AiGrunn· 2026

TITAAN

Building a Foundation Model for Dutch Aerial Imagery

Joey Antonisse · Data Scientist

Sopra Steria · for Het Waterschapshuis

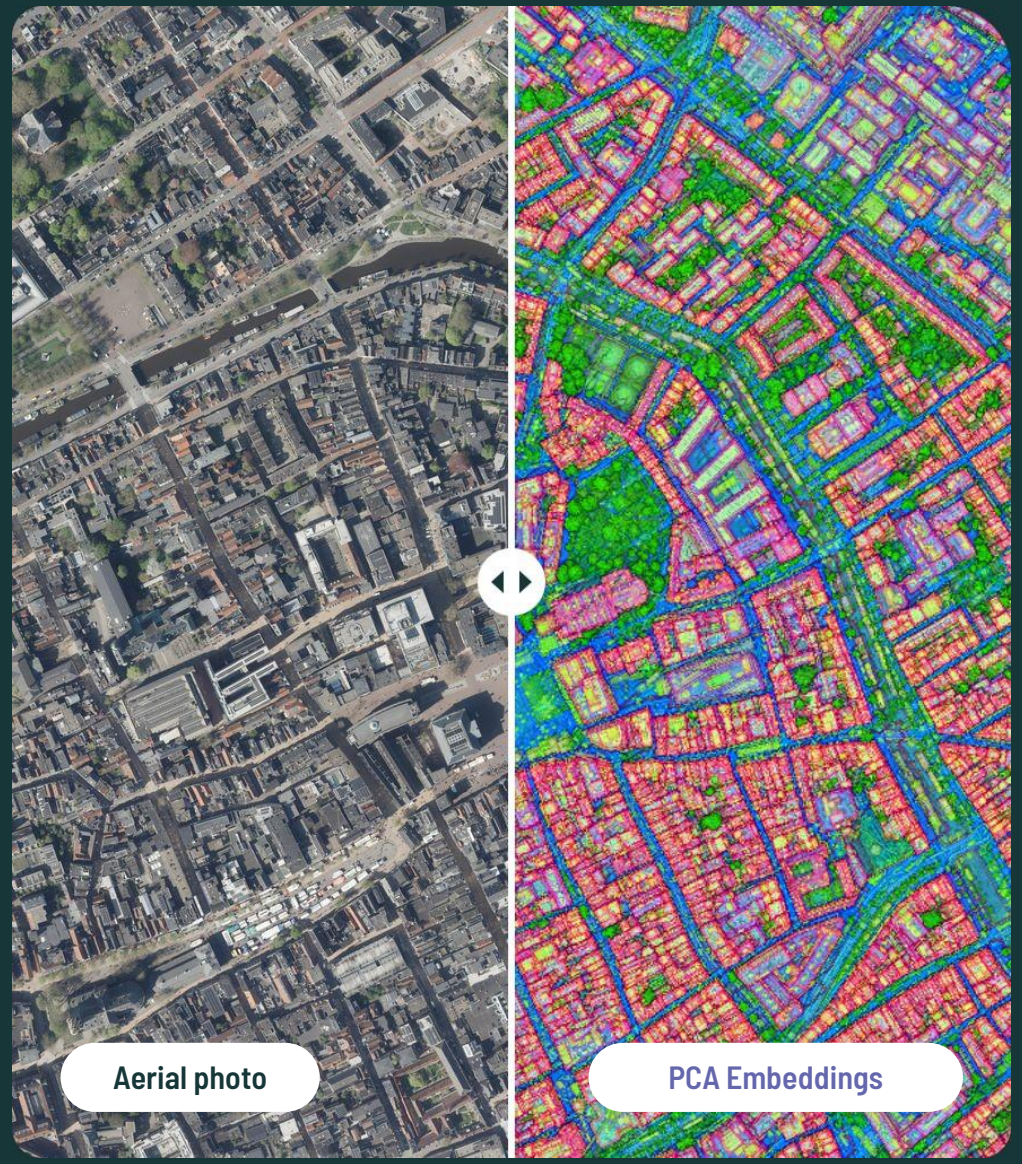




Photo: Herman Groenbroek

About me

From software to AI

I'm Joey, Data Scientist at Sopra Steria in Groningen. I work on TITAAN for Het Waterschapshuis.

How I got here

01

Software engineer

Web Development, Laravel

02

Master's degree

Intelligent Systems & Visual Computing

03

Data Scientist

Computer vision, now for Dutch aerial photos

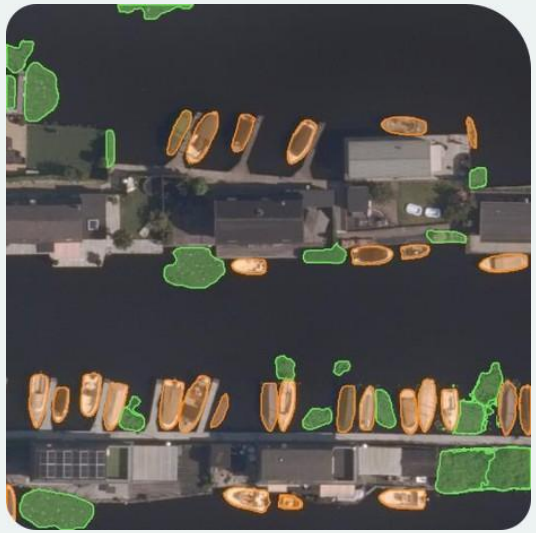


Away from the screen

Classical piano. Still practising.

Same photos, a different question every time

The whole country is photographed from the air every year. Every organisation looks for something else in it.



Waternet / AGV

Boats, floating plants



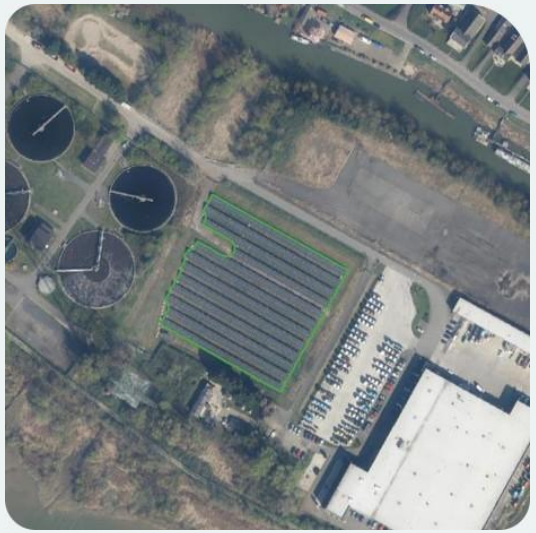
A water authority

Jetties along the water



Kadaster

Greenhouses



Kadaster

Solar fields

Aerial imagery: Beeldmateriaal Nederland, via PDOK

The old way: one question, one full project

For jetties (steigers) we made a dataset, chose a model, trained it and tested it. It worked.

01 **Make a dataset** → 02 **Choose a model** → 03 **Train the model** → 04 **Test and deploy**



× N

Then came boats, buildings, floating plants…

Every new question restarts at step 1.

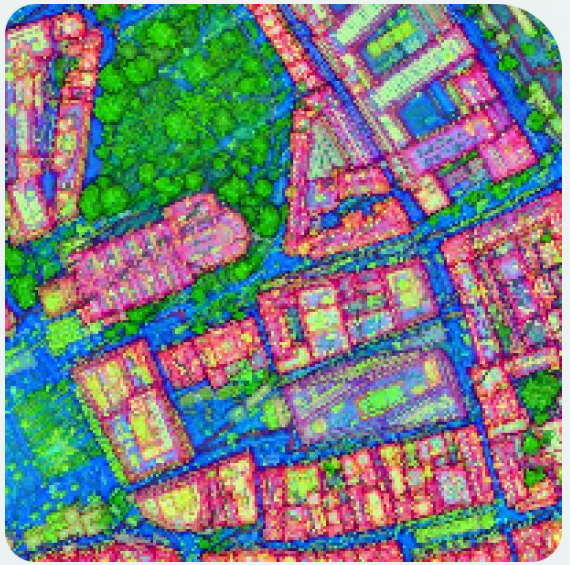
The idea: learn the landscape once

A foundation model learns from imagery without labels. Each new question only adds a small 'head'.



Foundation model

trained once, without labels



- Water
- Buildings
- Jetties
- Boats
- Change

1 Lots of photos, no labels

2 One big model, trained once

3 An embedding for each 16x16 pixels

4 Small heads, a few labels each

Training data

The model learns from colour, infrared and height at once, and describes each square with 1,024 numbers.



1 km × 1 km · 3 layers

The training data

Colour photo, infrared and height (AHN)

448 × 448 pixels

One patch

36 m, the three layers stacked

16 × 16 pixels

One square

1.28 m: a tree, in all three layers

the embedding

1,024 numbers

One description for all three layers

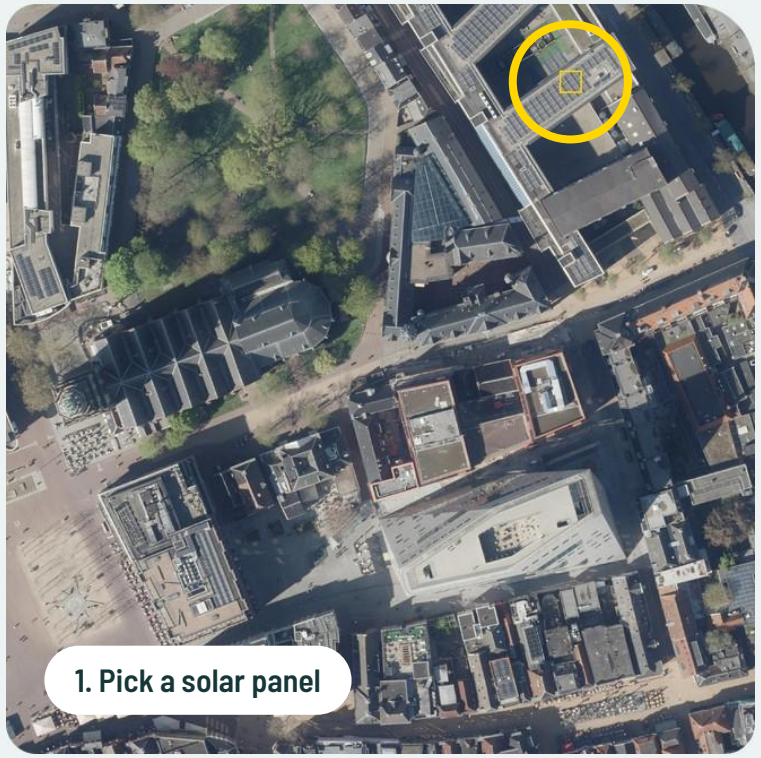
PCA → colour

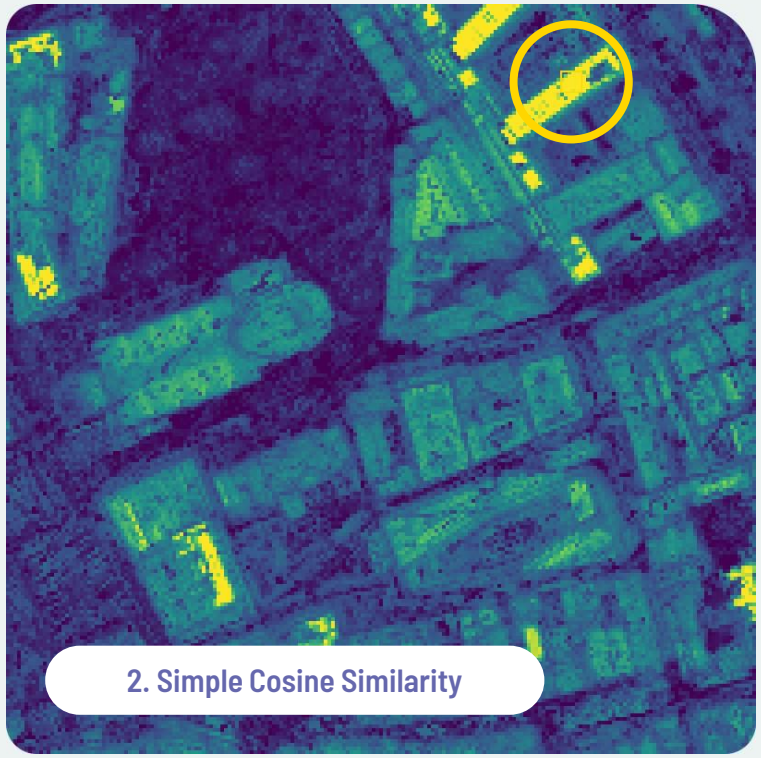
Every square

Groups the model found by itself

Point at a solar panel, find the others

Similar places get similar embeddings. So you can point at a solar panel and search for the rest.





TITAAN-SD, colour photo only.

Bright = very similar

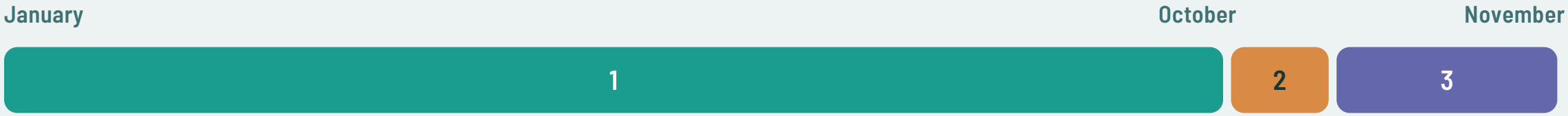
Every square is coloured by how much it resembles the circled point.

No labels, no training

We only compared embeddings. A small head with labels makes it a detector.

Building it took three attempts

Most of the year went into attempt 1. Attempts 2 and 3 followed within a few weeks in October.



Attempt 1

TerraMind

Train our own model that combines all sensors, from scratch.

Works, but slow and complex

Attempt 2

DINOv3

7 billion parameters, trained on 1.7 billion images

Better

Attempt 3

DINOv2

300 million parameters

Open and strong: TITAAN-SD

Attempt 1: we trained our own model

TerraMind hides most of an image and learns to predict what is missing. We trained it on our own data.



1 Cut the photo into patches

2 Hide most of them

3 The model predicts what is missing

9 months

TerraMind is built for satellite images. For our 8 cm photos we had to train every part ourselves.

The result: TITAAN-MTP. After training it for 2 weeks on 8xH100 GPUs

Compare models: a full probe test set

The big model is frozen. We train only a small head on its embeddings, and score it on unseen areas.

Aerial photo
an area never seen in training

→

Big model
frozen: learns nothing new

→

Embeddings
one for every 1.28 m square

→

Small head
trained on a few labels

→

Score
compared with the official dataset

In this talk: the buildings score

Can a small head draw the buildings from the model's embeddings?

Score is based on IoU

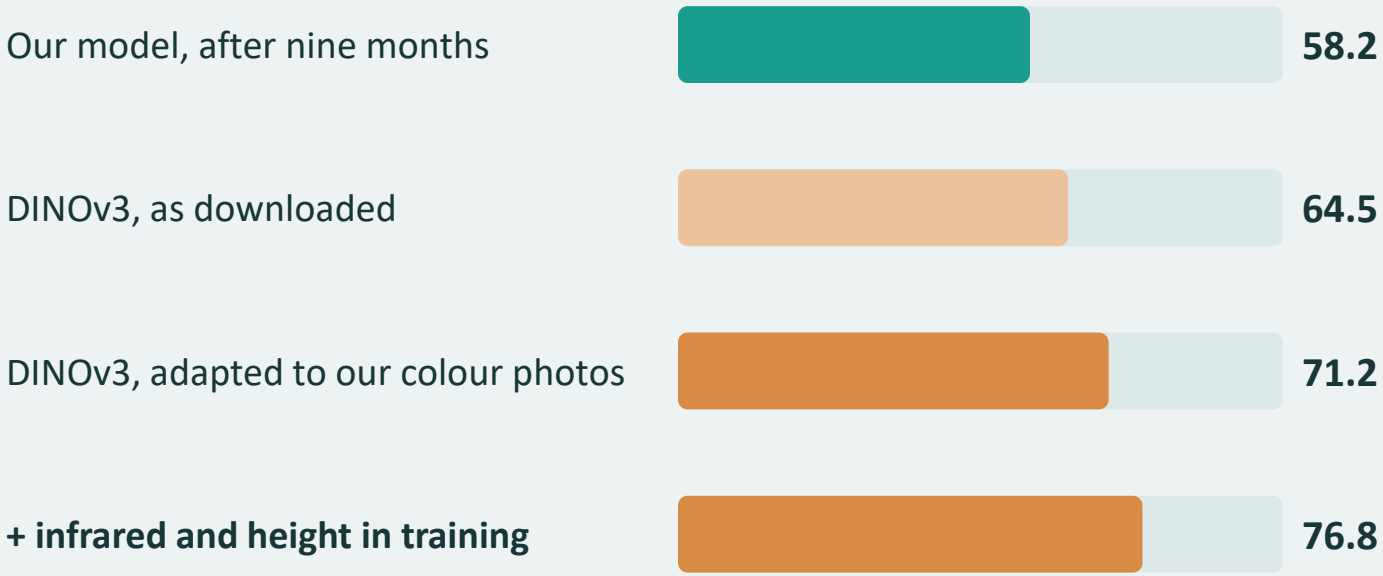


Score = overlap ÷ total area

Higher is better. Scale: 0 to 100.

Attempt 2: a model we never trained beat ours

DINOv3 (ViT-L) learned from ordinary web photos. It had never seen our imagery, and still beat our own model.



3 days

to adapt DINOv3 to our own photos, without labels. Our own model took nine months.

Buildings score (0 to 100, higher is better) · colour photo only

Then we read the licence

We are a public-sector project. Other organisations must be able to use, adapt and share our model.

DINOv3 · attempt 2

DINOv3 License

Its conditions pass on to every model you derive from it.

→

DINOv2 · its predecessor

Apache 2.0

Free to use, adapt and share.

Our way forward

Lesson: check the licence of the model weights on day one.

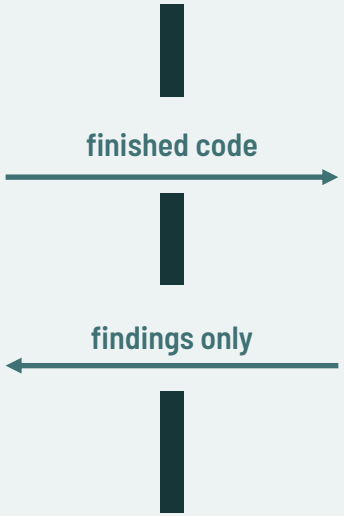
Attempt 3: the same recipe, in a clean room

Our attempt-2 code was built on DINOv3 code, so we rebuilt the recipe from papers, on the open DINOv2.

Implementation room

A fresh start that writes the new code. It may only see:

- our own code
- the open DINOv2 code
- published papers



Audit room

A separate auditor compares the result with what we may not use:

- the DINOv3 code
- its settings

TITAAN-SD The same training recipe, rebuilt from published papers on an open model.

How TITAAN-SD learns: student and teacher

Two copies look at the same place. The student sees less, but must describe it the same way.



the same place

Teacher

sees the whole view and every sensor

the teacher is a slow average of the student

Student

sees small crops, with sensors hidden

Embeddings must match

the only training signal

Step 1

Colour photos

Training on RGB

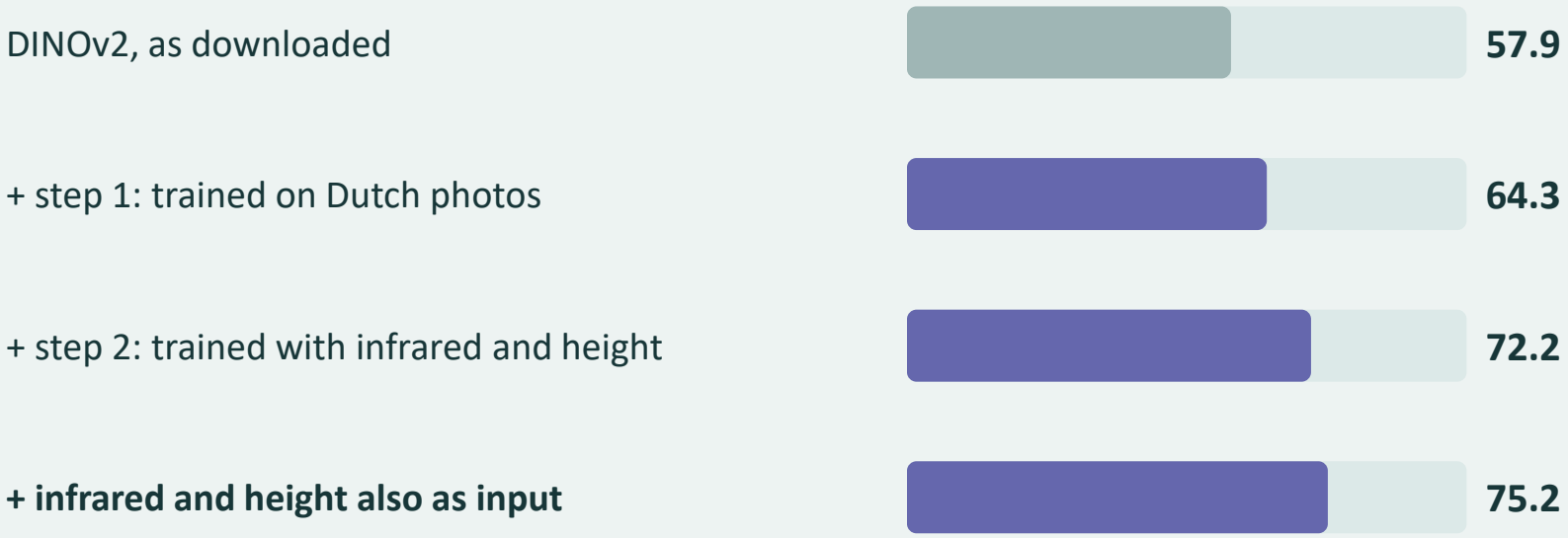
Step 2

+ infrared and height

Plus a mask that says where data is missing.

It works: every step adds something

The buildings score, from DINOv2 as you can download it to TITAAN-SD with all its inputs.



+17.3

points on buildings. No labels were used to train the model.

Honest note

Not everything improves. On boats and floating vegetation, the original DINOv2 stays slightly ahead.

Two models, each with its own strength

Attempt 1 was not wasted. With all sensors as input, the TerraMind model is still slightly ahead.

Colour photo only

the input we always have



SD wins

With all sensors

photo, infrared and height



MTP ahead

Buildings score (0 to 100, higher is better)

Which one do I use?

Only a colour photo

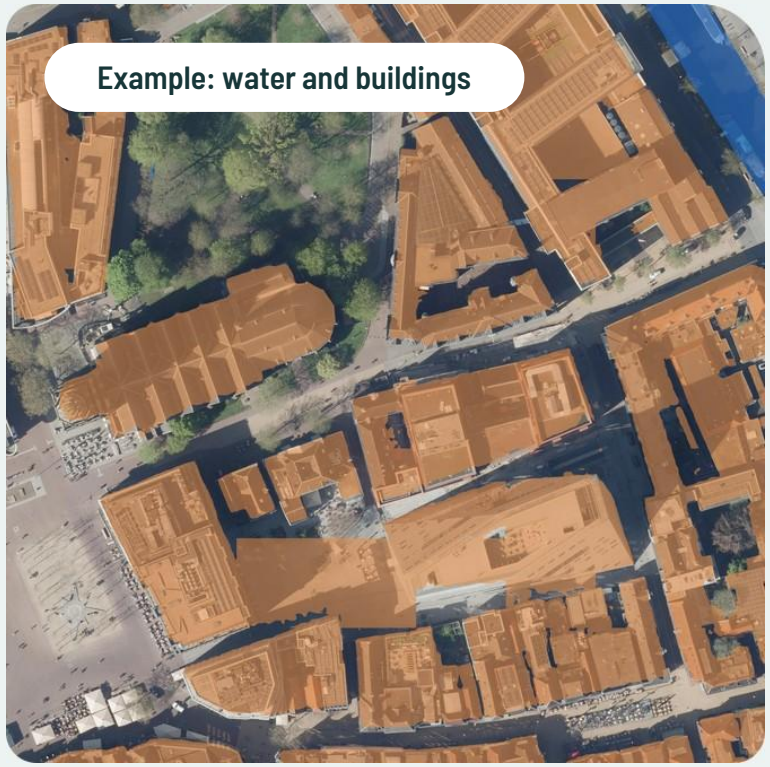
TITAAN-SD

Photo, infrared and height

TITAAN-MTP

A new question is now a small job

Bring your own labels, train a small head on the shared embeddings, and get a map.



BGT model, colour photo only. Blue: water. Orange: buildings.

Before: every new question

make a dataset → choose a model → train → test and deploy

Now: every new question

Your labels → **Small Head or Fine Tuning** → **Detections**

Same embeddings, four kinds of answers

- **Segmentation** — water, buildings
- **Object detection** — boats, greenhouses
- **Point detection** — tree tops
- **Regression** — a number per spot

Three things to take home

One lesson from each attempt.

From attempt 1 · TerraMind

Measure, and compare models

One probe test set, also for models you didn't train. It changed our plan.

From attempt 2 · DINOv3

Adapt before you build

Adapting a strong general model took days. Building our own took months.

From attempt 3 · DINOv2

Read the licence on day one

A clean room kept our release open.

Thank you

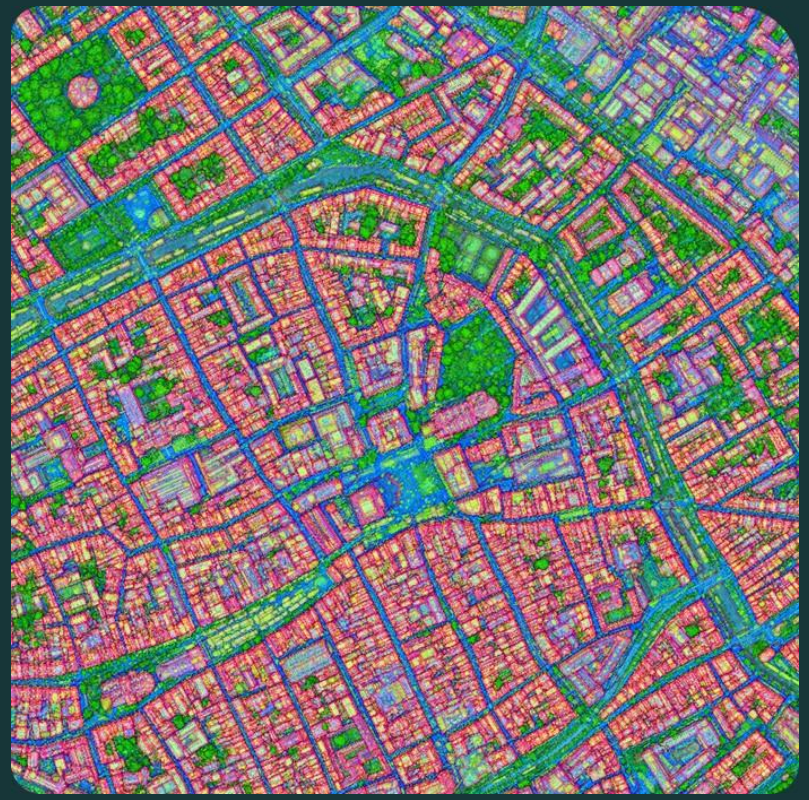
Next: Dutch foundation model for satellite images



joeyantonisse.nl
linkedin.com/in/joeyantonisse



Jonathan Gerbscheid
linkedin.com/in/jonathan-gerbscheid



huggingface.co/hwh-datascience

The raw material: five years of imagery

All three attempts learn from the same public data: aerial photos, infrared and height, on one grid.

16.6 TB
of imagery to download

455k
files, aligned on one grid

5 years
2022 to 2026, 8 cm per pixel

40 hours
just to download it

Input

RGB
The colour photo

8 cm per pixel. The one input we always have.

CIR
Colour-infrared

Shows plants and water differently from a normal photo.

AHN
Height

Two layers: the surface with roofs and trees, and the bare ground.

Labels

BGT
The official map

Buildings, water and roads. Our source of labels for testing.

All scores: colour photo only

Frozen model, small head. Scores from 0 to 100. A dash means: not measured in this set-up.

Task	DINOv2, as downloaded	TITAAN-SD step 1	TITAAN-SD step 2	TITAAN-MTP	DINOv3 (reference)
Water (IoU)	71.0	75.3	**77.2**	76.5	83.7
Buildings (IoU)	57.9	64.3	**72.2**	58.2	76.8
Jetties (AP)	51.2	**54.9**	–	43.5	61.3
Boats (IoU)	**76.8**	75.8	72.6	69.1	–
Floating plants (IoU)	84.7	82.6	84.4	**85.8**	–
Lost in shadow (lower is better)	18.4	–	**14.8**	24.9	–

Best value per row in purple, among the models we can share. DINOv3: after the same two steps, for reference only; its licence does not allow an open release, and it saw 448 instead of 392 pixels per patch. Source: project evaluation reports, October 2026.

All scores: photo, infrared and height

Frozen model, small head. Scores from 0 to 100.

Task	TITAAN-SD step 2 (03)	TITAAN-SD, best read-out	TITAAN-MTP	DINOv3 (reference, not open)
Water (IoU)	79.8	85.6	**87.2**	84.6
Buildings (IoU)	75.2	76.3	**77.5**	77.9
Jetties (AP)	58.5	**63.1**	57.2	63.4

- Best read-out: the average of the model's deeper layers (blocks 13 to 24).
- A second identical training run differed by about 1 point on buildings, so small gaps prove little.
- Much of the gain from the extra sensors comes from the 'missing data' mask.
- DINOv3: after the same two steps, for reference only (not open; saw 448 instead of 392 pixels per patch). Purple: best among the models we can share.