

AUTOMATIC GEOREFERENCING OF LAND CONSOLIDATION MAPS WITH AI

Simon Šanca, Sjur Kristoffer Dyrkolbotn, Leiv Bjarte Mjøs, Arve Leiknes, Helge Nysæter

Organized by



Gold sponsor:



Silver sponsor:



Bronze sponsors:





Motivation:

- Imagine a cadaster with a lack of coordinate information:
 - boundaries with unknown coordinates, fictive properties,
 - lack of coordinate information and boundary marks.
- This is the case in many parts of Norway, little or no view into the historical development of land, leading to legal issues and problems between landowners.
- How to find and correct these mistakes from already available historical data?
 - including historical maps, land consolidation maps, historical measurements and records.
- **Aim:**
 - Add some coordinate information to maps, that do not have this information yet.
 - Automate cadastral mapping.

Organized by



Gold sponsor:



Silver sponsor:



Bronze sponsors:





RESEARCH PROJECT (2022 – 2026):

How can we use AI to improve the Norwegian cadastre?

- Automatic error detection and error correction with AI,
- developing automated methods to improve the cadastre.
- Research areas:
 - circle properties (sirkeleiendommer) → unknown boundary,
 - computer vision, AI, NLP,
 - cadastral document detection system,
 - automatic georeferencing with AI,
 - Voronoi polygons,
 - cost surfaces,
- Open-source software used: QGIS, GRASS, Postgres and PostGIS, Python, R, PyTorch, ...



Red – correct/known coordinates
Blue – proposed/Voronoi diagrams

Organized by



Gold sponsor:



Silver sponsor:



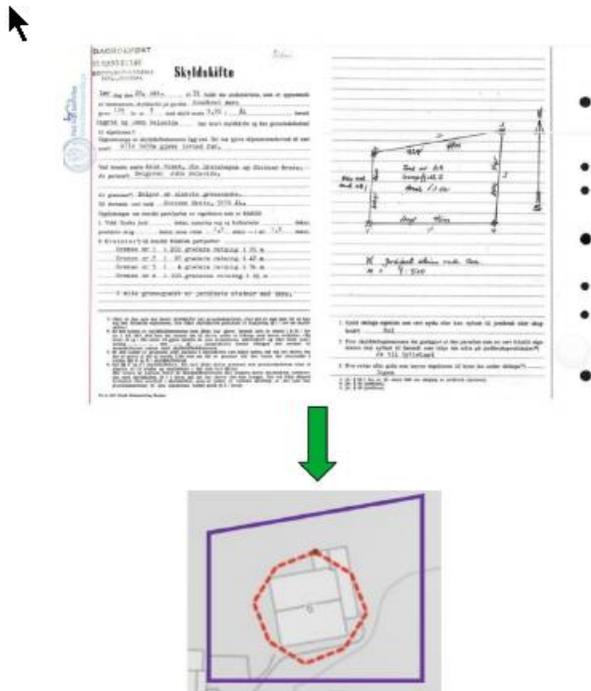
Bronze sponsors:



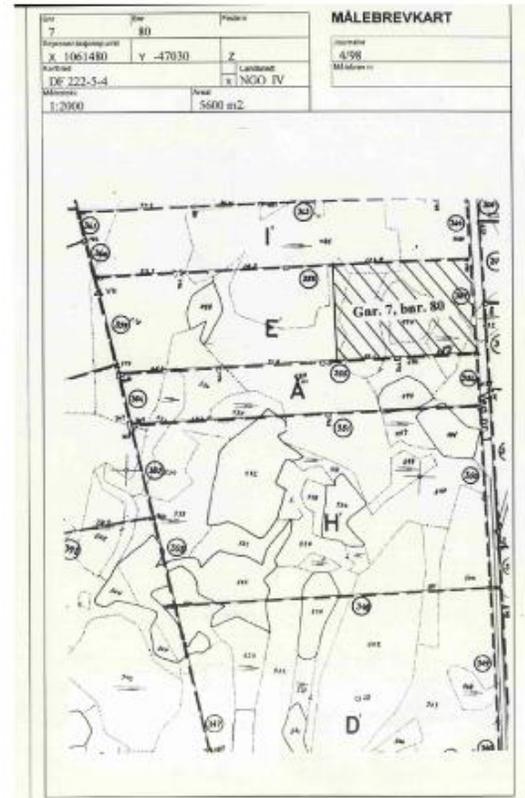


THINGS I AM TRYING AND FAILING TO DO:

Cadastral document detection system



AI-based cadastral mapping



AI-based georeferencing



Organized by



Gold sponsor:



Silver sponsor:



Bronze sponsors:





AI-based GEOREFERENCING - what/why/possible?

- 1) **Georeferencing** is a process that takes known ground control points and connects them to the raster in order to apply a coordinate system and projection to the image.
 - 2) With **artificial intelligence (AI)** we can teach algorithms to make human decisions. This enables the automation of tasks, including the detection of ground control / tie points for georeferencing.
 - 3) This is what we mean by **AI-based georeferencing**.
- **Why?** Automation, improved decision-making, cost reduction in the cadastre.
 - **Possible?** Yes, because:
 - We can detect objects on images/videos with computer vision.
 - We can analyze text, sound and time-series with AI.
 - **AIM:** Use AI to automate georeferencing, mostly focusing on the detection of tie-points, that transform the map without coordinate information into a map with defined coordinate information.

Organized by



Gold sponsor:



Silver sponsor:



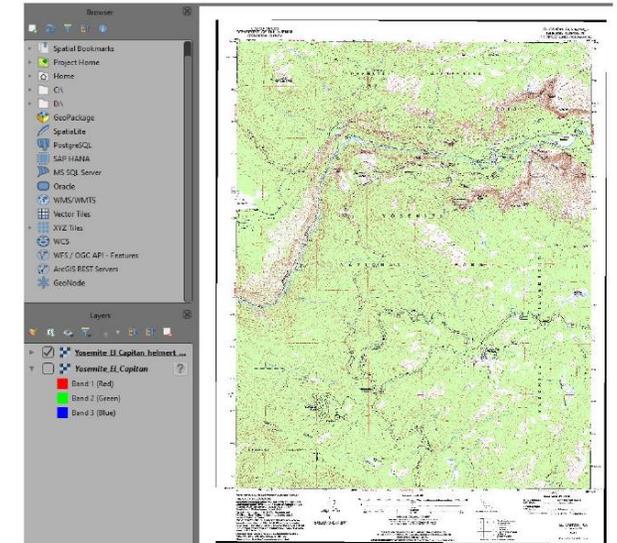
Bronze sponsors:





THE PROCESS OF GEOREFERENCING MAPS – what can be automated with AI?

- 1) Import image to be georeferenced
- 2) **Identify a series of tie-points either from coordinates (if the map has a coordinate grid) or following natural phenomena on the map.**
- 1) Use the tie-points to permanently transform the map from its existing location to the spatially correct location.
- 2) The connection between one control point on the source map and the target map is a link.
- 3) Export the georeferenced image.



Organized by



Gold sponsor:



Silver sponsor:



Bronze sponsors:

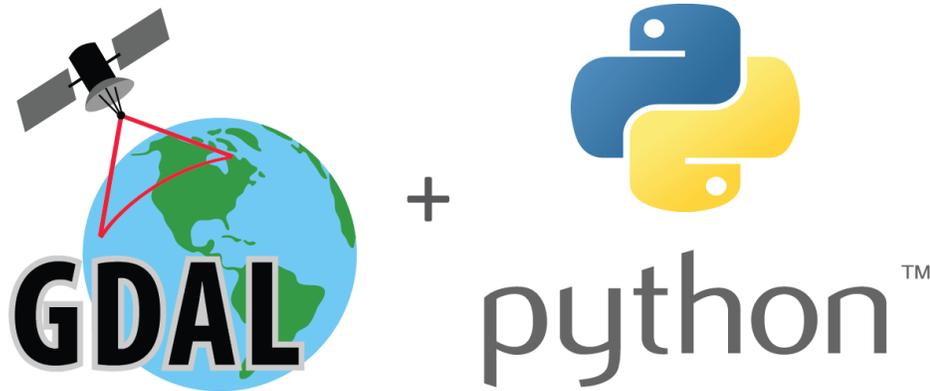




GDAL to the rescue!

Geospatial Data Abstraction Library

- <https://gdal.org/index.html>



Perfect combo: Like bread and Hagelslag.

```

1  from osgeo import gdal, ogr, osr
2
3  # Input map image and its four reference points (pixels) corresponding to real-world coordinates
4  input_map_image = 'input_map_scandinavia.jpg'
5  reference_points = [
6      {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.4194, 'lat': 5.7749},
7      {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.5294, 'lat': 5.5549},
8      {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.3694, 'lat': 5.1255},
9      {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.1294, 'lat': 5.5412},
10     {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.1194, 'lat': 5.4249},
11 ]
12
13 # Create a new GDAL dataset for the output georeferenced image
14 output_georeferenced_image = 'output_map_georef_scandinavia.tif'
15 ds = gdal.GetDriverByName('GTiff').Create(output_georeferenced_image, 400, 400, 3)
16
17 # Load the input map image
18 input_ds = gdal.Open(input_map_image)
19 ds.GetRasterBand(1).WriteArray(input_ds.GetRasterBand(1).ReadAsArray())
20 ds.GetRasterBand(2).WriteArray(input_ds.GetRasterBand(2).ReadAsArray())
21 ds.GetRasterBand(3).WriteArray(input_ds.GetRasterBand(3).ReadAsArray())
22
23 # Set the geotransformation matrix to map the reference points
24 gt = [reference_points[0]['lon'], (reference_points[1]['lon'] - reference_points[0]['lon']) / 400, 0,
25       reference_points[0]['lat'], 0, (reference_points[2]['lat'] - reference_points[0]['lat']) / 400]
26 ds.SetGeoTransform(gt)
27
28 # Set the spatial reference information (WGS84)
29 srs = osr.SpatialReference()
30 srs.ImportFromEPSG(4326)
31 ds.SetProjection(srs.ExportToWkt())
32
33 # Clean up and close the datasets
34 ds = None
35 input_ds = None
36
37 print(f"Georeferenced map saved to {output_georeferenced_image}")
    
```

Organized by



Gold sponsor:



Silver sponsor:



Bronze sponsors:



```
1 from osgeo import gdal, ogr, osr
2
3 # Input map image and its four reference points (pixels) corresponding to real-world coordinates
4 input_map_image = 'input_map_scandinavia.jpg'
5 reference_points = [
6     {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.4194, 'lat': 5.7749},
7     {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.5294, 'lat': 5.5549},
8     {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.3694, 'lat': 5.1255},
9     {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.1294, 'lat': 5.5412},
10    {'pixel_x': 100, 'pixel_y': 100, 'lon': 65.1194, 'lat': 5.4249},
11 ]
12
13 # Create a new GDAL dataset for the output georeferenced image
14 output_georeferenced_image = 'output_map_georef_scandinavia.tif'
15 ds = gdal.GetDriverByName('GTiff').Create(output_georeferenced_image, 400, 400, 3)
16
17 # Load the input map image
18 input_ds = gdal.Open(input_map_image)
19 ds.GetRasterBand(1).WriteArray(input_ds.GetRasterBand(1).ReadAsArray())
20 ds.GetRasterBand(2).WriteArray(input_ds.GetRasterBand(2).ReadAsArray())
21 ds.GetRasterBand(3).WriteArray(input_ds.GetRasterBand(3).ReadAsArray())
22
23 # Set the geotransformation matrix to map the reference points
24 gt = [reference_points[0]['lon'], (reference_points[1]['lon'] - reference_points[0]['lon']) / 400, 0,
25       reference_points[0]['lat'], 0, (reference_points[2]['lat'] - reference_points[0]['lat']) / 400]
26 ds.SetGeoTransform(gt)
27
28 # Set the spatial reference information (WGS84)
29 srs = osr.SpatialReference()
30 srs.ImportFromEPSG(4326)
31 ds.SetProjection(srs.ExportToWkt())
32
33 # Clean up and close the datasets
34 ds = None
35 input_ds = None
36
37 print(f"Georeferenced map saved to {output_georeferenced_image}")
38
```



HISTORICAL MAPS, CHALLENGES and LIMITATIONS

Georeferencing historical maps can be challenging due to many drawbacks, including:

- a) Limited metadata** – historical maps may lack detailed metadata or context regarding their creation, making it challenging to determine the map’s source, date or purpose.
- b) Map distortions and inaccuracies** – historical maps may contain distortions, errors or inaccuracies introduced during their creation.
- c) Lack of geographic reference points / coordinates** – historical maps often lack clear and recognizable geographic reference points including coordinates or known landmarks.
- d) Changes of landscape over time** – the physical landscape presented on historical maps may have undergone significant changes over time, including alterations in coastlines, rivers and urban development. Accounting these changes when georeferencing is crucial.
- e) High diversity of maps** – historical maps do not follow standardized design practices. Especially older maps use various design principles.
- f) Different measurements units** – different units to measure lengths, including walking, steps, sometimes with measuring tape.

Organized by

kadaster



UNIVERSITY
OF TWENTE.



Gold sponsor:



Silver sponsor:



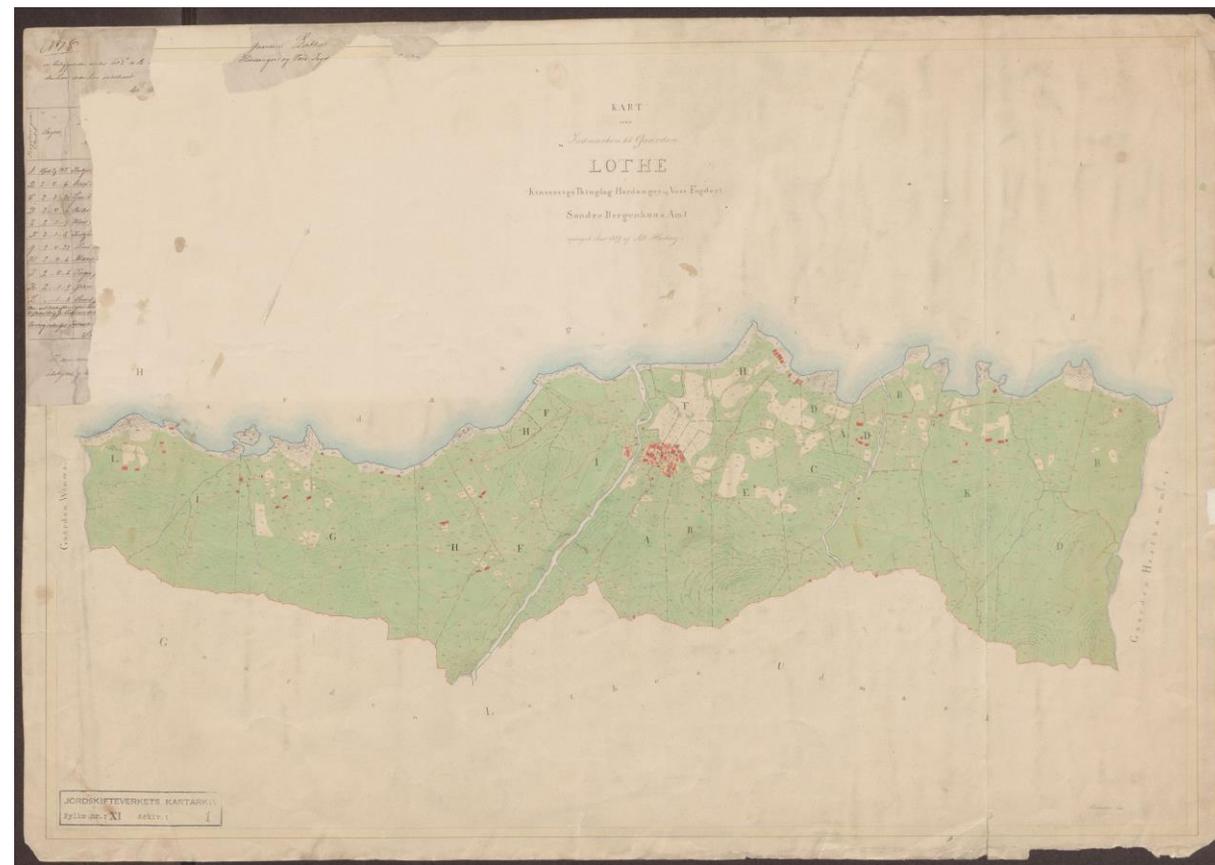
Bronze sponsors:





MAPS TO BE USED:

- <https://media.digitalarkivet.no/view/58619/1?indexing>
- around 45000 scanned maps as .JPG or .PDF



Organized by



Gold sponsor:



Silver sponsor:



Bronze sponsors:





Digital Transformation for Responsible Land Administration

FIG Commission 7 & 2 Annual Meeting 2023

2-4 October 2023, Deventer



Organized by

kadaster



UNIVERSITY OF TWENTE.



Gold sponsor:



Silver sponsor:



Bronze sponsors:





WHAT CAN BE USED AS A TIE-POINT?

- water/lakes/coastline/streams
- buildings/churches
- mountains
- Automate the detection of tie-points with AI.
- Semantic segmentation.
- Perform the georeferencing with GDAL and Python.



Organized by



Gold sponsor:



Silver sponsor:



Bronze sponsors:

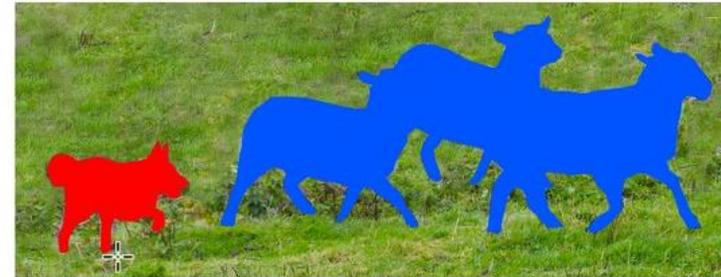




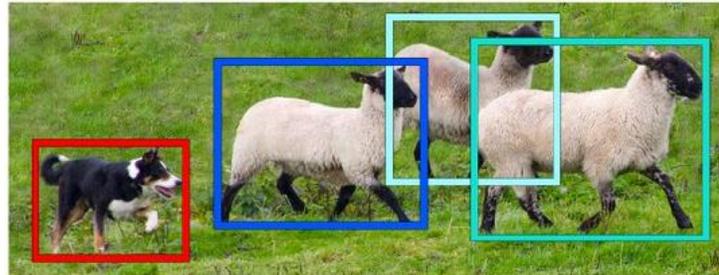
COMPUTER VISION – ability to recognize images



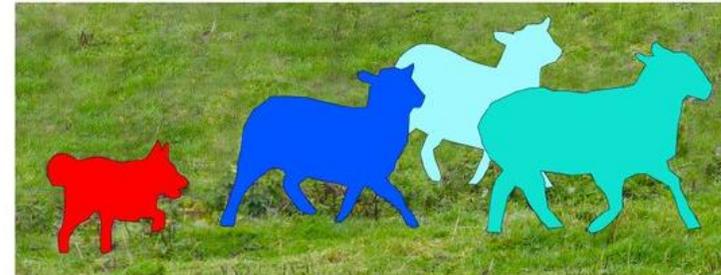
Image Recognition



Semantic Segmentation



Object Detection



Instance Segmentation

<https://desupervised.io/computer-vision>

Organized by



Gold sponsor:



Silver sponsor:



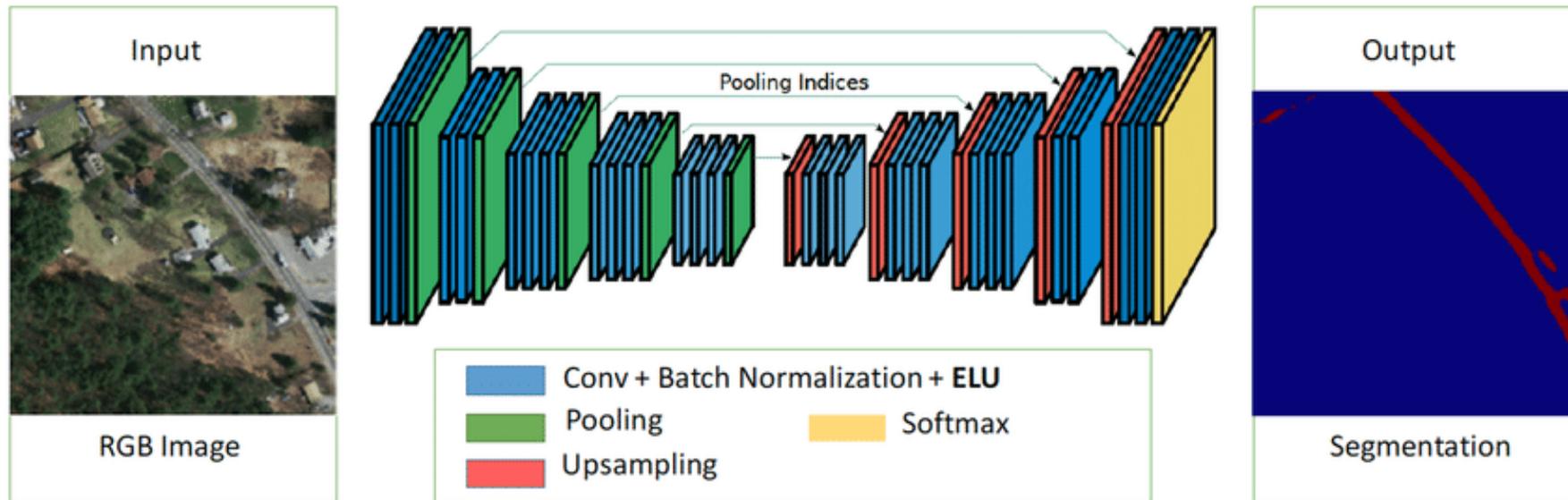
Bronze sponsors:





SEMANTIC SEGMENTATION to automatically detect the tie-points.

- Computer vision task in which the goal is to categorize each pixel in an image into a class or object.
- Output: dense pixel-wise segmentation map of an image, where each pixel is assigned to a specific class or object.



Organized by



Gold sponsor:



Silver sponsor:

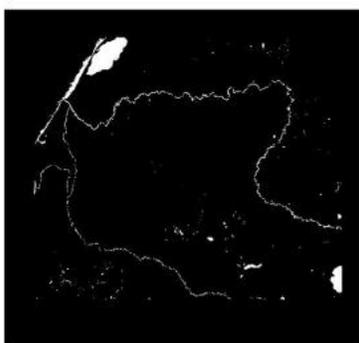
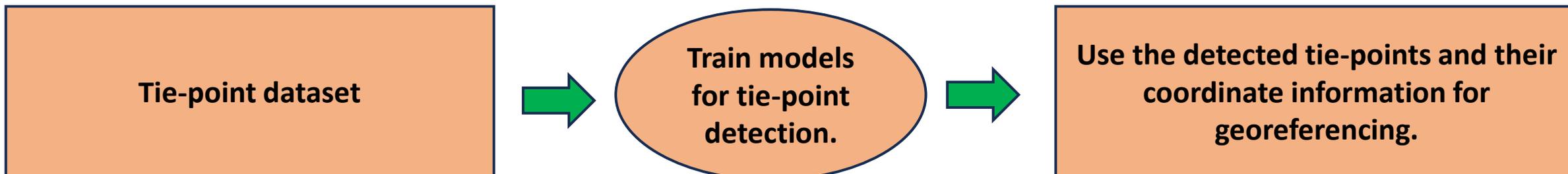


Bronze sponsors:



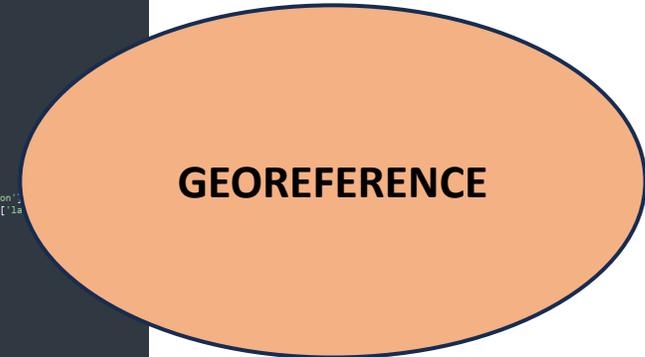


METHODOLOGY:



```

1 from osgeo import gdal, ogr, osr
2
3 # Input map image and its four reference points (pixels) corresponding to real-world coordinates
4 input_map_image = 'input_map_scandinavia.jpg'
5 reference_points = [
6     { 'pixel_x': 100, 'pixel_y': 100, 'lon': 65.4194, 'lat': 5.7749},
7     { 'pixel_x': 100, 'pixel_y': 100, 'lon': 65.5294, 'lat': 5.5549},
8     { 'pixel_x': 100, 'pixel_y': 100, 'lon': 65.3694, 'lat': 5.1255},
9     { 'pixel_x': 100, 'pixel_y': 100, 'lon': 65.1294, 'lat': 5.5412},
10    { 'pixel_x': 100, 'pixel_y': 100, 'lon': 65.1194, 'lat': 5.4240},
11 ]
12
13 # Create a new GDAL dataset for the output georeferenced image
14 output_georeferenced_image = 'output_map_georef_scandinavia.tif'
15 ds = gdal.GetDriverByName('GTiff').Create(output_georeferenced_image, 400, 400, 3)
16
17 # Load the input map image
18 input_ds = gdal.Open(input_map_image)
19 ds.GetRasterBand(1).WriteArray(input_ds.GetRasterBand(1).ReadAsArray())
20 ds.GetRasterBand(2).WriteArray(input_ds.GetRasterBand(2).ReadAsArray())
21 ds.GetRasterBand(3).WriteArray(input_ds.GetRasterBand(3).ReadAsArray())
22
23 # Set the geotransformation matrix to map the reference points
24 gt = [reference_points[0]['lon'], (reference_points[1]['lon'] - reference_points[0]['lon'],
25     reference_points[0]['lat'], 0, (reference_points[2]['lat'] - reference_points[0]['lat']
26     ds.SetGeoTransform(gt)
27
28 # Set the spatial reference information (WGS84)
29 srs = osr.SpatialReference()
30 srs.ImportFromEPSG(4326)
31 ds.SetProjection(srs.ExportToWkt())
32
33 # Clean up and close the datasets
34 ds = None
35 input_ds = None
36
37 print(f"Georeferenced map saved to {output_georeferenced_image}")
  
```



Organized by



Gold sponsor:



Silver sponsor:



Bronze sponsors:





Kort maar krachtig.



Organized by

kadaster



UNIVERSITY
OF TWENTE.



Gold sponsor:



Silver sponsor:



Bronze sponsors:

