. This improvement is due to LightGlue's **transformer-based matching**, which uses an attention mechanism to learn correspondences between features more robustly than traditional methods like FLANN.
- **Robust to Snow and Clouds:**
  The higher number of matches (*199*) between the snow and no-snow images indicates that **LightGlue** is able to match keypoints even when the landscape is partially obscured by snow. This is because the attention mechanism in the transformer allows **LightGlue** to focus on important parts of the image that remain consistent despite environmental changes like snow cover.
- **Clouds Still Pose a Challenge:**
  While **LightGlue** performs reasonably well with *121 matches* in the cloud scenario, clouds still present a challenge since they obscure important parts of the image, reducing the number of matchable features. However, the use of transformers makes it more resilient to occlusion compared to **FLANN**.

3. ALIKED Results:

- *No Snow & No Clouds vs Snow & No Clouds*: 1019 matched keypoints
- *No Snow & No Clouds vs No Snow & Clouds*: 419 matched keypoints

Analysis:

- **Significantly Higher Matches:**
  ALIKED's performance is by far the best, with *1019 matched keypoints* in the snow scenario and *419 matches* in the cloud scenario. This substantial improvement is due to **ALIKED's learning-based keypoint detector** and descriptor generation, which are adaptive to changes in the scene.
- **Adaptability to Scene Changes:**
  Unlike the handcrafted features in **FLANN** or the pre-trained features in **LightGlue**, **ALIKED** is specifically designed to adapt to different image conditions (like snow and clouds). Its ability to detect keypoints that are invariant to scale, rotation, and environmental changes (like snow) allows it to find many more corresponding points between images that differ in snow or cloud cover.
- **Better Handling of Clouds:**
  ALIKED performs significantly better than the other methods when matching an image with clouds (*419 matches*), showing its robustness to partial occlusions like clouds. This is likely because its learned features are better at capturing underlying structures that remain consistent even when the surface appearance is altered by clouds or snow.

Conclusions

The differences in the number of matched keypoints across the methods can be attributed to their underlying feature extraction and matching strategies:

- **FLANN-Based Matcher (SIFT + KD-Tree):**
    - Struggles with significant visual changes (snow or clouds).
    - Performs poorly due to its handcrafted features, which are not as robust to scene variations.
- **LightGlue (SuperPoint/DISK + Transformer Matching):**
    - Performs better due to its transformer-based matching mechanism, which allows for better generalization and handling of changes like snow or partial occlusions (clouds).
    - Still, the fixed feature extractors like **SuperPoint** or **DISK** limit the number of matches when environmental changes are severe.
- **ALIKED (Adaptive Keypoint Detection + LightGlue):**
    - Performs the best due to its learning-based keypoint detector that is specifically designed to handle large variations in the scene, such as snow or clouds.
    - Its ability to learn and adapt to environmental changes results in many more matched keypoints, even in challenging scenarios like cloud cover or snow.