Lecture 11: GeoAI

Marina Georgati

Apr 17, 2023

IT UNIVERSITY OF COPENHAGEN

Objectives & agenda

- Introduction to GeoAI and Machine Learning
- Concepts
- Tools in Python
- Hands-on exercises

What is Geospatial Artificial Intelligence (GeoAI)?

GeoAI is the integration of geospatial studies and AI, especially machine learning and deep learning methods and the latest AI technologies in both academia and industry

Subject

- Development of intelligent computer programs to mimic the processes of human perception, spatial reasoning, and discovery about geographical phenomena and dynamics;
- To advance our knowledge;
- To solve problems in human environmental systems and their interactions, with a focus on spatial contexts and roots in geography or geographic information science (GIScience).
- Requires the knowledge of AI theory, programming and computation practices as well as geographic domain knowledge

---

Gao, S. (2021). Geospatial Artificial Intelligence (GeoAI). https://doi.org/10.1093/OBO/9780199874002-0228

Terminology I - GIScience

Geographic Information Science or Systems?

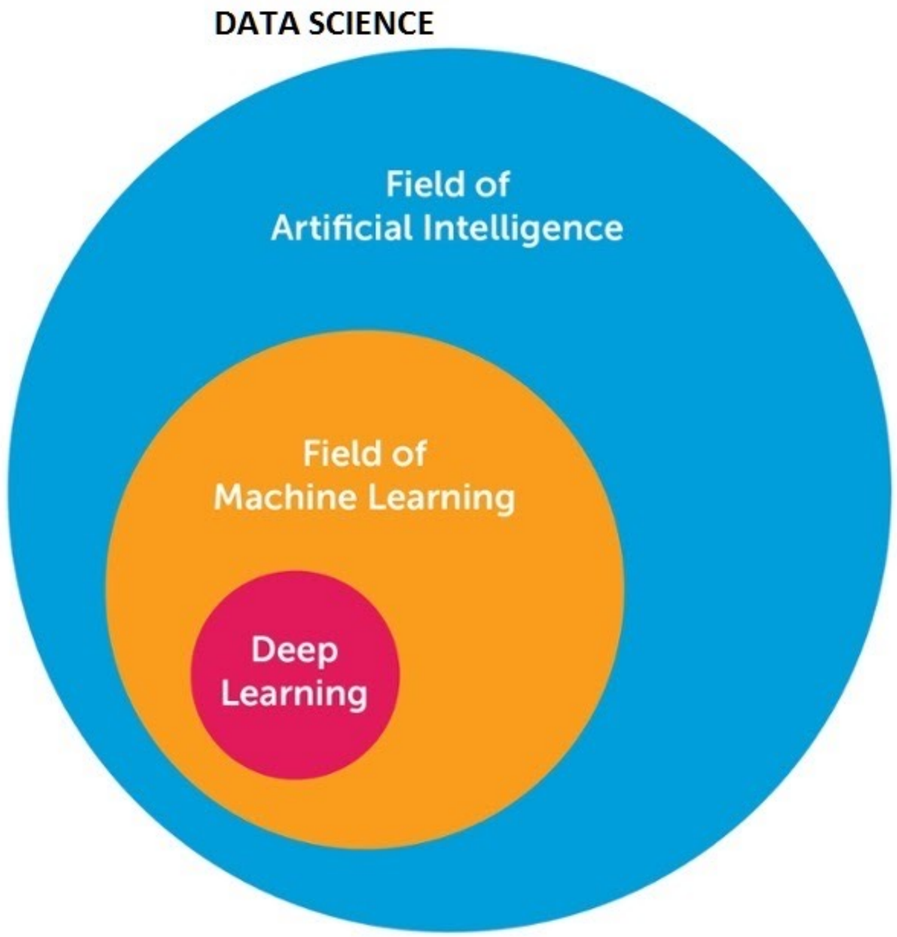
Geographic Information Systems is a computer-based tool that analyses, stores, manipulates and visualises geographic information, usually in a map.
In short, GIS is the analysis, storage, visualization and management of geographic data.

But there is another ‘S’.

This ‘S’ stands for Geographic Information Science or simply GIScience, which involves more of the conceptual ideas for how GIS is implemented.


what GIS is like in college with super clean data
what GIS is like in the real world with messy data

Terminology II – AI and ML



Data Science

- All about data
- Data manipulation
- Analysis of patterns

Artificial Intelligence

- Enables machines to execute reasoning by replicating human intelligence

Machine Learning

- Subsection of AI
- Systems learn automatically and improve from experience

Deep Learning

- Form of Machine Learning
- Use of Neural Networks
- High-dimensional arrays

Working with data: Art & Science

Data exploration

- One-dimensional Data
- Two-dimensional Data
- Many-dimensional Data

Cleaning and munging

- Data types and conversions
- Missing values and outliers

Rescaling

Dimensionality reduction

'Experts often possess more data than judgment'
- Colin Powell

What is Geospatial Artificial Intelligence (GeoAI)?

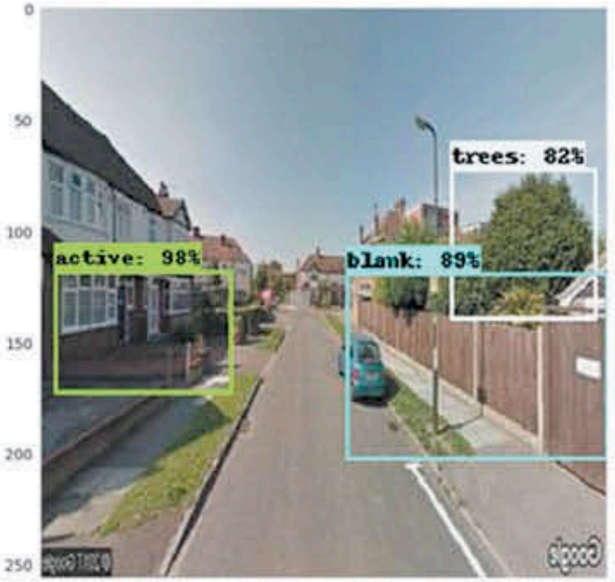
Recent progress in Artificial Intelligence (AI) techniques, the large-scale availability of high-quality data, as well as advances in both hardware and software to efficiently process these data, are transforming a range of fields from computer vision and natural language processing to autonomous driving and healthcare. For instance, the availability of high-resolution geographic data and high-performance computing techniques together with deep learning fuel progress in fast and accurate object detection.

Recent examples of GeoAI work include the detections of terrain features (Li and Hsu 2020) and densely-distributed building footprints (Xie et al. 2020), information extraction from scanned historical maps (Duan et al. 2020), semantic classification (e.g., LiDAR point clouds) (Guo and Feng 2020), novel methods for spatial interpolation (Zhu et al. 2020), and advances in traffic forecasting (Polson and Sokolov 2017, Ren et al. 2020). Similarly, machine learning and natural language processing are facilitating the extraction of geographic information from unstructured (textual) data, such as news articles and Wikipedia pages (Hu 2018) as well as the matching of natural features in multiple gazetteers (Acheson et al. 2020).

Janowicz, K., et al. (2019). GeoAI: spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond, IJGIS, https://doi.org/10.1080/13658816.2019.1684500

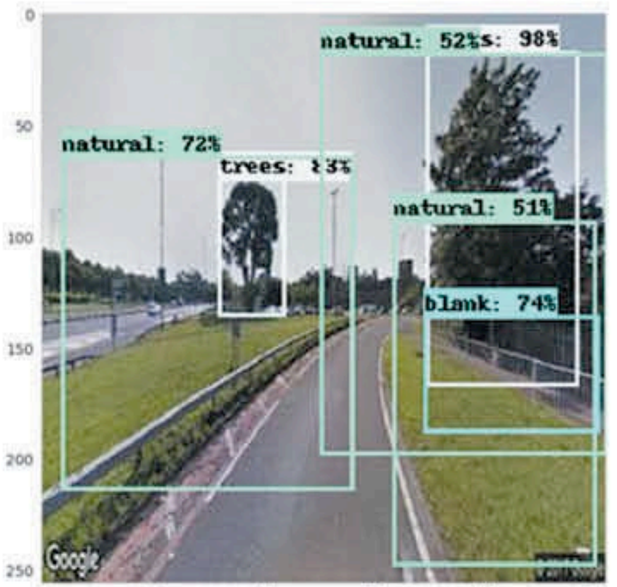
Street-Frontage-Net: urban image classification using deep convolutional neural networks

Blank Frontages



Single-Active Frontages

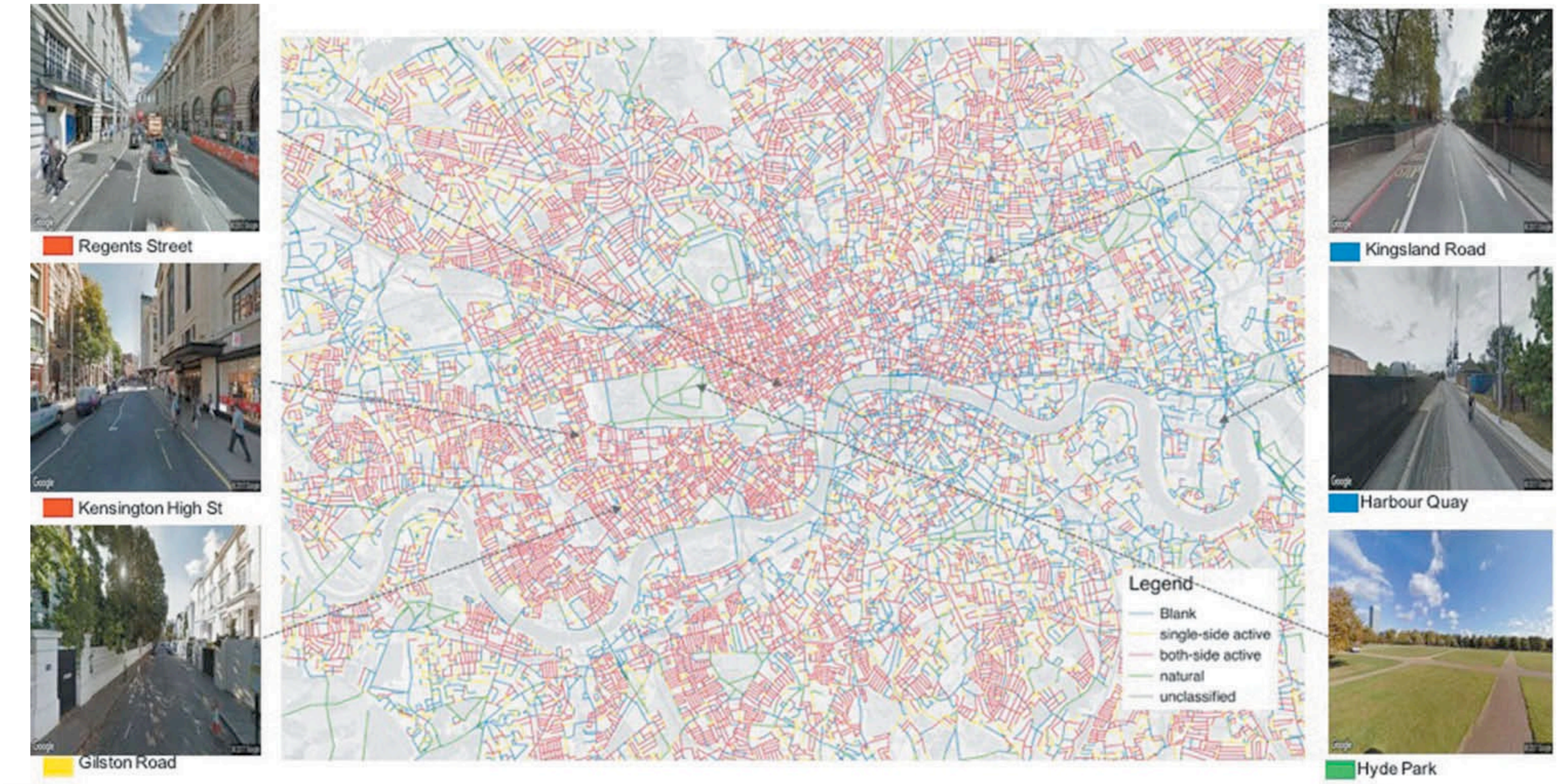
Both-Active Frontages



Non-Urban Frontages

Law, S., et al., 2020. Street-frontage-net: urban image classification using deep convolutional neural networks. International Journal of Geographical Information Science, 1–27.

Street-Frontage-Net: urban image classification using deep convolutional neural networks



Law, S., et al., 2020. Street-frontage-net: urban image classification using deep convolutional neural networks. International Journal of Geographical Information Science, 1–27.

Efficient Machine Learning for Large-Scale Urban Land-Use Forecasting in Sub-Saharan Africa

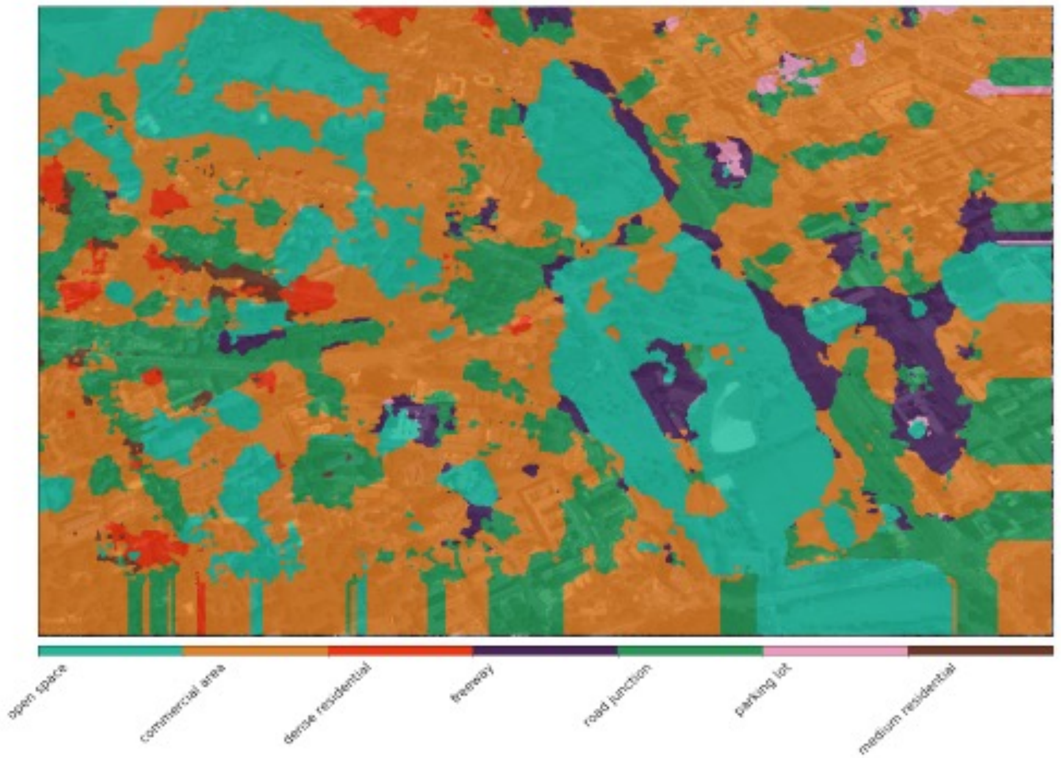


Figure 2. Segmented map from patch-based prediction

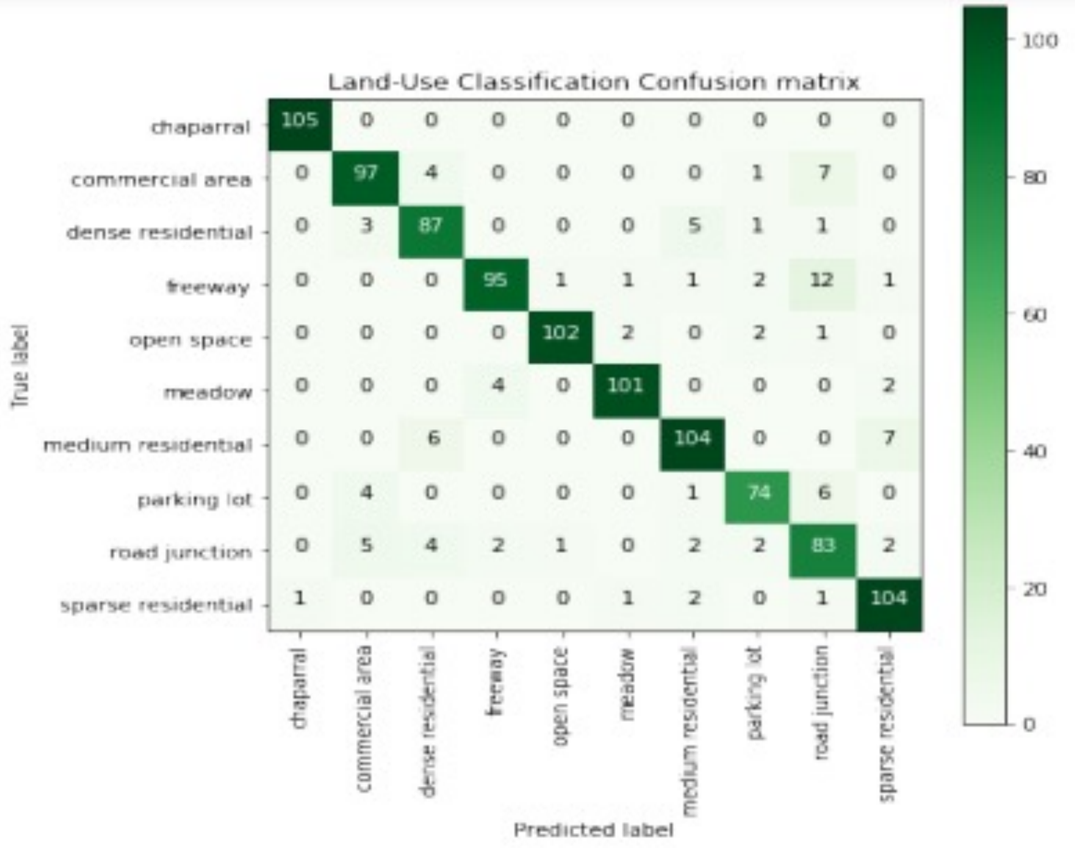
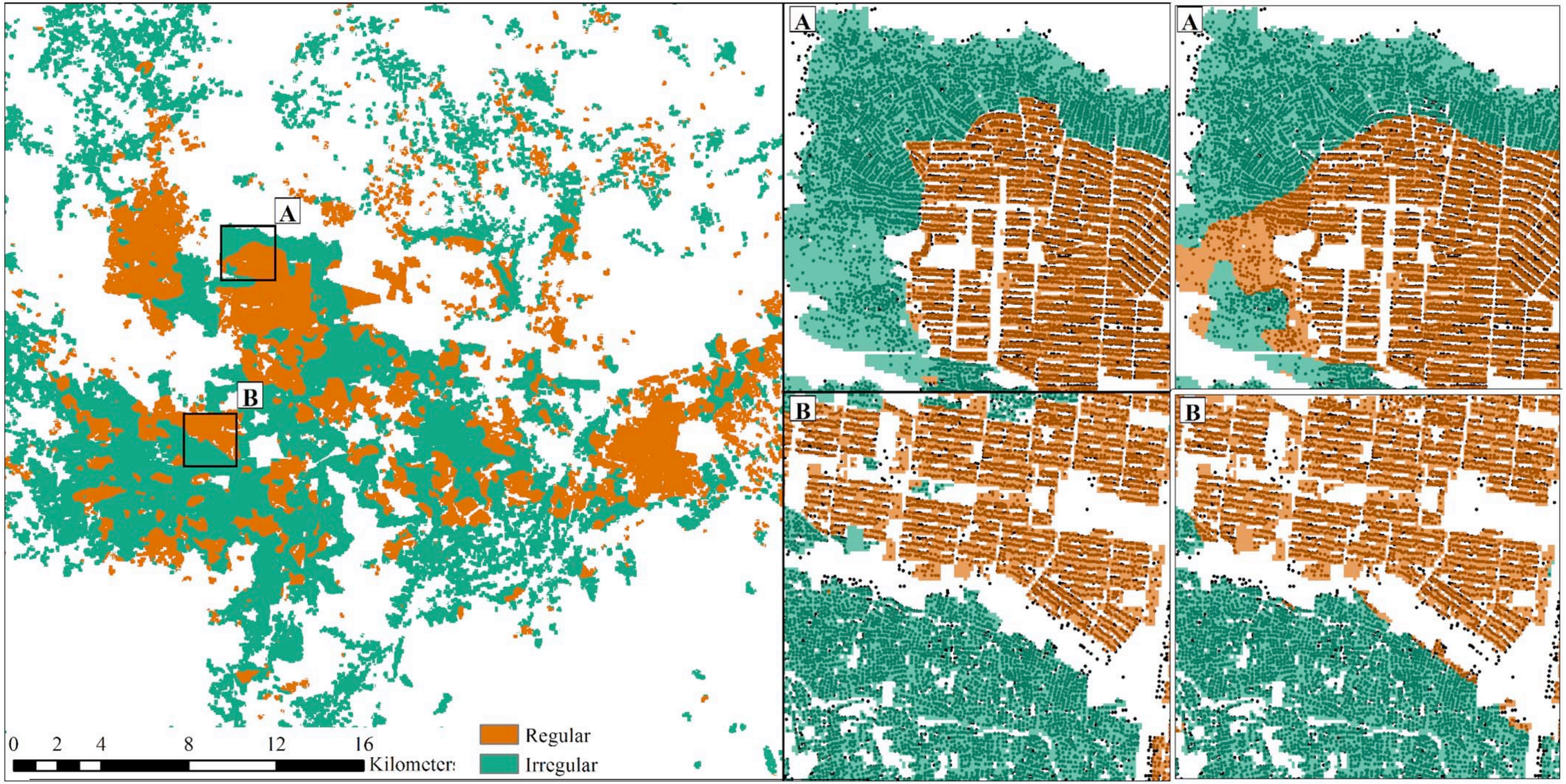


Figure 1. Confusion matrix for model's performance, overall precision and recall values are 0.907 and 0.908 respectively.

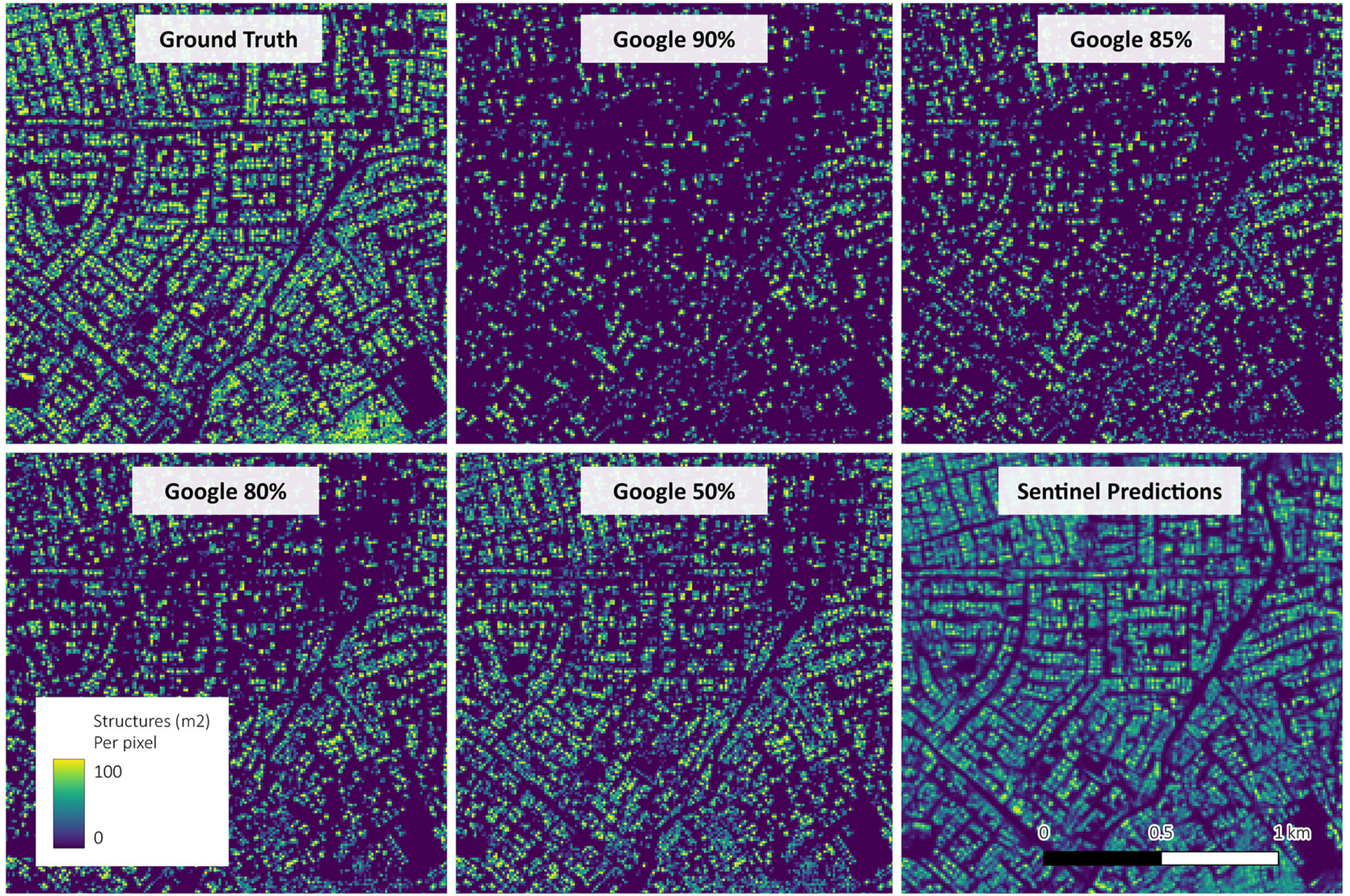
D. Omeiza, "Efficient Machine Learning for Large-Scale Urban Land-Use Forecasting in Sub-Saharan Africa," 2019.

Identifying residential neighbourhood types from settlement points in a machine learning approach



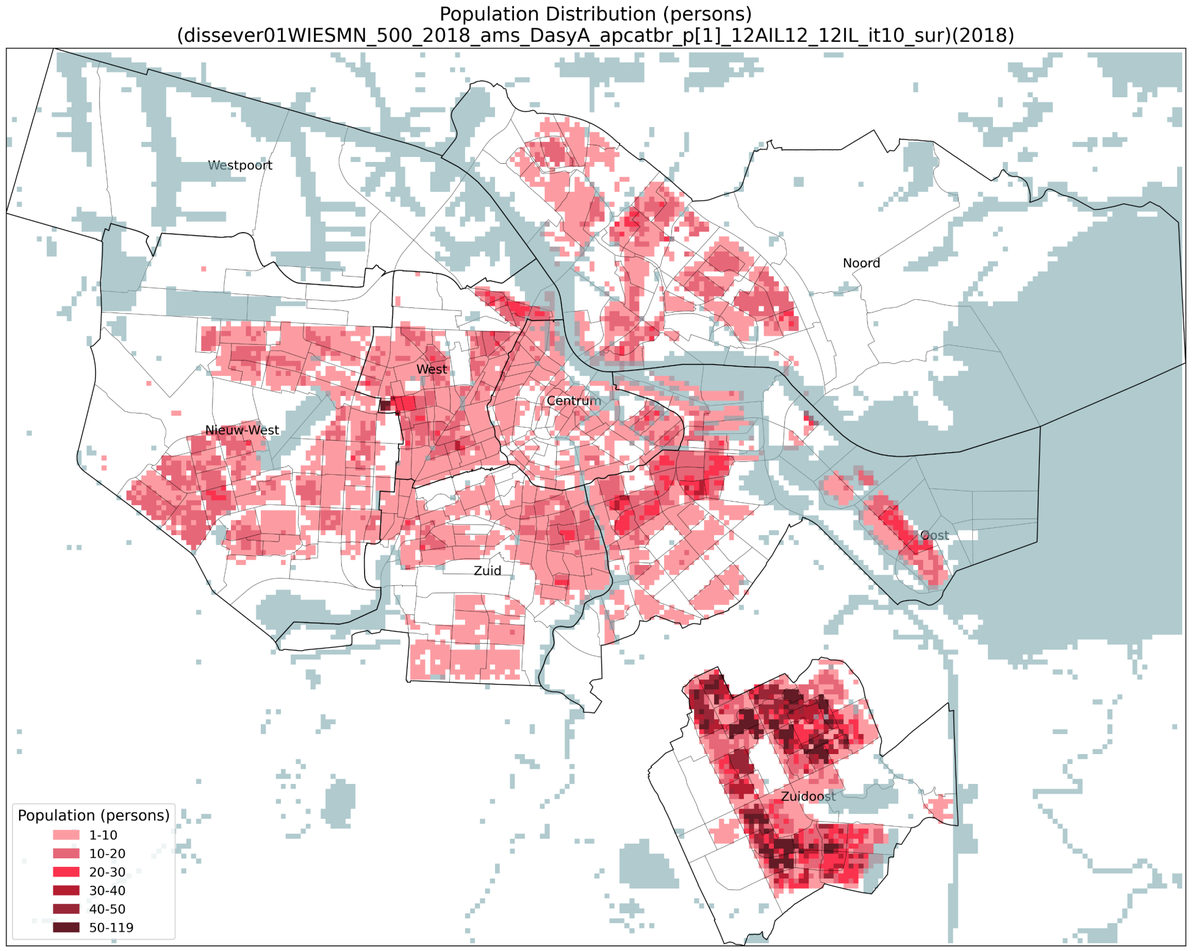
Jochem WC, Bird TJ, Tatem AJ. Identifying residential neighbourhood types from settlement points in a machine learning approach. Computers, Environment and Urban Systems. 2018;69:104-113.

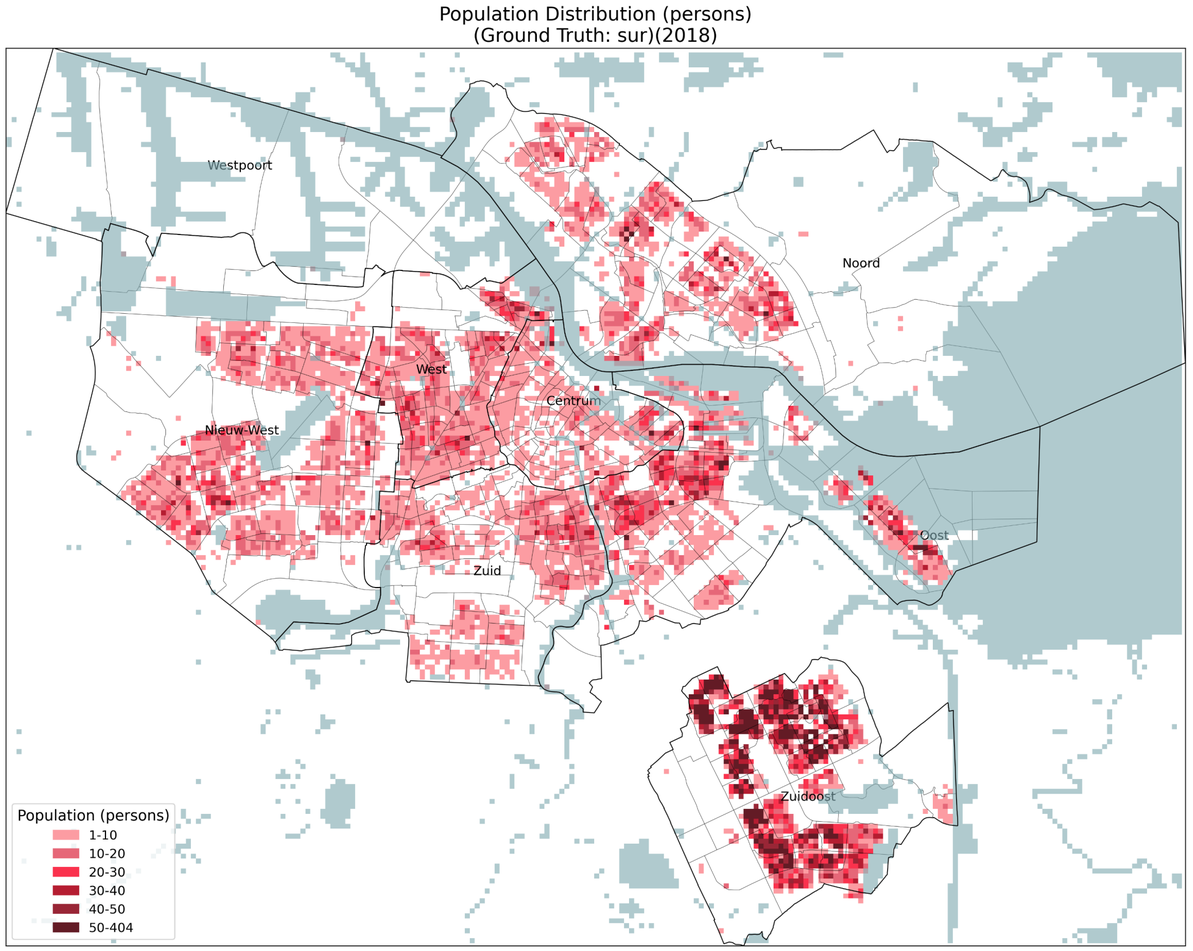
A deep learning method for creating globally applicable population estimates from sentinel data



Fibæk, C.S., et al., 2022. A deep learning method for creating globally applicable population estimates from sentinel data. Transactions in GIS, 00, 1–29.

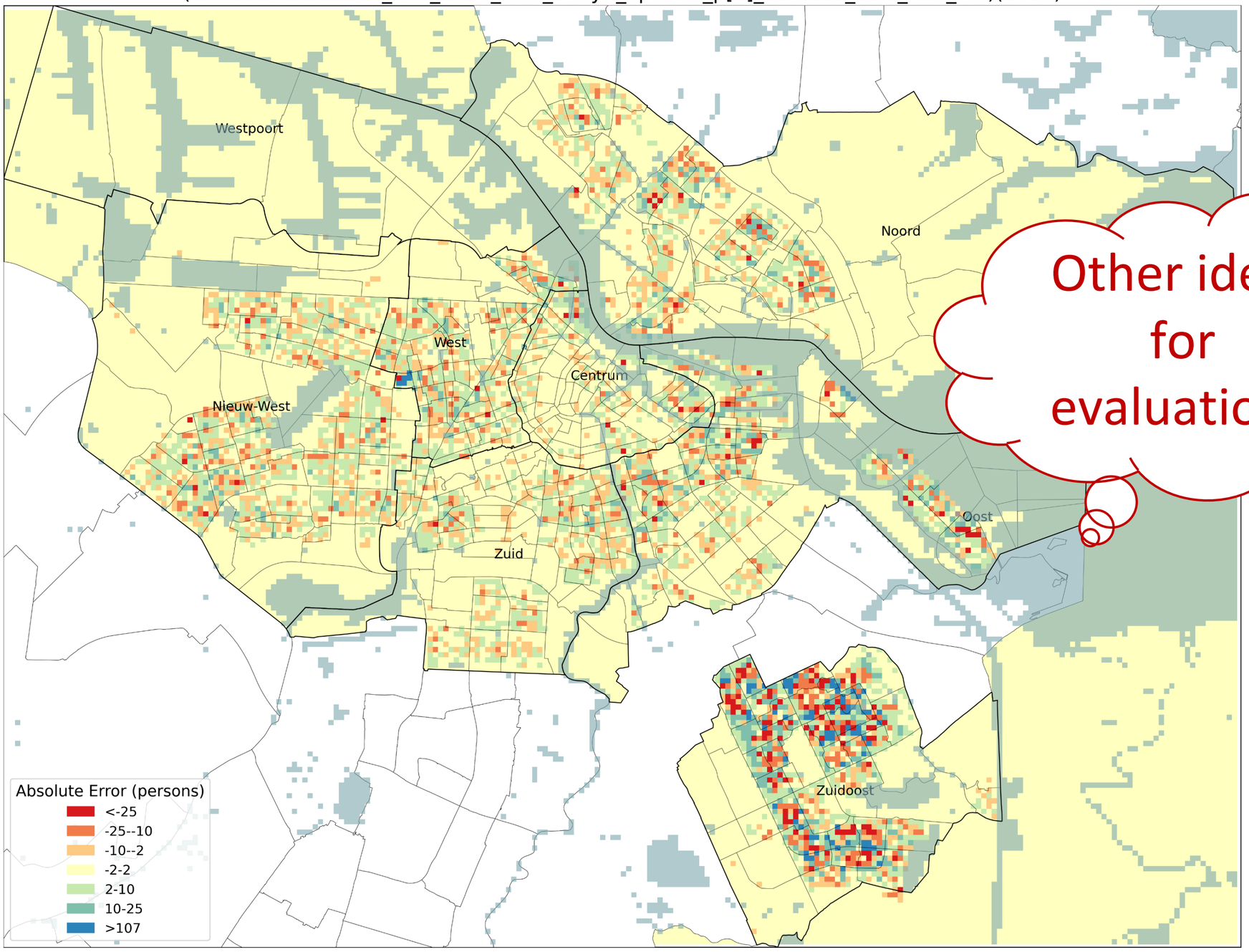
Population estimations





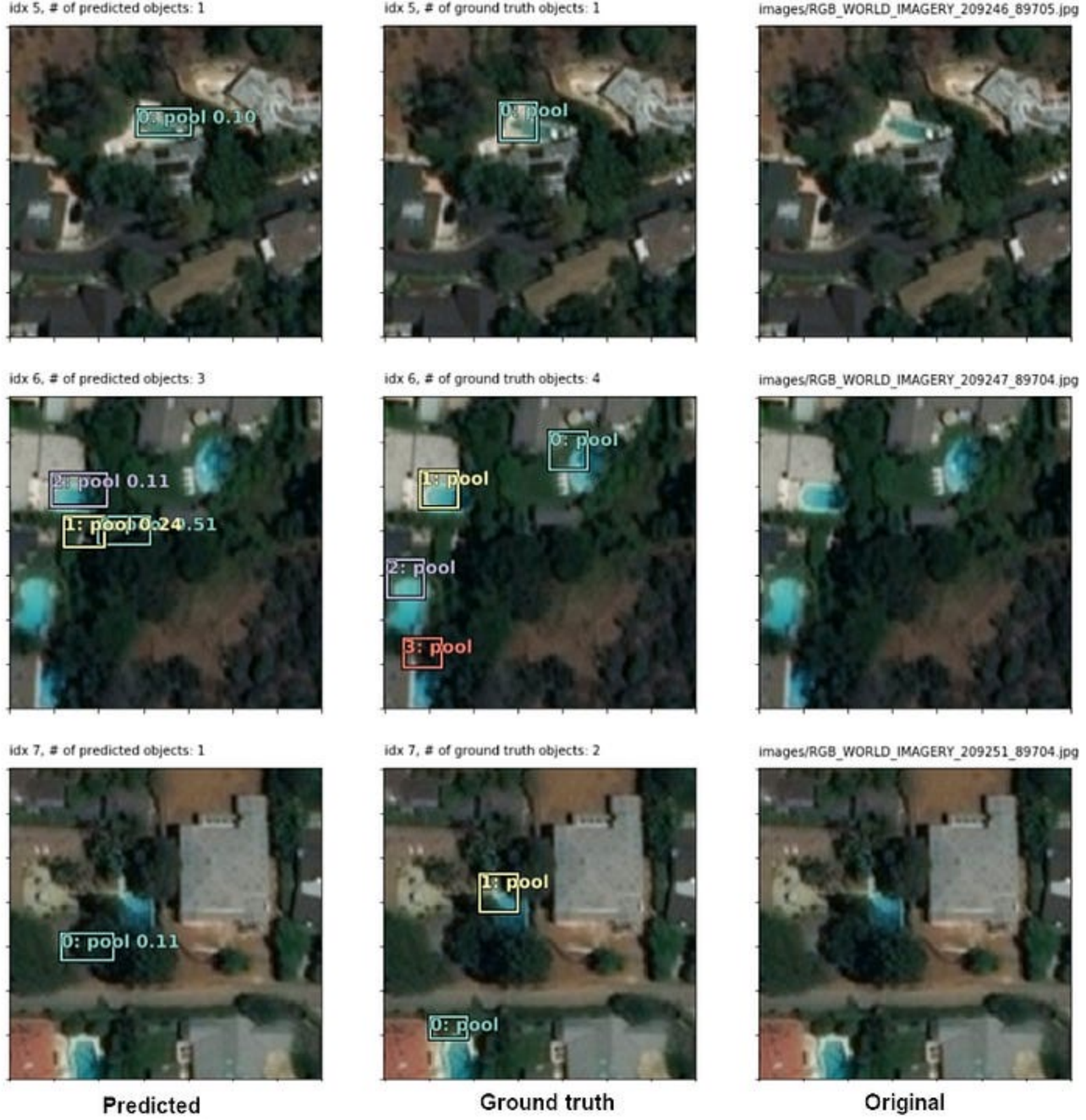
M. Georgati, J. Monteiro, B. Martins, C. Keßler, and H. S. Hansen, "Multi-Output Spatial Disaggregation of Population Data Using Gradient Boosting and Deep Learning Models."

Absolute Error (persons)
(dissever01WIESMN_500_2018_ams_DasyA_apcatbr_p[1]_12AIL12_12IL_it10_sur)(2018)



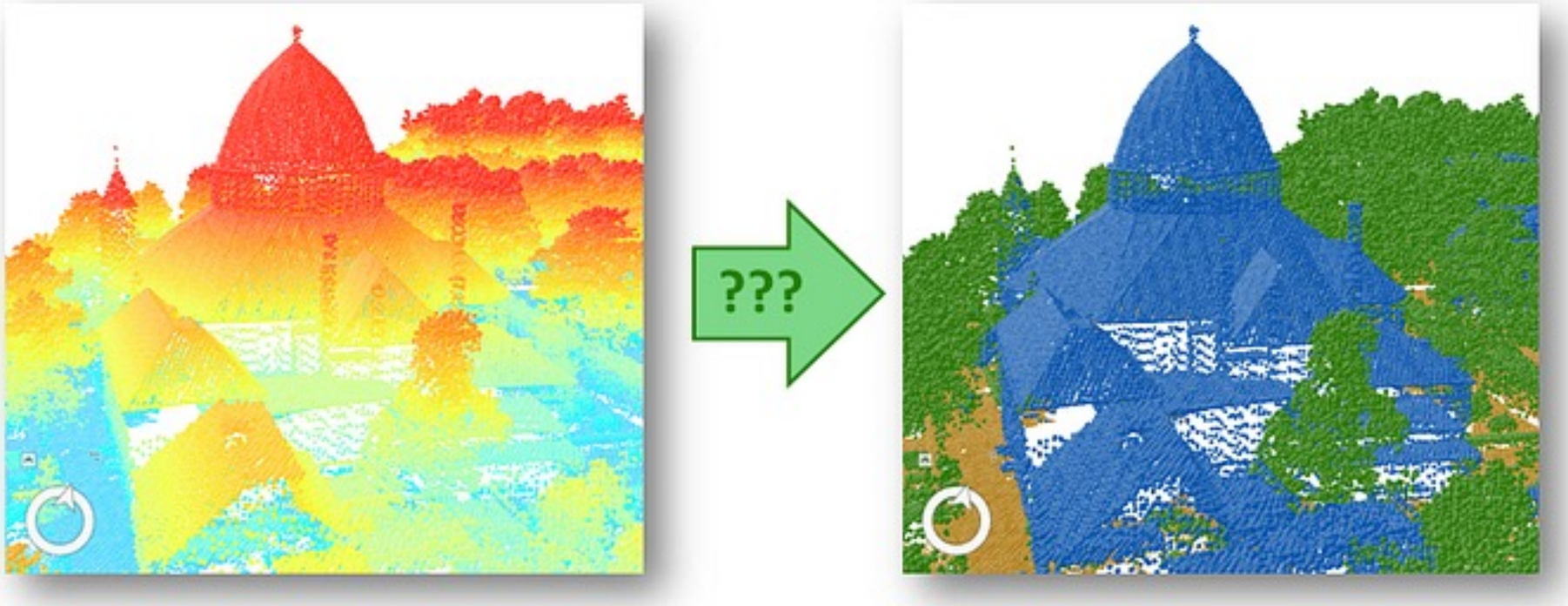
Other ideas for evaluation?

Population Swimming pool detection and classification using deep learning



https://medium.com/geoai/swimming-pool-detection-and-classification-using-deep-learning-aaf4a3a5e652

3D cities: Deep Learning in three-dimensional space



From a pure X-Y-Z point cloud to classes of objects… Is this possible with Deep Learning?

---

https://medium.com/geoai/3d-cities-deep-learning-in-three-dimensional-space-29f9dafdfd73

Why spatial is special in AI?

Location of events is interesting because everything is located. But also, because understanding where things happen is key element to understanding why they happen in the first place.

Geographic location is often the key for linking heterogeneous data sets that have been intensively used for training advanced AI models

Mutual contributions:

- Geographical questions/ unsolved problems can now be addressed better using AI rather than traditional methods
- With AI, new theories and approaches can be developed to build models and data pipelines in GIS

We can make more powerful machine learning models.

Janowicz, K., Sieber, R., & Crampton, J. (2022). GeoAI, counter-AI, and human geography: A conversation. Dialogues in Human Geography, 12(3), 446–458. https://doi.org/10.1177/20438206221132510

Responsible AI

Accountability and who determines what constitutes accountability (big tech companies, researchers , individual and collective responsibility).

Janowicz, K., Sieber, R., & Crampton, J. (2022). GeoAI, counter-AI, and human geography: A conversation. Dialogues in Human Geography, 12(3), 446–458. https://doi.org/10.1177/20438206221132510

Two ways of looking at GeoAI

- Tool builders: Applying AI to a geo-specific problem.

- Building spatial explicit models and bake into AI or ML models an understanding of geographic space and time - A great research question

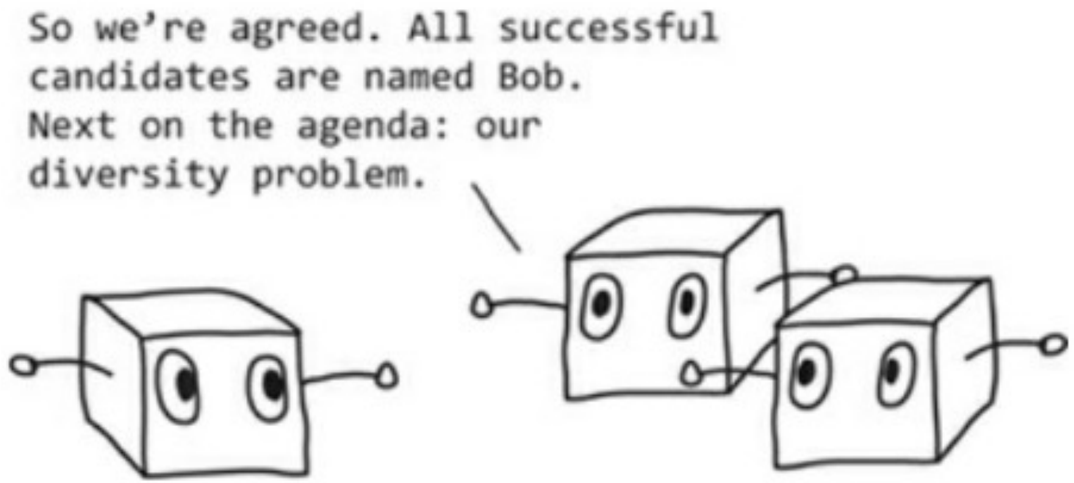
Spatially explicit models perform better than non-spatially explicit models in this domain.

Machine Learning is great for:

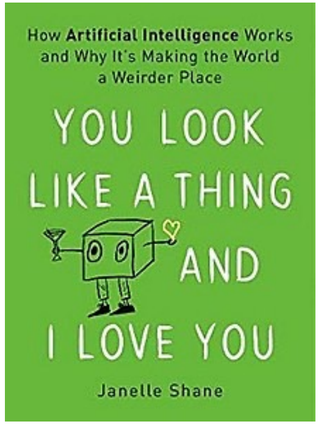
- Problems that require a lot of hand-tuning or long lists of rules
- Complex problems with no good solution through traditional means
- Fluctuating environments
- Getting insights about complex problems and large amounts of data

Challenges of ML

- “Bad algorithm”
- “Bad data”

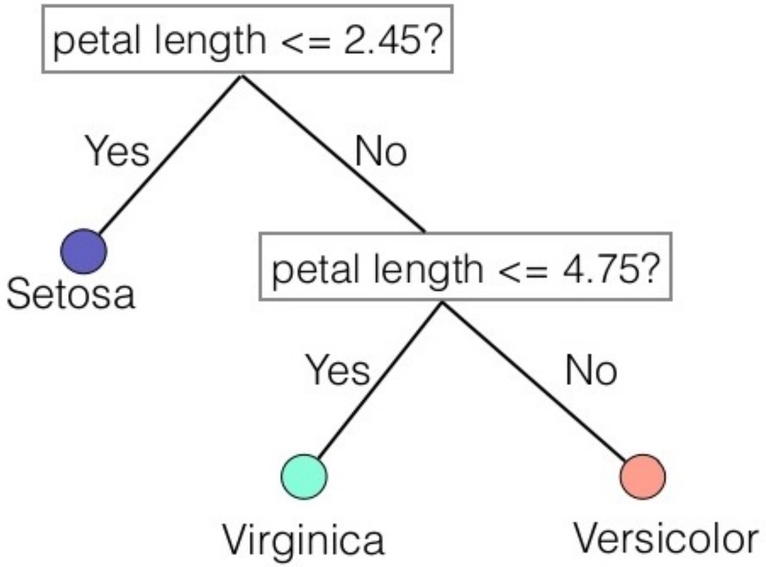


- ‘The narrower the task, the smarter the AI’
- ‘Insufficient data does not compute’

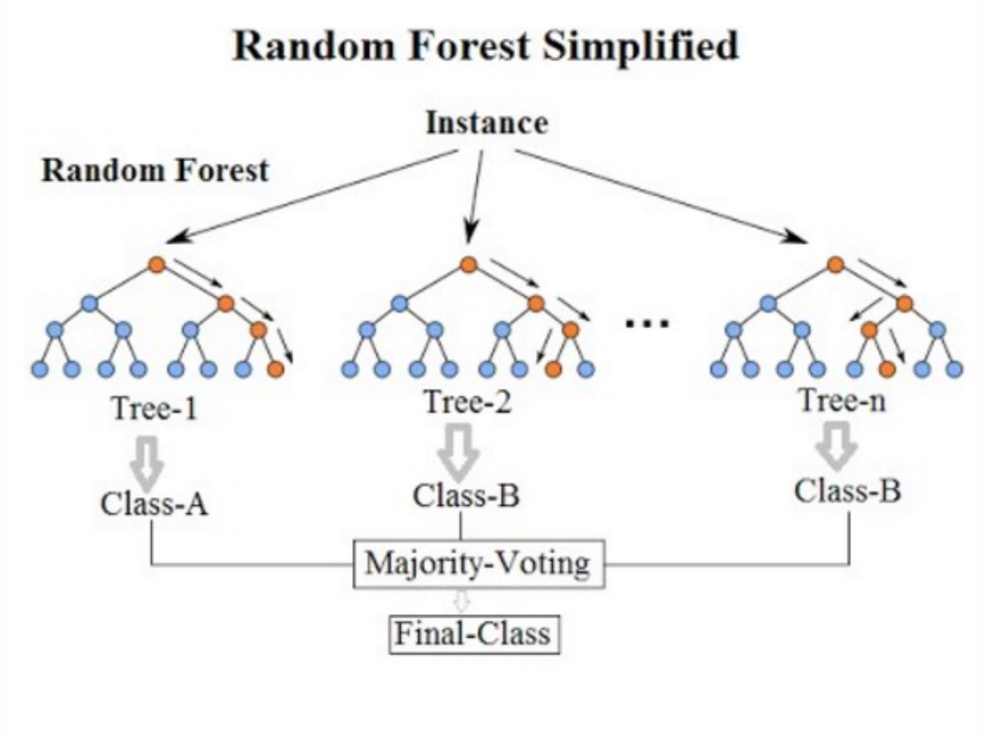


J. Shane (2019) You Look Like a Thing and I Love You: How Artificial Intelligence Works and Why It's Making the World a Weirder Place

Decision Trees and Random Forest



Overfits easily



Resistant to overfitting

Practically, in python …

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification

X, y = make_classification(n_samples=1000, n_features=4,
                           n_informative=2, n_redundant=0,
                           random_state=0, shuffle=False)


clf = RandomForestClassifier(max_depth=2, random_state=0)
clf.fit(X, y)


print(clf.predict([[0, 0, 0, 0]]))
```

Practically, in python …

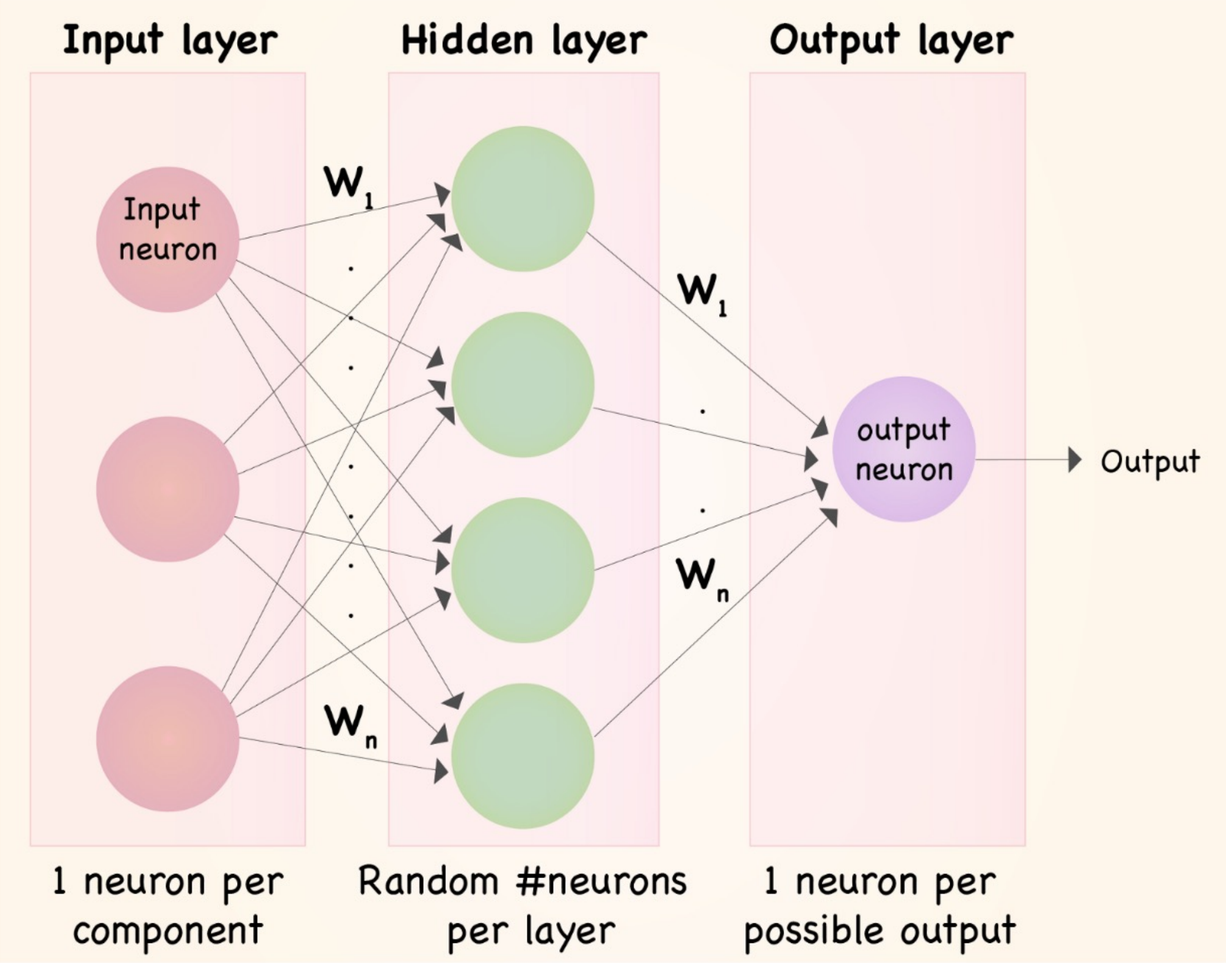
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.datasets import make_regression


X, y = make_regression(n_features=4, n_informative=2,
                       random_state=0, shuffle=False)


regr = RandomForestRegressor(max_depth=2, random_state=0)
regr.fit(X, y)


print(regr.predict([[0, 0, 0, 0]]))
```

Deep Neural Networks



- Requires lots of computing power
- Can be very broadly applied
- More difficult to tune than random forest
- Optimizers, loss functions, activation functions etc..

- Types: GANs, RNN (LSTM), CNN

Practically, in python …

```
# Load the tensorflow library
from tensorflow.keras import Model, Input
from tensorflow.keras.layers import BatchNormalization, Dropout, Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.losses import SparseCategoricalCrossentropy
```

Practically, in python …

```
# First step is to define our model.

# How many hidden layers should it have?
# How should the weights initially be set?
# How many neurons should each layer have?
# Which activation function should we use?

def define_model(shape, name):
    model_input = Input(shape=shape[1:], name="input")

    model = Dense(64, activation="relu", kernel_initializer="he_normal")(model_input)
    model = Dropout(0.1)(model) # <-- Reducing overfitting by disabling neurons
    model = Dense(32, activation="relu", kernel_initializer="he_normal")(model)
    model = Dense(16, activation="relu", kernel_initializer="he_normal")(model)

    # We need 5 neurons in our output. This corresponds to the classes we want to predict.
    # softmax activation outputs a probability of each class.
    predictions = Dense(5, activation='softmax')(model)

    return Model(inputs=[model_input], outputs=predictions)

model = define_model(X_train.shape, "input")
```

Practically, in python …

```
# Compile the model. Using Adam (Industry standard) and the above loss function.
# Adam takes a learning rate, which is the most important parameter to tune
# if you want to improve your model performance.
model.compile(
    optimizer=Adam(learning_rate=0.001),
    loss=SparseCategoricalCrossentropy(from_logits=True),
)

# Fit the training data
model.fit(
    x=X_train,
    y=y_train,
    epochs=25, # <-- How many times will we update the weights
    verbose=1,
    batch_size=16, # <-- How many rows in each step before updating
    validation_split=0.2,
)

# Predict our test set
y_pred = model.predict(X_test)
y_pred[0]
```

Tools in Python

Geodata Science

Pandas

Geopandas

- Shapely
- Fiona
- Matplolib

NumPy

PySal

Machine & Deep Learning

- Scikit-learn
- Tensorflow
- PyTorch
- Torchgeo

Visualization

- Matplotlib
- Seaborn
- Bokeh (D3.js)
- PySal (Viz)


ChatGPT
every dev
right now
Stack Overflow
Documentations
made with mematic

Your experiences…

- What was the problem and the purpose?
- Regression? Classification?
- What data did you use to train the model?
- What model did you use?
- How did you assess the performance?

Can you use ML in the frame of your project?

Discuss in groups

- What would be the problem and the purpose to address?
- Is it a regression or classification problem?

- How would you state the problem so that the machine understands it?
- What model would you use? Why?

- What data would you need to train the model?
- How would you assess the performance?

Today’s exercise

Prediction of Population Distribution with Random Forest Regressor

Problem

How can we estimate the population distribution in Copenhagen at high resolution with open data?

Tools: Pandas, Numpy, Rasterio, Scikit-learn

Short introduction to Random Forest Regression

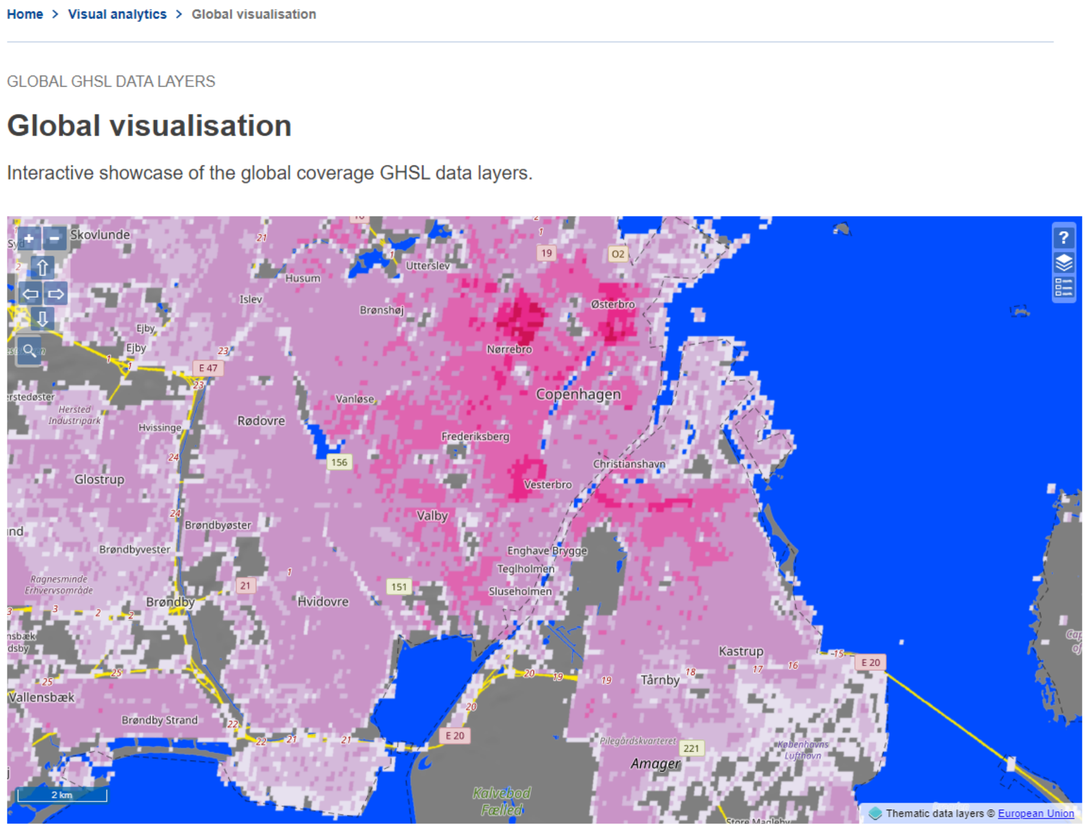
Prediction of Population Distribution with RF

We will use the GHS-POP layer as target layer

What training datasets are necessary for predicting the distribution of the population in Copenhagen?

Discuss in groups and put your ideas here

Menti : **4303 8910**

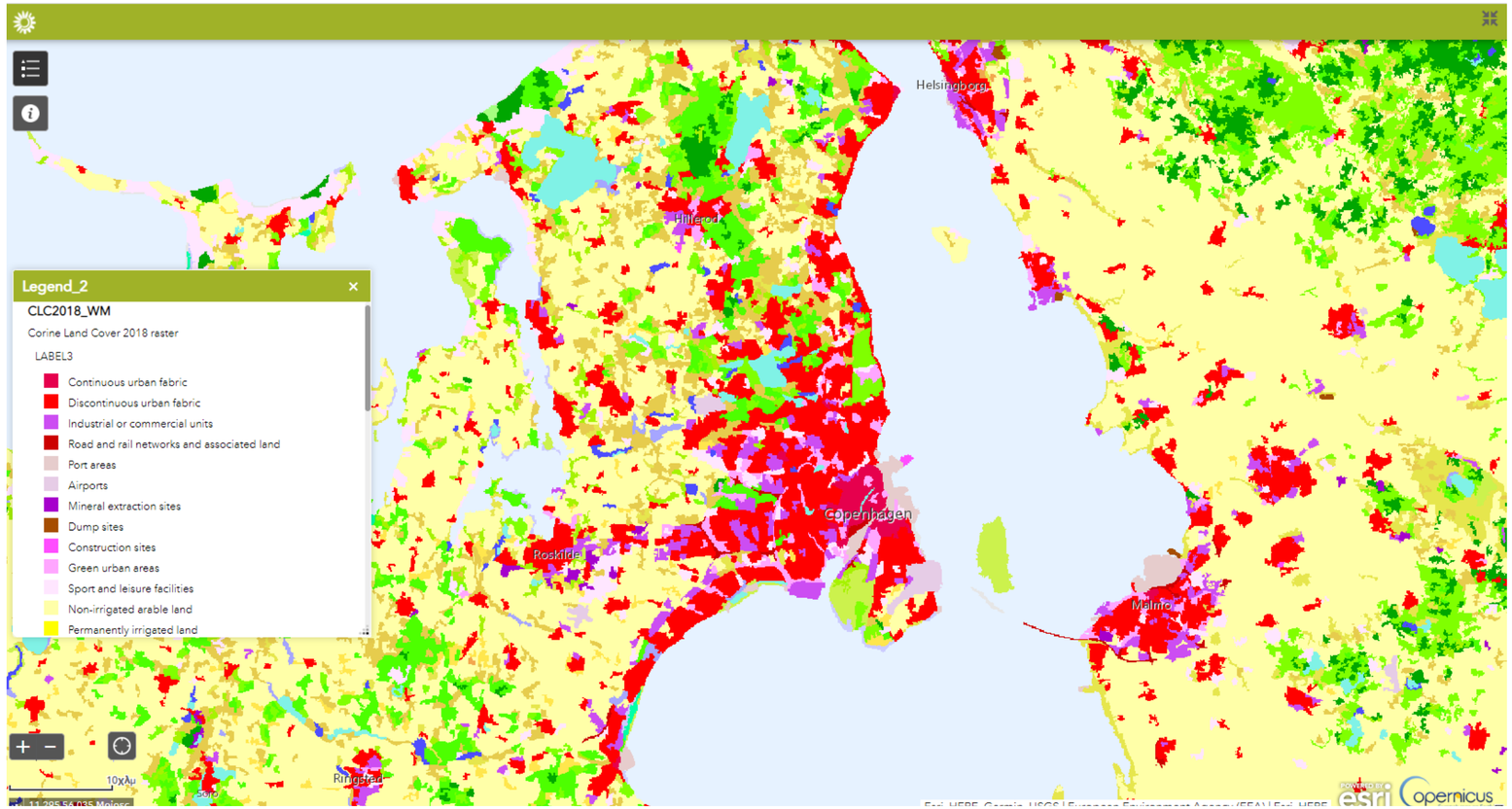


Available datasets in Learn-IT

Take a look at the provided datasets and make some visualizations in python.

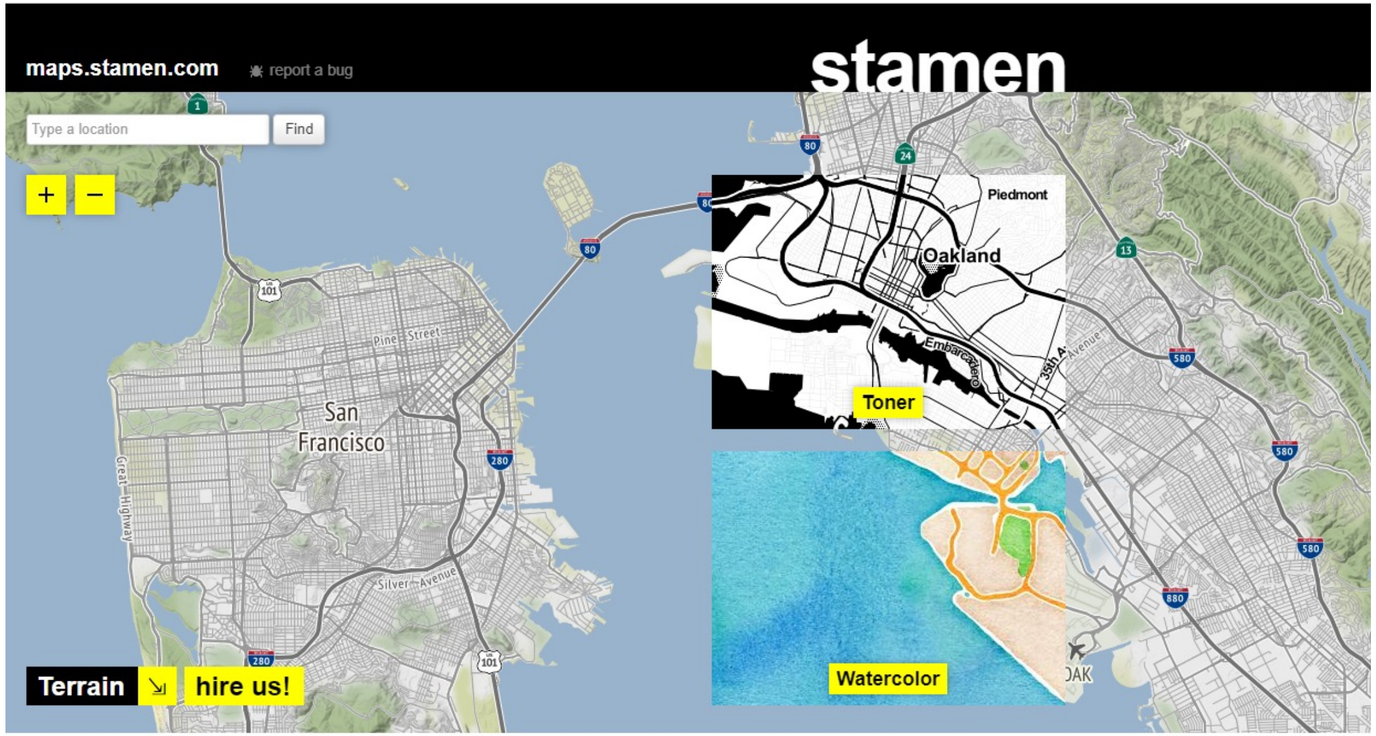
What are your observations?

Can you produce other useful datasets?



Next week: Mobility

That’s all for today!!!



Next time together, GeoAI
Until then, questions at LearnIT or geor@itu.dk