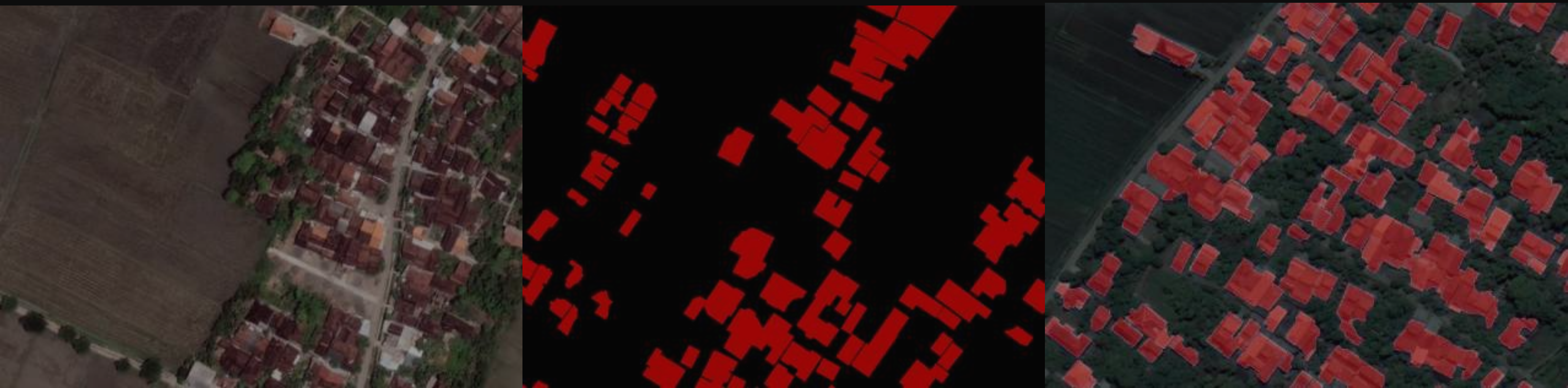
Developing Property and Tax Registration Using AI Algorithms:

Automated Mapping of The Municipality of Kendal, Indonesia



The Presenters

Konstantinos Christopoulos

Project Leader, Digireg Netherlands BV

Konstantinos.christopoulos@digireg.nl

Jati Pratomo

Country Director, Digireg Indonesia

Jati.pratomo@digireg.id

Outline

1. Background: Challenges in Tax Registrations
2. Methods
3. Result and Future Development

1

Background

Context: Property Tax in Indonesia

- **How be property tax calculated?**

(Size of the land * ZNT * tax rate) + (size of building * ZNT * tax rate)

- Self assessment – no geometric accuracy and actual size is questionable
- Not really relate to the land parcel ownership → 1 tax object can have various land ownership, vice versa

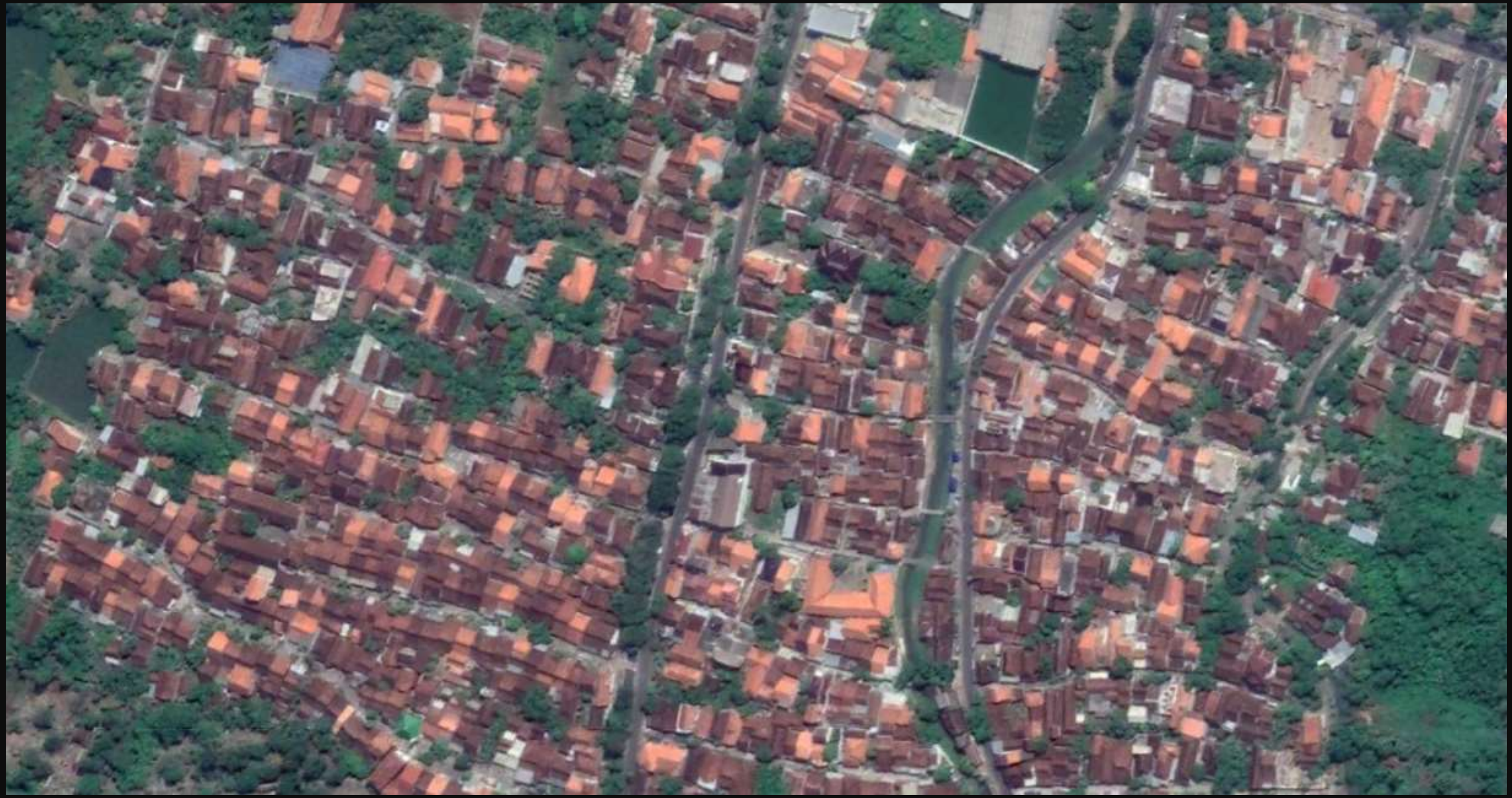
- Self assessment – no geometric accuracy and actual size is questionable
- Not always updated if there is new building or if they are extended (inc. extra floor)

ZNT – Land Price Zone

Aims of the project:

1. How to update the tax data without doing all manual building drawing?
2. How to integrate the current tax data – where it is built based on self assessment and has very low accuracy on the geometry?

Challenge: Information Extractions



Typical Settlement in Kendal region

- Built-up area is built naturally and often difficult to extract building as a single object.

- Manual building delineations require times and may resulted in a bias on the boundary based on the operators’ perceptions

Case Study

Municipality of Kendal, Central Java Province, Indonesia

Land Area: Approx. 1,000 sq. km
Population: 1.01 million (Statistic Bureau - 2020)
Built-up ratio approx. 10%
Image used: Worldview-3 (0.3 m resolution)

Characteristics:

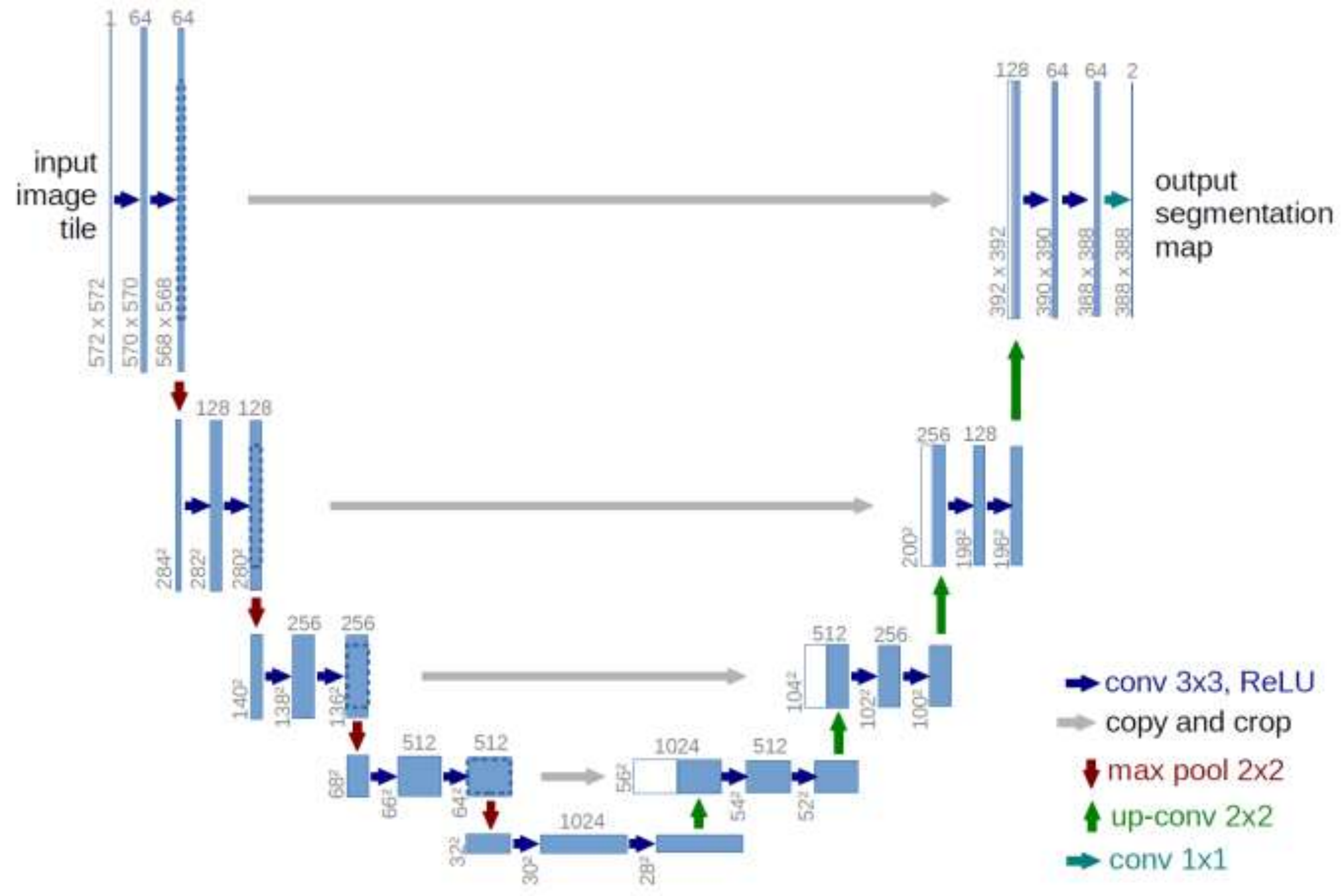
- Predominately by agriculture
- Located nearby the provincial capital (Semarang) Industrial activity has arose in the northern part
- Passed by trans-java highway which create a lot of local economic activity
- Built-up area growth rapidly but the availability of the spatial data cannot cope with its fast changes

2

Methods

Method Research

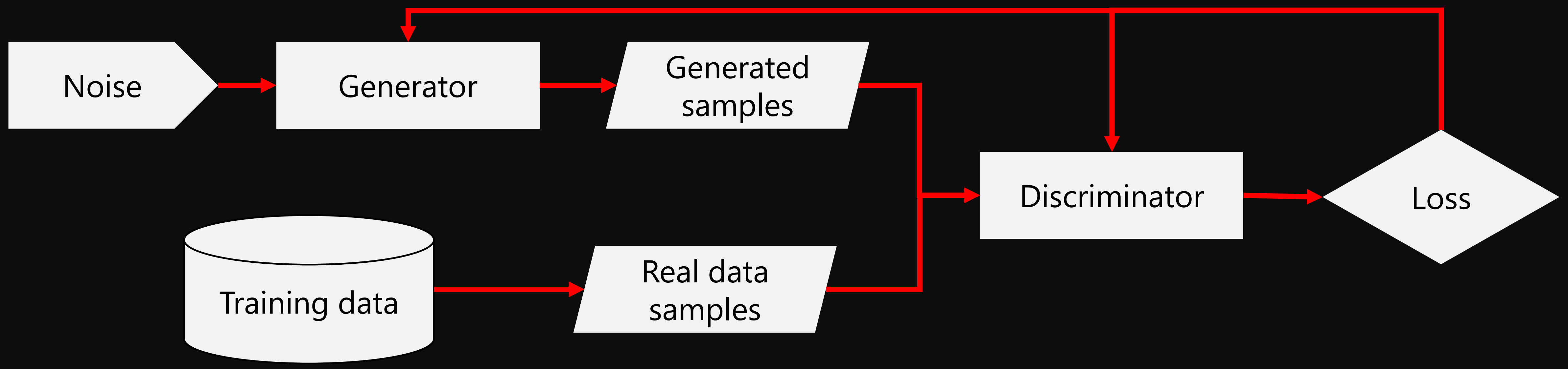
- Mask R-CNN
- U-Net
- GAN



Source: **U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015 (Ronneberger**, O;, **Fischer**, P;, B**rox, T.)**

Proposed Methods: GANs

- Generative Adversarial Networks (GANs)
- Yann Le Cun described GANs as “the most interesting idea in the last 10 years in Machine Learning”.



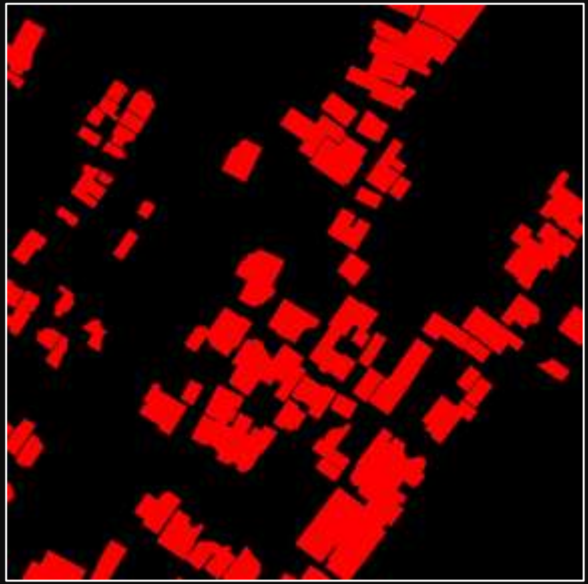
Workflow

1. Pre-processing
2. Training
3. Classification
4. Post-processing

Pre-processing

- Select training area
- Digitize '*Building*' / '*Non-Building*' classes

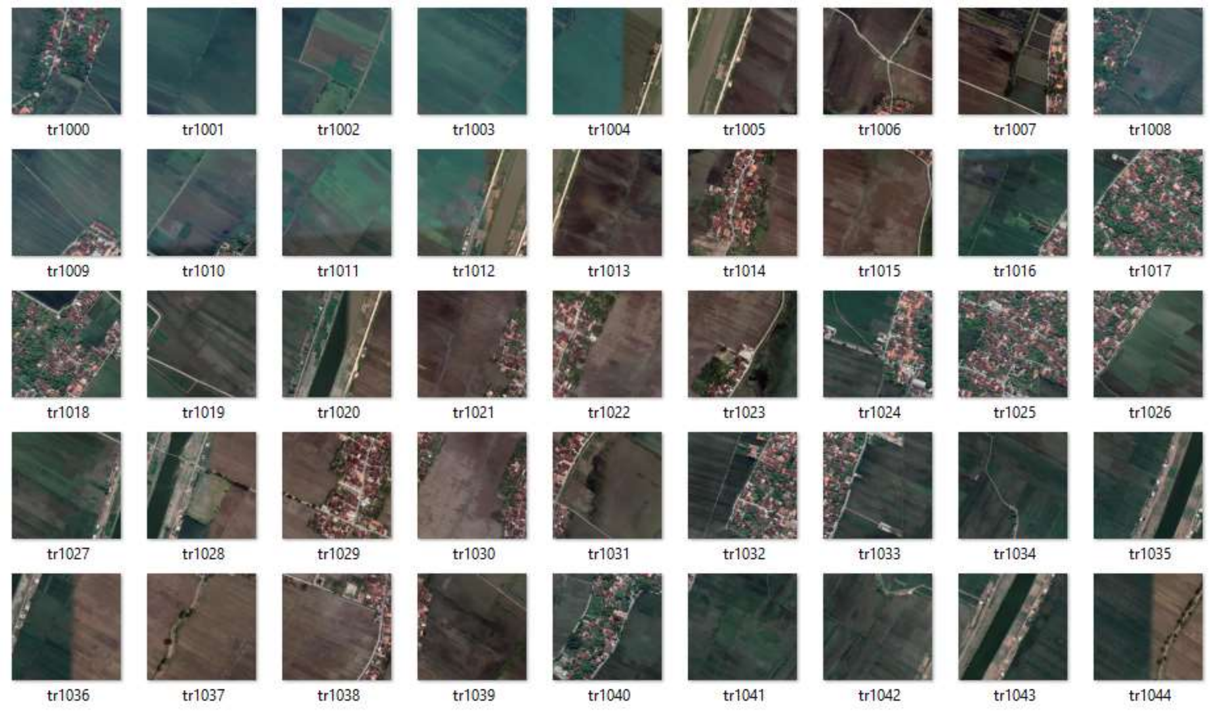
Satellite Image

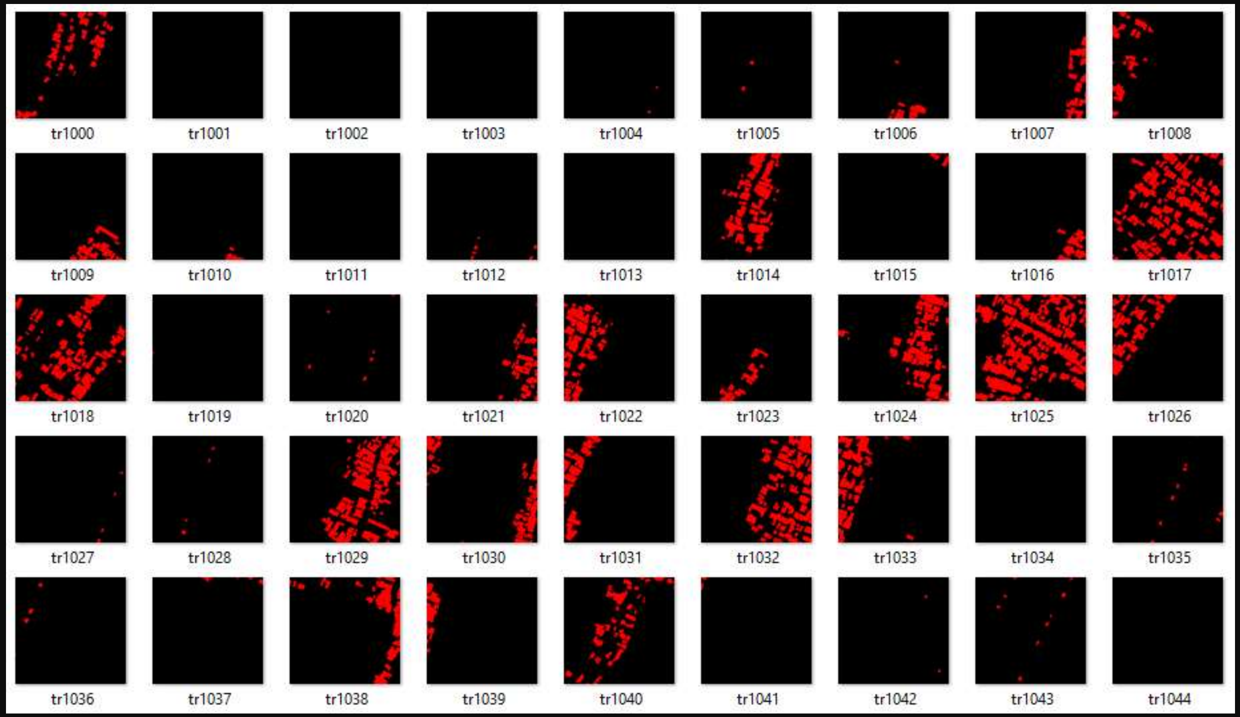


Label

Pre-processing

- Rasterize training data (image + label tiles)
- Attach a colour table to the one-band TIFF
- Convert to RGB .jpeg file





Pre-processing & Training

- Resize the satellite image and the label image to 1024 x 1024 pixels tiles
- All the processes are automated by Common geospatial libraries GDAL and OGR
- 50 training sample tiles in total
- Train model – 150 epochs used

```
def rasterizeBGT(fn_vec, raster_folder, ref_folder, label_folder):
    i = 0
    files_to_process = glob.glob(os.path.join(raster_folder, '*.tif')
    for data_path in files_to_process:
        ras_ds = gdal.Open(data_path, gdal.GA_ReadOnly)
        vec_ds = ogr.Open(fn_vec)
        lyr = vec_ds.GetLayer()
        geot = ras_ds.GetGeoTransform()
        sr = ras_ds.GetProjection()

        ref =ref_folder + '/ref' + str(i) + '.tif'
        drv_tiff = gdal.GetDriverByName("GTiff")
        chn_ras_ds = drv_tiff.Create(ref, ras_ds.RasterXSize, ras_ds.
        chn_ras_ds.SetGeoTransform(geot)
        chn_ras_ds.SetProjection(sr)

        gdal.RasterizeLayer(chn_ras_ds, [1], lyr, options=['ATTRIBUTE
        chn_ras_ds.GetRasterBand(1).SetNoDataValue(0)
        chn_ras_ds = None
```

```
original_ds = gdal.Open(src_path, 0)
sr = original_ds.GetProjection()
gt = original_ds.GetGeoTransform()
del original_ds
```

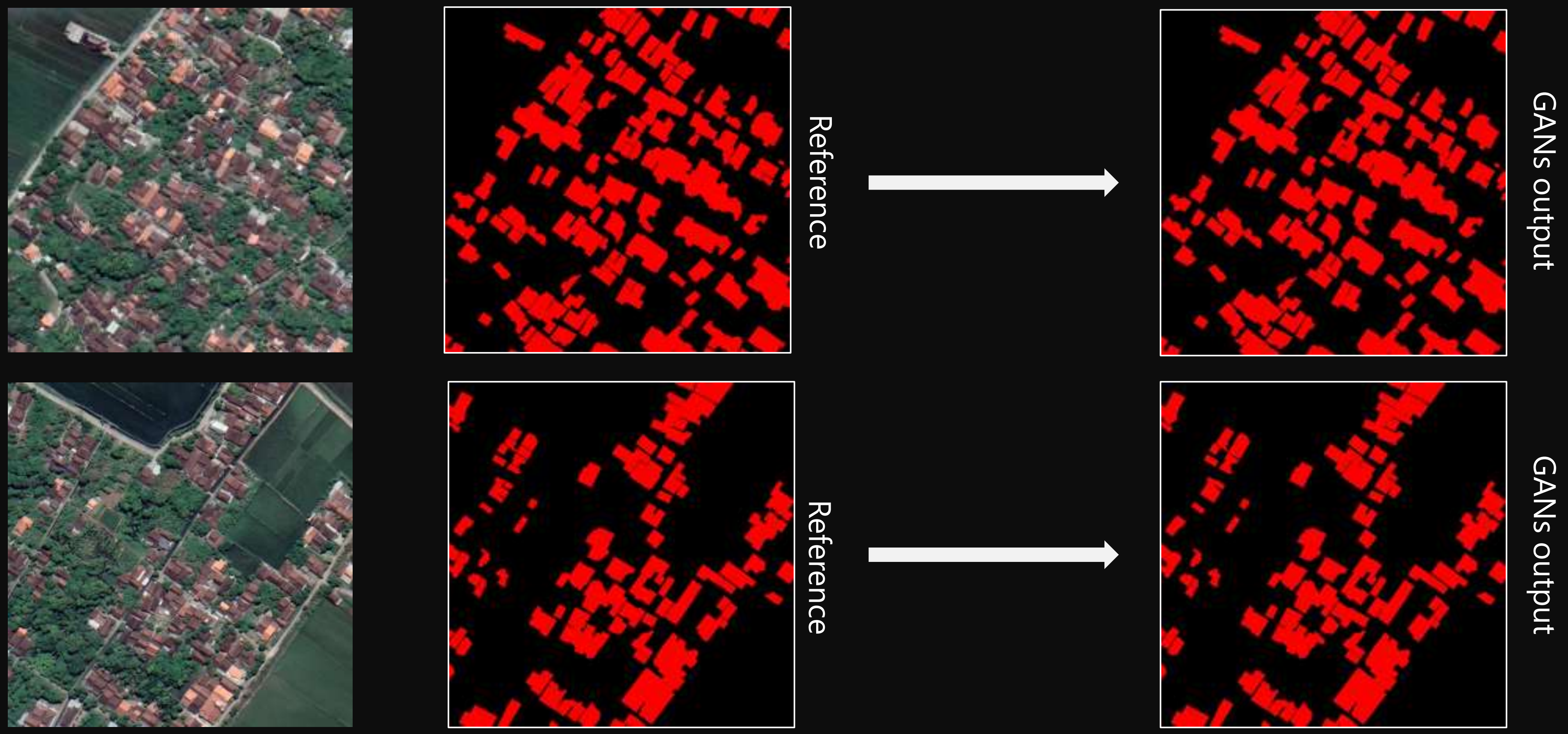

```
def rasterizeBGT(fn_vec, raster_folder, ref_folder, label_folder):
    i = 0
    files_to_process = glob.glob(os.path.join(raster_folder, '*.tif')
    for data_path in files_to_process:
        ras_ds = gdal.Open(data_path, gdal.GA_ReadOnly)
        vec_ds = ogr.Open(fn_vec)
        lyr = vec_ds.GetLayer()
        geot = ras_ds.GetGeoTransform()
        sr = ras_ds.GetProjection()

        ref =ref_folder + '/ref' + str(i) + '.tif'
        drv_tiff = gdal.GetDriverByName("GTiff")
        chn_ras_ds = drv_tiff.Create(ref, ras_ds.RasterXSize, ras_ds.
        chn_ras_ds.SetGeoTransform(geot)
        chn_ras_ds.SetProjection(sr)

        gdal.RasterizeLayer(chn_ras_ds, [1], lyr, options=['ATTRIBUTE
        chn_ras_ds.GetRasterBand(1).SetNoDataValue(0)
        chn_ras_ds = None
```

3

Result & Future Development

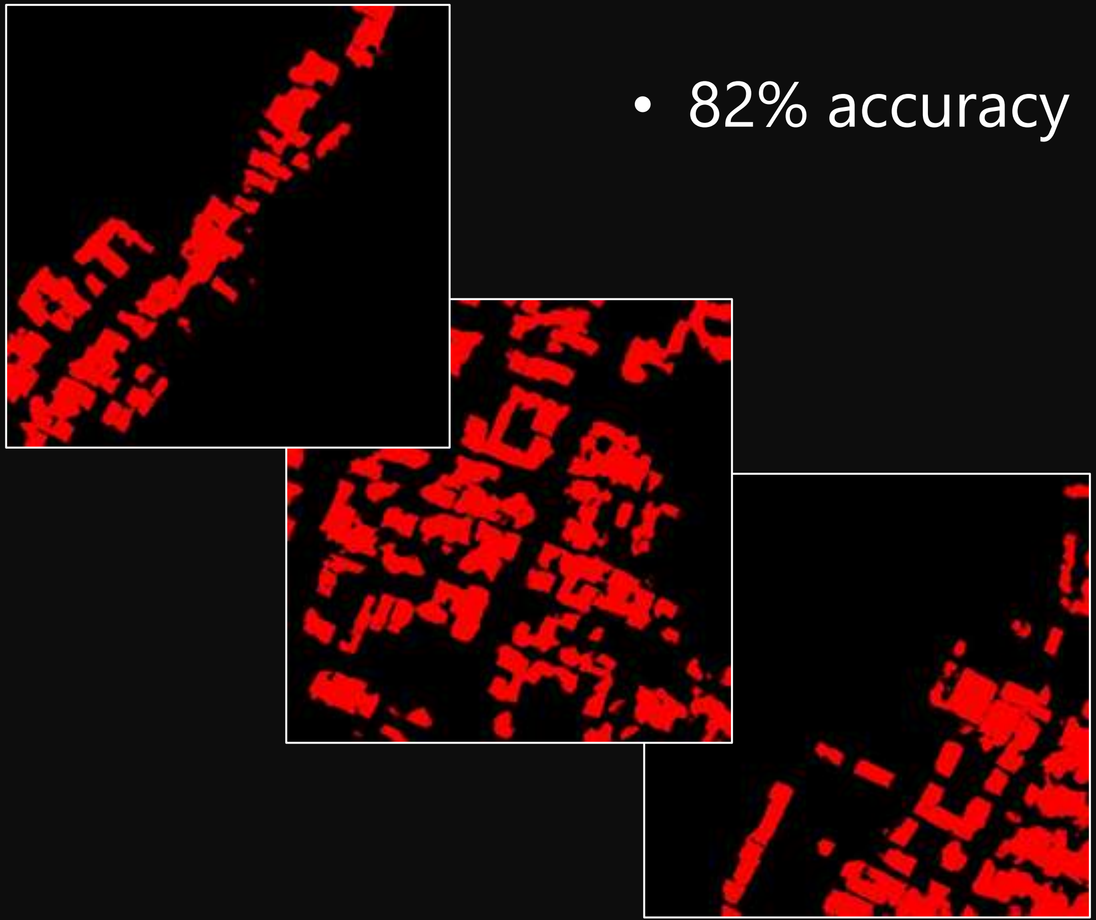
Preliminary Output

- Classification of *same* area



Preliminary Output

- Classification of *different* area



- 82% accuracy

Further Steps

- Accuracy assessment
- Data post-processing (cleaning, correct geometry)
- Multi-signalering
- Quality control
- Implement to Taxation Process

Conclusions

- Tax objects need to be updated annually; nonetheless manual detection is costly.
- GANs performs very well in a region such as Kendal. Using more training data, we expect to increase the accuracy and better building shape.
- GANs is 2x cheaper and 4x faster than manual detection, which can improve budget efficiency as well as tax object data collection.
- By using GANS, Municipality can save approx. 30,000 EUR/year cost for updating the data.
- Having an updated data may increase up to 26% of property tax in our study area.

Netherlands

Schoolstraat 7

5961 EE Horst

T +31 (0)77 - 208 6012

Email: info@digireg.nl

https://digireg.nl

Indonesia

Jl RS Fatmawati 3A

Jakarta 12420

T +62 (0)21 - 2782 9000

Email: info@digireg.id

https://digireg.id

Kenya

11 Lenana Road

P.O. Box 28896-00100,

Nairobi

T +254 (0) 703 779 751

Email:

information@digireg.co.ke