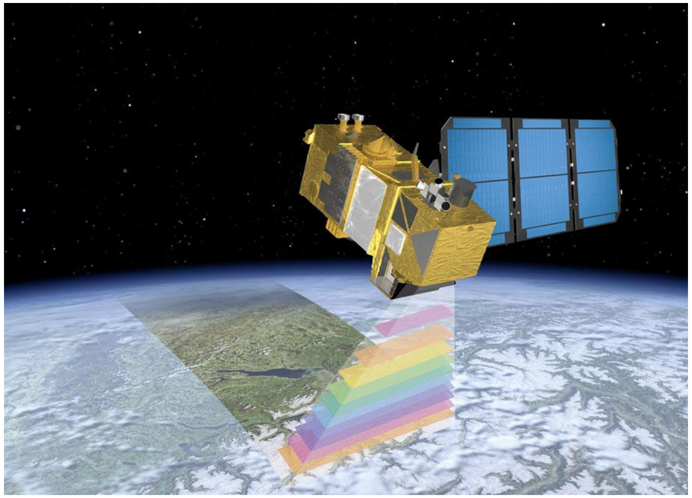
Hackathon 2019

Introduction

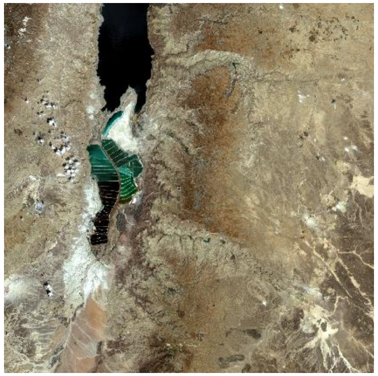
- **Sentinel-2 image time series**
- **Unsupervised learning / Weakly supervised learning**

Sentinel-2 image time series

Sentinel-2: Introduction



- Sentinel-2 mission is composed of two Earth observation satellites launched in 2015.
- It provides high-resolution optical images using multiple spectral bands from the visible and near infrared to the shortwave infrared band.
- Full, free and open data provided by Amazon Web Services and Google Cloud Platform.



Dead Sea, Israel
30/08/2015

Los Angeles, USA
19/05/2016

Landes, France
20/08/2016

Valencia, Spain
10/01/2018

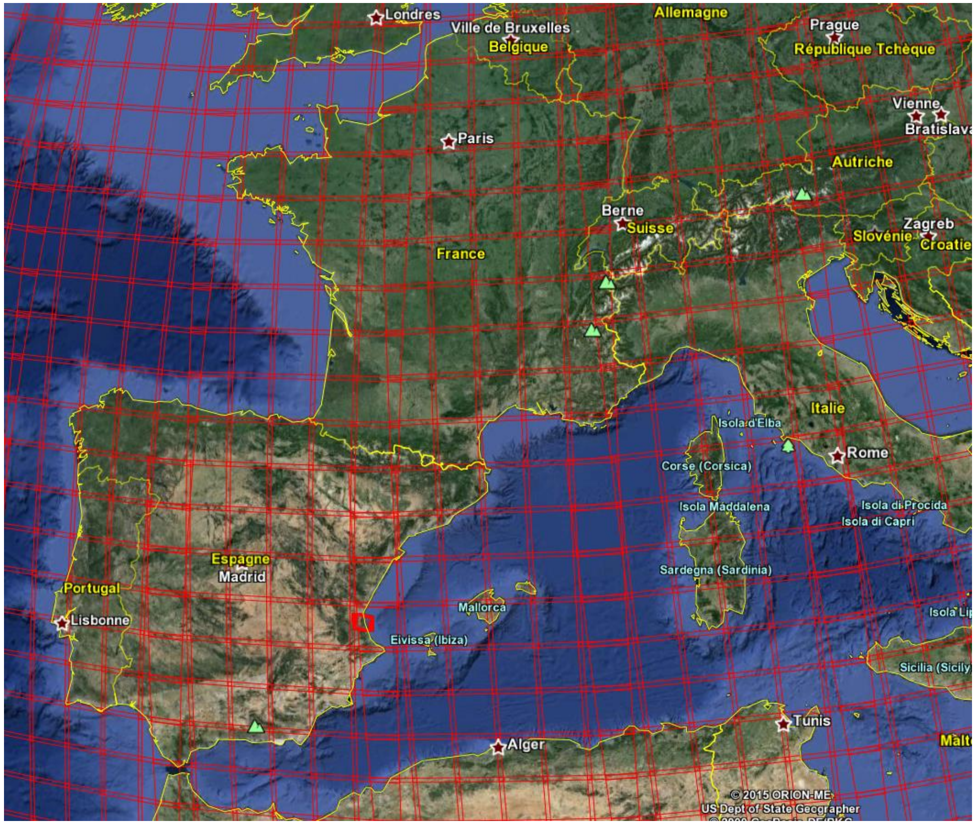
Sentinel-2: Overview

Data organization

Earth's surface is divided into square **tiles** of approximately 100 km on each side.

Each tile is identified by an ID. For instance, the tile of Toulouse is 31-T-CJ.

For the Hackathon, we selected 42 tiles containing several regions of interest such as the Amazon rainforest, the Dead Sea, the city of Los Angeles, the Great Sandy Desert, circular fields in Saudi Arabia, among others.



Sentinel-2: Overview

Temporal resolution

It is the amount of time, expressed in days, that elapses before a satellite revisits a particular point in the Earth's surface. Sentinel-2 temporal resolution is 5 days.
A series of images acquired in time order from the same geographical region is called **time series**.

Example: Time series from Southern France (tile 31-T-CJ)

07/11/2017 12/11/2017 17/11/2017 22/11/2017 27/11/2017

For the Hackathon, we provide a dataset composed of time series of length12.

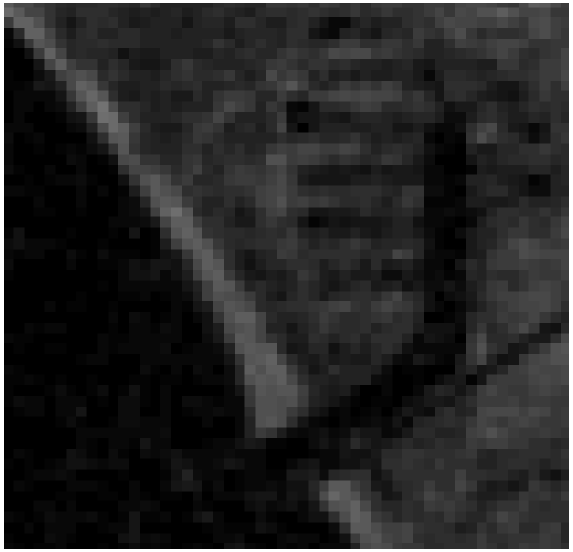
Spatial resolution

Satellite image pixels represents a certain area on Earth’s surface. There are 10 meter, 20 meter and 60 meter resolutions available.

Example: Los Angeles, USA

Band 2 (10 m resolution)

Band 5 (20 m resolution)



Band 1 (60 m resolution)

For the Hackathon, we use images at 10 meters resolution.

Sentinel-2: Overview

Spectral resolution

It stands for the number of spectral bands and their position in the electromagnetic spectrum in which the satellite can collect reflected radiance.

Band	Central wavelength (nm)	BW (nm)
2	496.6	98
3	560.0	45
4	664.5	38
8	835.1	145

10m spatial resolution bands

Band	Central wavelength (nm)	BW (nm)
5	703.9	19
6	740.2	18
7	782.5	28
8a	864.8	33
11	1613.7	143
12	2202.4	242

20m spatial resolution bands

Band	Central wavelength (nm)	BW (nm)
1	443.9	27
9	945.0	26
10	1373.5	75

60m spatial resolution bands

For the Hackathon, we use the bands 2, 3, 4 and 8 (the 10 meters resolution bands)

Sentinel-2: Concepts

Tile

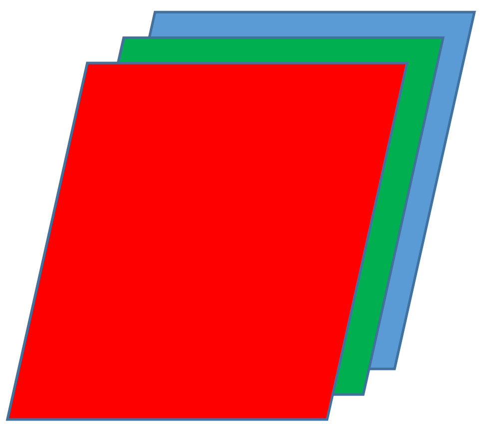
A tile is a square area of approximately 100 km x 100 km that corresponds to a given region on Earth’s surface.

Time series

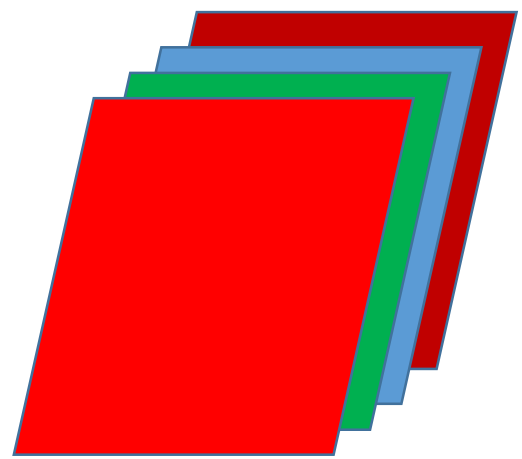
A time series is a series of images acquired in time order from the same geographical region.

Image channels/bands

A color image is composed of 3 channels (Red, Green, Blue). Similarly, a Sentinel-2 image is composed of 4 channels (Red=Band 4, Green= Band 3, Blue= Band 2 + an extra channel = Band 8)



RGB image



Sentinel-2 image

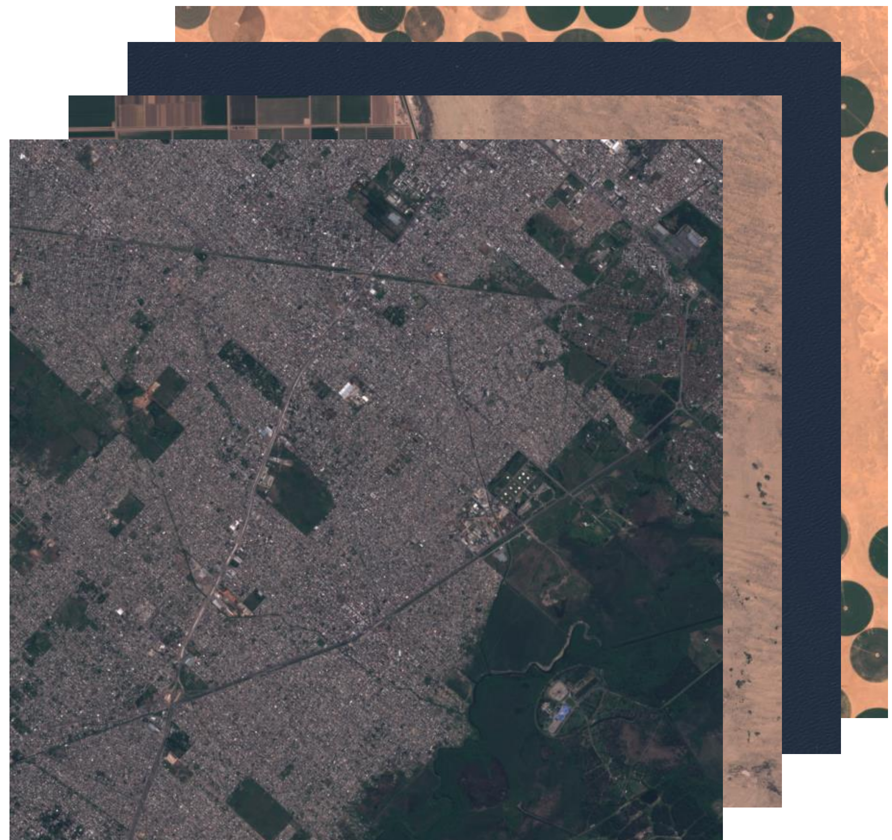
Challenge description

Challenge description

- **Objective**

Let I_{t_o} be an **image patch** acquired at time t_o and spatial coordinates (x, y). The goal is to retrieve the image that corresponds to the same spatial coordinates (x, y) on **images** acquired at time $t_f \neq t_o$.

Image patch I_{t_o} (x, y, t_o)



Images acquired at $t_f \neq t_o$

Challenge description

- **Objective**

Let I_{t_o} be an **image patch** acquired at time t_o and spatial coordinates (x, y). The goal is to retrieve the image that corresponds to the same spatial coordinates (x, y) on **images** acquired at time $t_f \neq t_o$.

- **Example:**

Image patch I_{t_o}
(x, y, t_o)

Target I_{t_f}
(x, y, t_f)

Images acquired at $t_f \neq t_o$

Challenge description

- **Objective**

Let I_{t_o} be an **image patch** acquired at time t_o and spatial coordinates (x, y). The goal is to retrieve the image that corresponds to the same spatial coordinates (x, y) on **images** acquired at time $t_f \neq t_o$.

- **Example:**

Image patch I_{t_o}
(x, y, t_o)

Target I_{t_f}
(x, y, t_f)

Images acquired at $t_f \neq t_o$

Challenge description

- **Objective**

Let I_{t_o} be an **image patch** acquired at time t_o and spatial coordinates (x, y). The goal is to retrieve the image that corresponds to the same spatial coordinates (x, y) on **images** acquired at time $t_f \neq t_o$.

- **Example:**

Image patch I_{t_o}
(x, y, t_o)

Target I_{t_f}
(x, y, t_f)

Images acquired at $t_f \neq t_o$

Challenge description

- **Objective**

Let I_{t_o} be an **image patch** acquired at time t_o and spatial coordinates (x, y). The goal is to retrieve the image that corresponds to the same spatial coordinates (x, y) on **images** acquired at time $t_f \neq t_o$.

- **Example:**

Image patch I_{t_o}
(x, y, t_o)

Target I_{t_f}
(x, y, t_f)

Images acquired at $t_f \neq t_o$

Challenge description

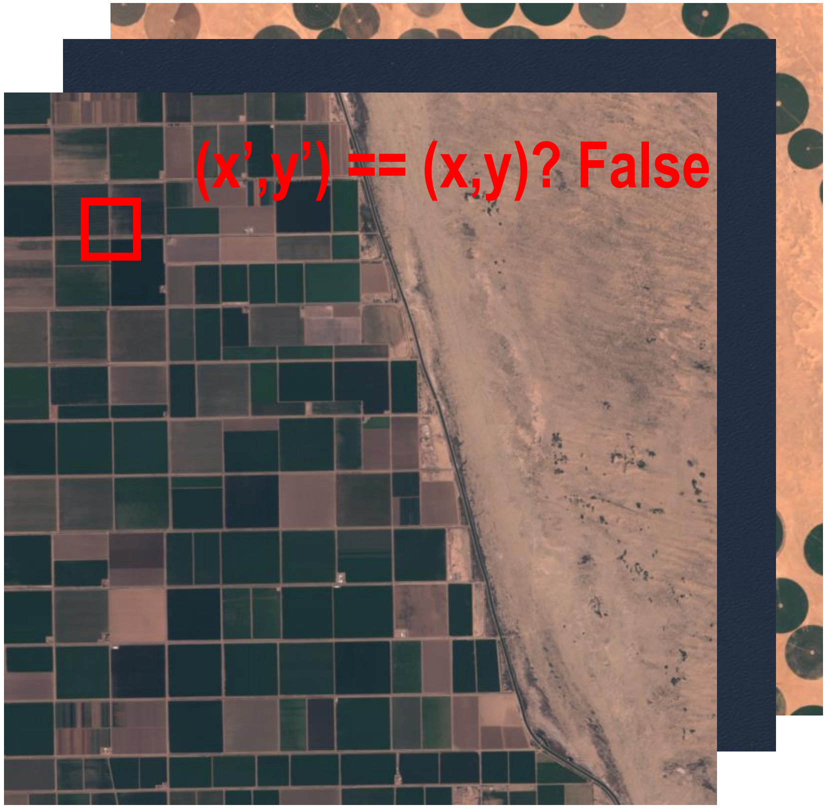
- **Objective**

Let I_{t_o} be an **image patch** acquired at time t_o and spatial coordinates (x, y). The goal is to retrieve the image that corresponds to the same spatial coordinates (x, y) on **images** acquired at time $t_f \neq t_o$.

- **Example:**

Image patch I_{t_o} (x, y, t_o)

Target I_{t_f} (x, y, t_f)



Images acquired at $t_f \neq t_o$

Challenge description

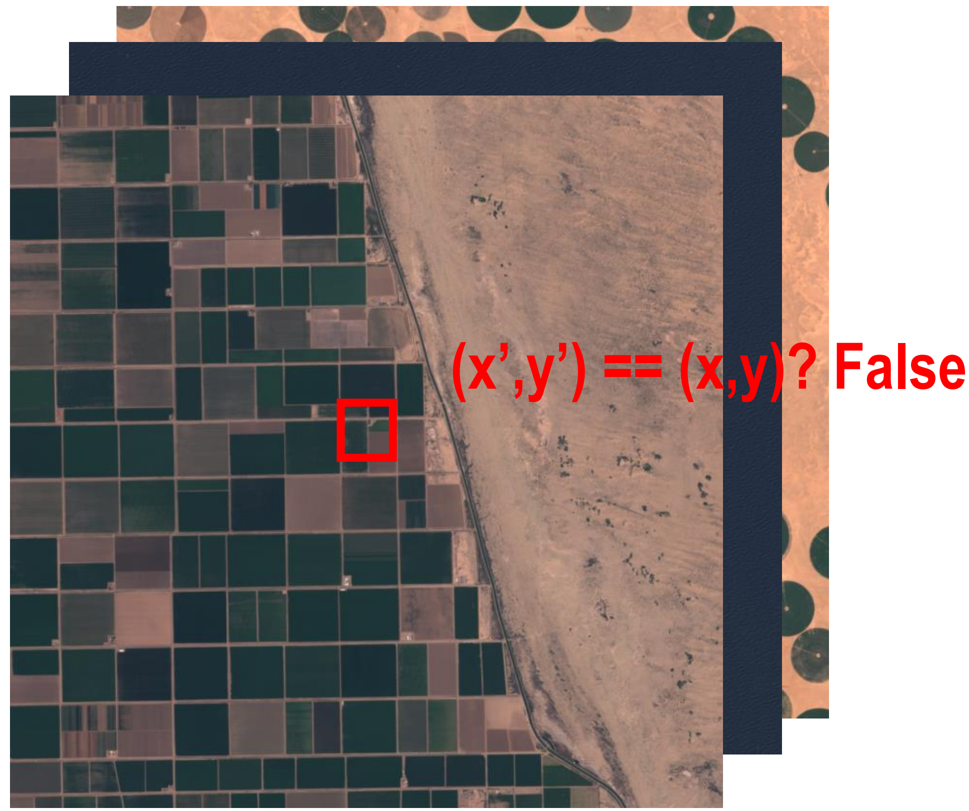
- **Objective**

Let I_{t_o} be an **image patch** acquired at time t_o and spatial coordinates (x, y). The goal is to retrieve the image that corresponds to the same spatial coordinates (x, y) on **images** acquired at time $t_f \neq t_o$.

- **Example:**

Image patch I_{t_o}
(x, y, t_o)

Target I_{t_f}
(x, y, t_f)



Images acquired at $t_f \neq t_o$

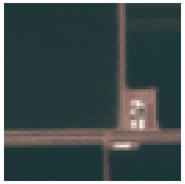
Challenge description

- **Objective**

Let I_{t_o} be an **image patch** acquired at time t_o and spatial coordinates (x, y). The goal is to retrieve the image that corresponds to the same spatial coordinates (x, y) on **images** acquired at time $t_f \neq t_o$.

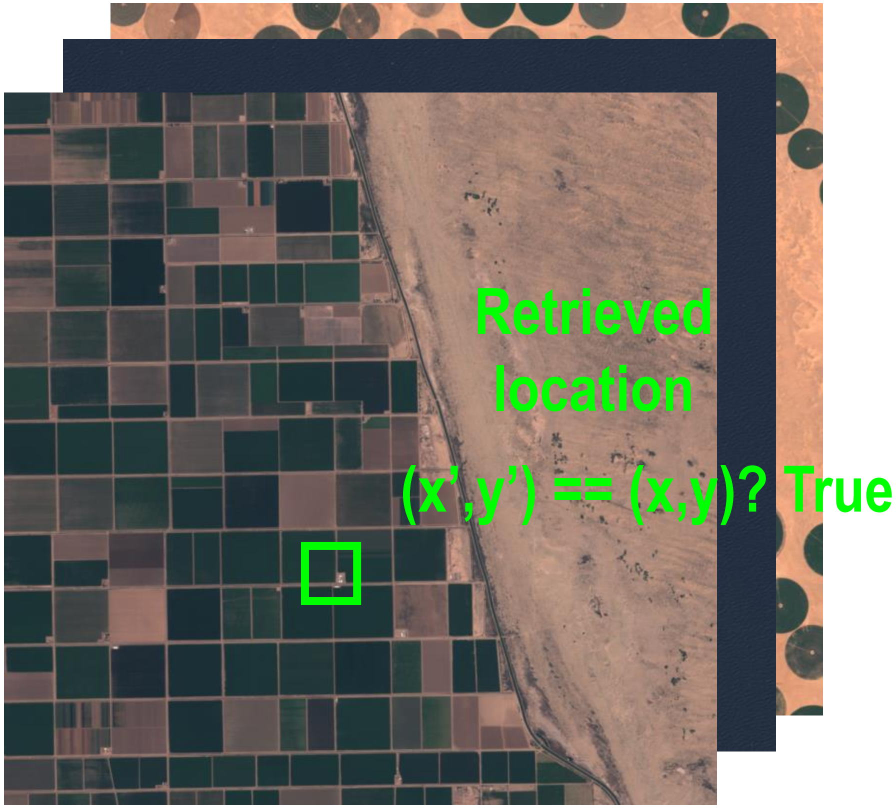
- **Example:**

Image patch I_{t_o}
(x, y, t_o)



Target I_{t_f}
(x, y, t_f)

Image retrieval algorithms must take into account possible changes! For instance, seasonal changes and new buildings.



Images acquired at $t_f \neq t_o$

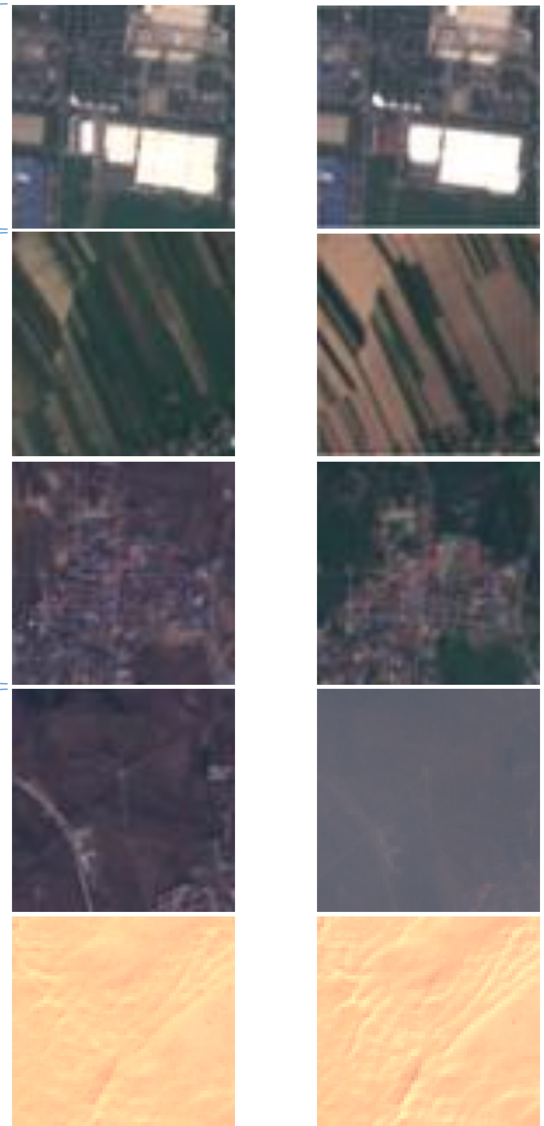
Examples

Input Target

Elementary level

Intermediate level

Advanced level



Generally, image retrieval algorithms perform well in urban scenarios since the city provides details that can be easily identified.

Image retrieval algorithms must be able to deal with seasonal changes in order to identify the image patch location.

Image retrieval algorithms must be able to deal with satellite acquisition condition (cloud coverage, view angles, light conditions).

Some image patches are notoriously difficult to locate even for humans (oceans, deserts, forests, etc.).

Challenge interest

- **Learn the common information across the image time series.**
- **Create a shared representation → exclusive representation**
- **Split the representation of image time series into a spatial and temporal representations.**
- **Develop further tasks based on these representations:**
  - **Image classification.**
  - **Image segmentation.**
  - **Image change detection.**

Challenge data

Challenge definitions

Band

A band corresponds to an image channel.

Image

An image is composed of 4 bands of size 512 x 512. It is represented by an array of size 512 x 512 x 4.

Time series

A time series is composed of 12 images from the same geographical region in time order. It is represented by an array of size 12 x 512 x 512 x 4.

Image patch

An image patch is composed of 4 bands of size 64 x 64. It is represented by an array of size 64 x 64 x 4.

Challenge data

- ## Training phase

  We provide a training dataset containing 4200 time series extracted from 42 different tiles. Each time series is composed of 12 images of size 512 x 512. Each image is composed of 4 channels: red, green, blue and near infrared. Dataset size ~ **100GB**. Format: **Tfrecord** (tf.data.Dataset functions are provided)

- ## Test phase

  In order to test the retrieval algorithms, we provide two datasets. The first dataset contains the image patches whose location we aim to retrieve. It is composed of 38400 image patches of size 64 x 64 x 4. The second dataset is composed of 600 images of size 512 x 512 x 4. It contains the target images at the same location as the image patches but at a different acquisition time.

Challenge score

Challenge submission

Submission format

The submission results must be formatted in a two column CSV file.
The first column corresponds to the Patch ID.
The second column correspond to the retrieved location of the top-left corner. The retrieved location is a tuple of 3 elements (**image_id**, **pixel_x**, **pixel_y**)

	A	B
1	Patch ID	Result
2	1	1,240,350
3	2	4,256,300
4	3	2,200,300
5	4	200,35,84
6	5	9, 333, 12

CSV submission example

Image patch

image_id = 3
image_id = 2
image_id = 1

Challenge submission

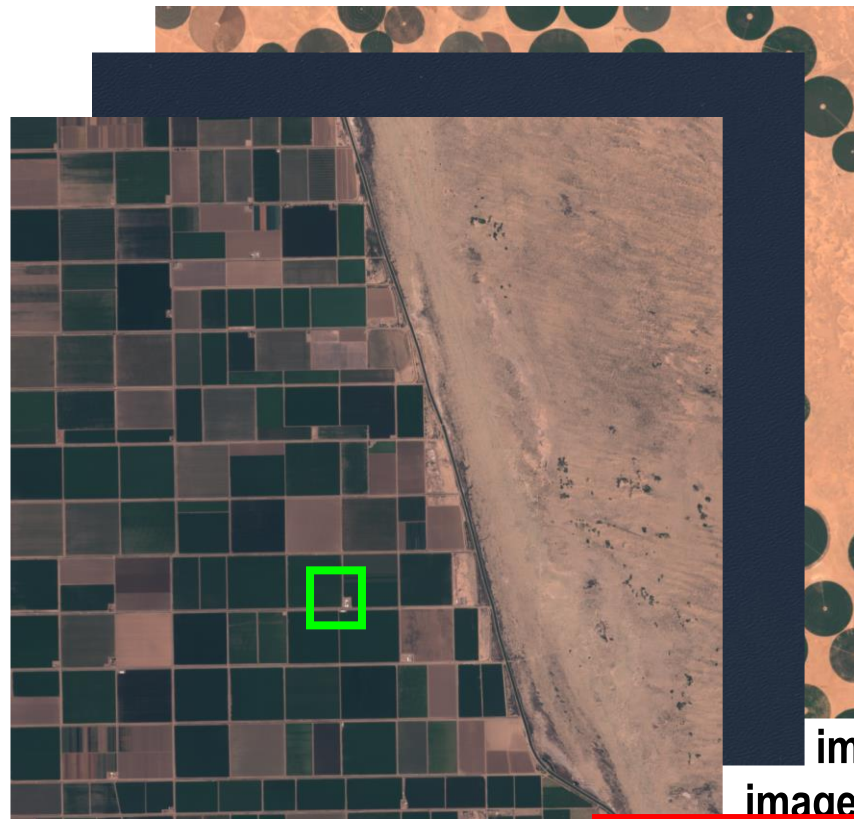
Submission format

The submission results must be formatted in a two column CSV file.
The first column corresponds to the Patch ID.
The second column correspond to the retrieved location of the top-left corner. The retrieved location is a tuple of 3 elements (**image_id**, **pixel_x**, **pixel_y**)

	A	B
1	Patch ID	Result
2	1	1,140,350
3	2	4,256,300
4	3	2,200,300
5	4	200,35,84
6	5	9, 333, 12

CSV submission example

Image patch



image_id = 3
image_id = 2
image_id = 1

Challenge submission

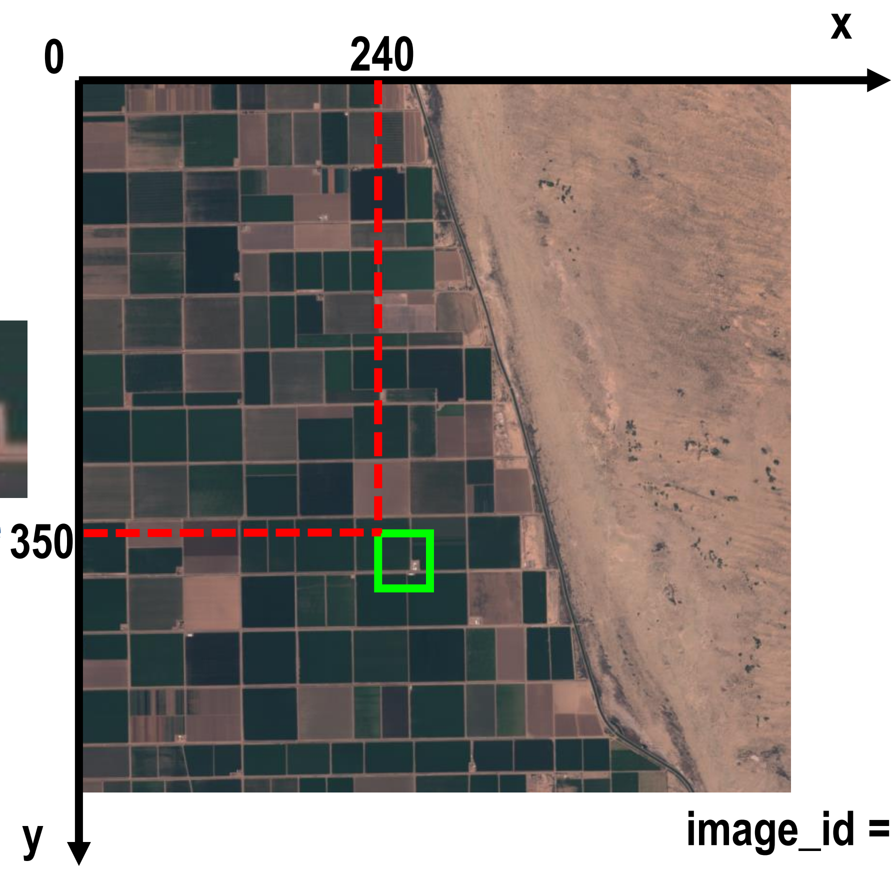
Submission format

The submission results must be formatted in a two column CSV file.
The first column corresponds to the Patch ID.
The second column correspond to the retrieved location of the top-left corner. The retrieved location is a tuple of 3 elements (**image_id**, **pixel_x**, **pixel_y**)

	A	B
1	Patch ID	Result
2	1	1,240,350
3	2	4,256,300
4	3	2,200,300
5	4	200,35,84
6	5	9, 333, 12

CSV submission example

Image patch



Score nature

The performance evaluation takes into account 2 model skills:

- The capacity to identify the image where the image patch belongs to.

$$SC_{img} = \frac{\#correctly\ retrieved\ images}{\#\ patches}$$

- The accuracy to identify the image patch location.

$$SC_{loc} = \frac{\#\ correctly\ retrieved\ locations}{\#\ patches}$$

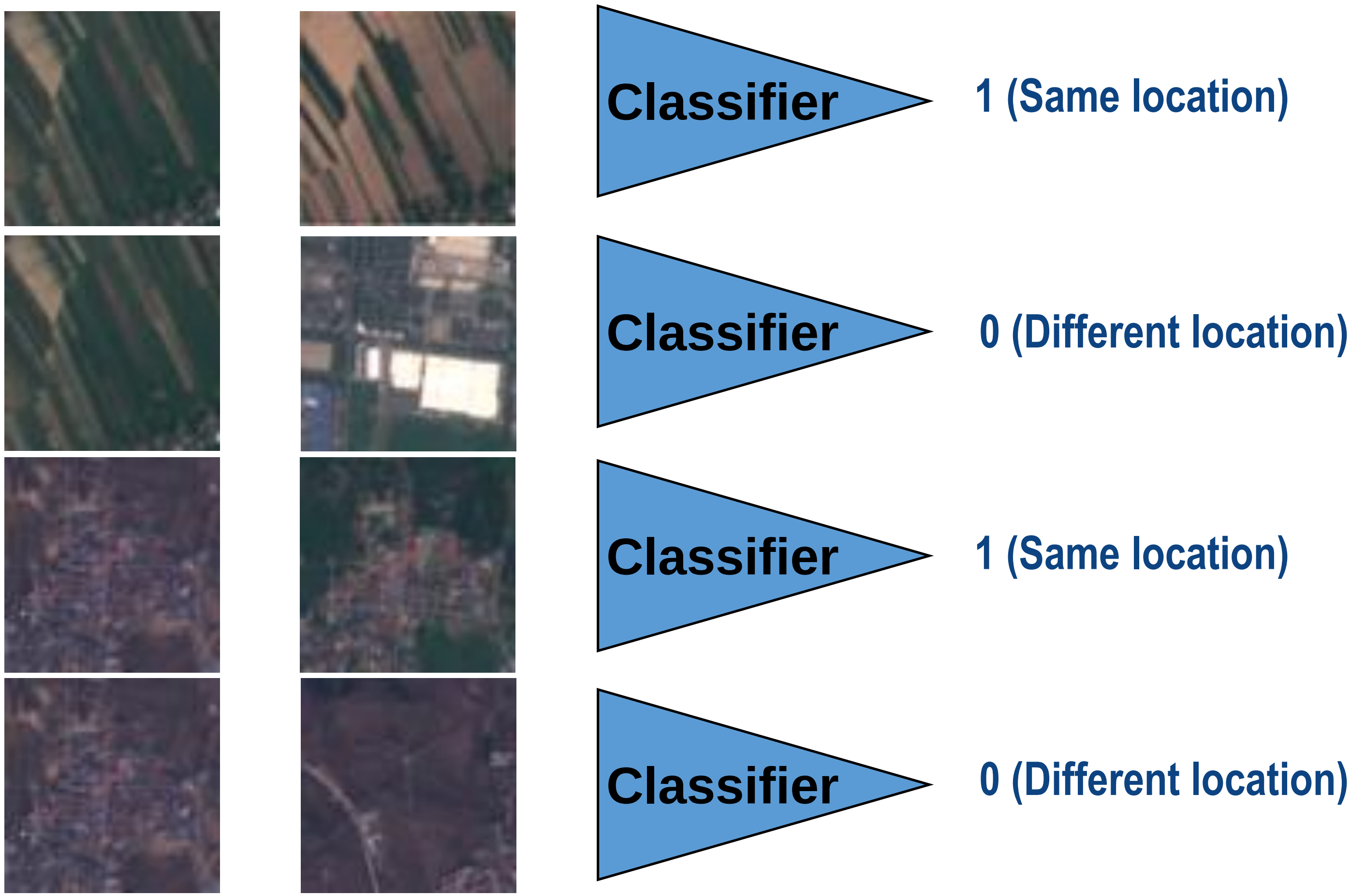
- The final score is a weighted score:

$$SC = 0.5\ * SC_{img} + 0.5\ * SC_{loc}$$

Solution ideas

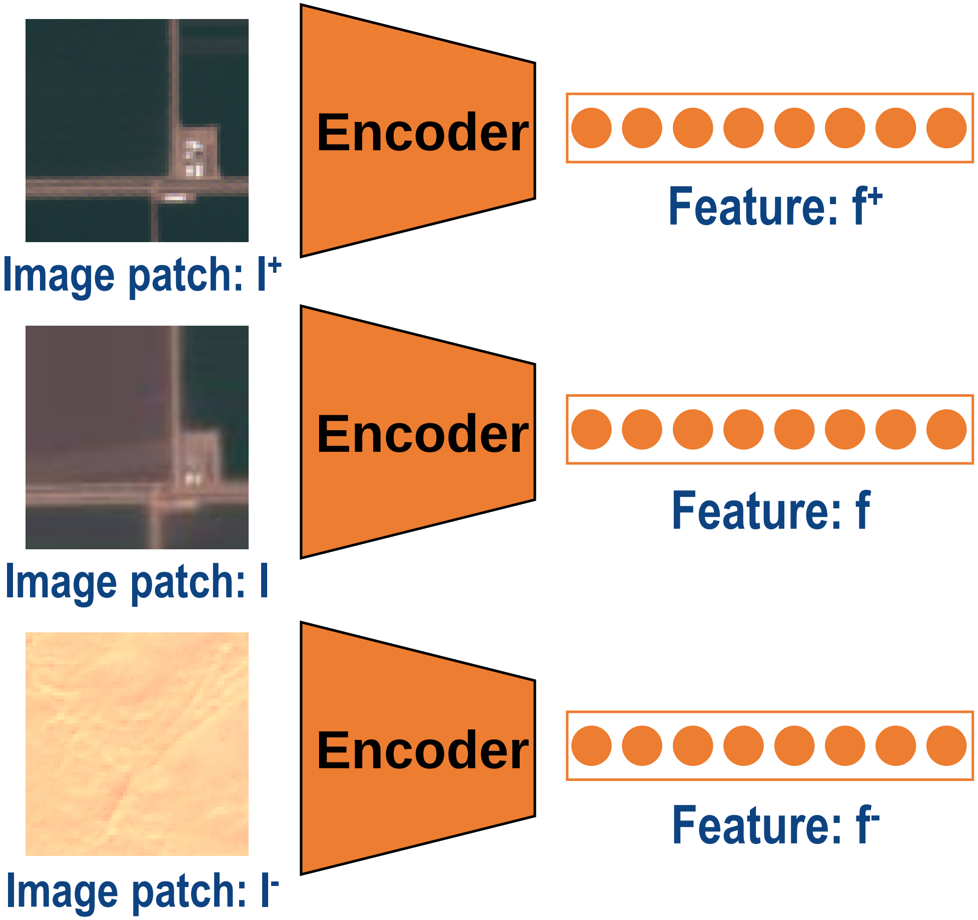
Naive classifier

Learn a classifier to decide whether two image patches come from the same location or not. Example:



Triplet loss

Learn an encoder to minimize the distance between the features from image patches at the same location and maximize the distance between the features from image patches at different location :



Objective function

$$L = || f - f^+ || - || f - f^- ||$$