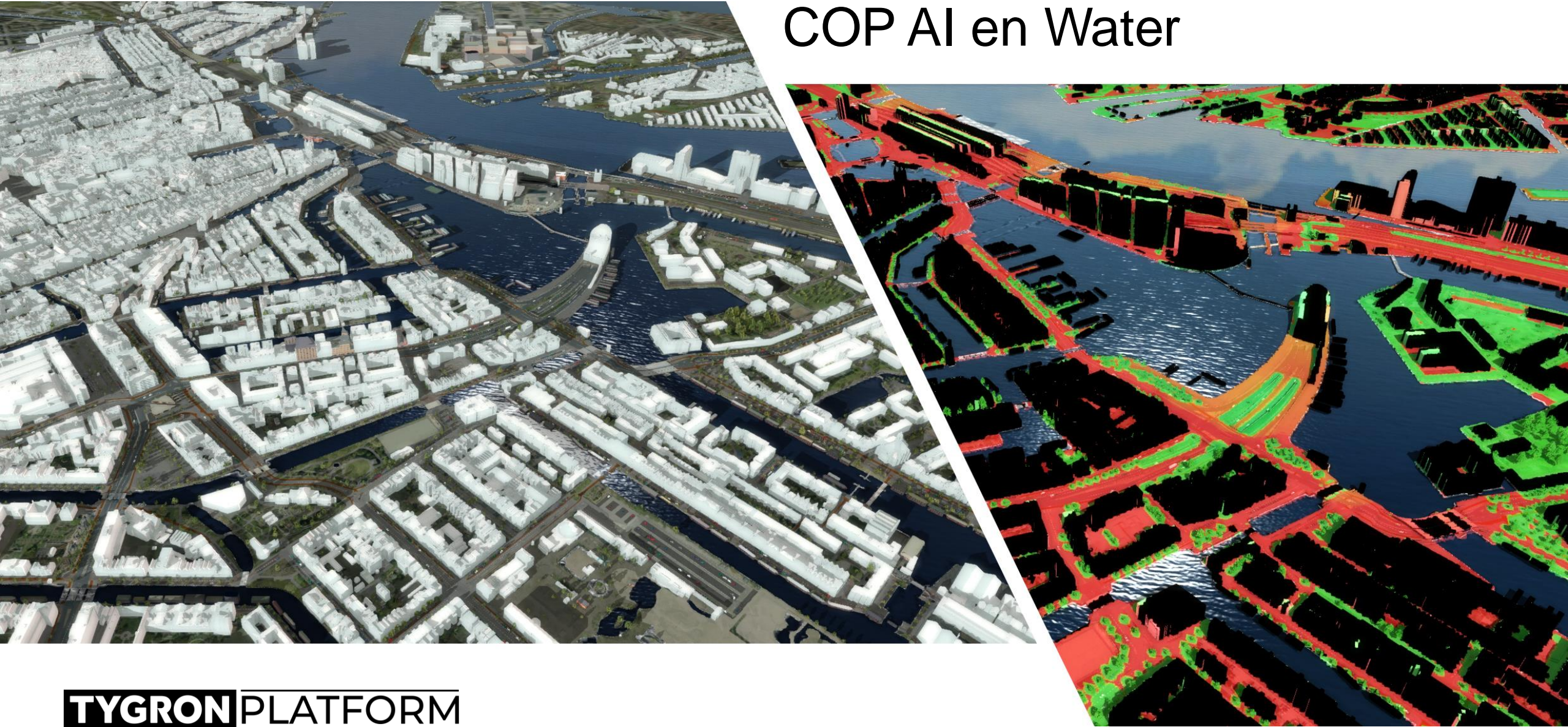


COP AI en Water



TYGRON PLATFORM

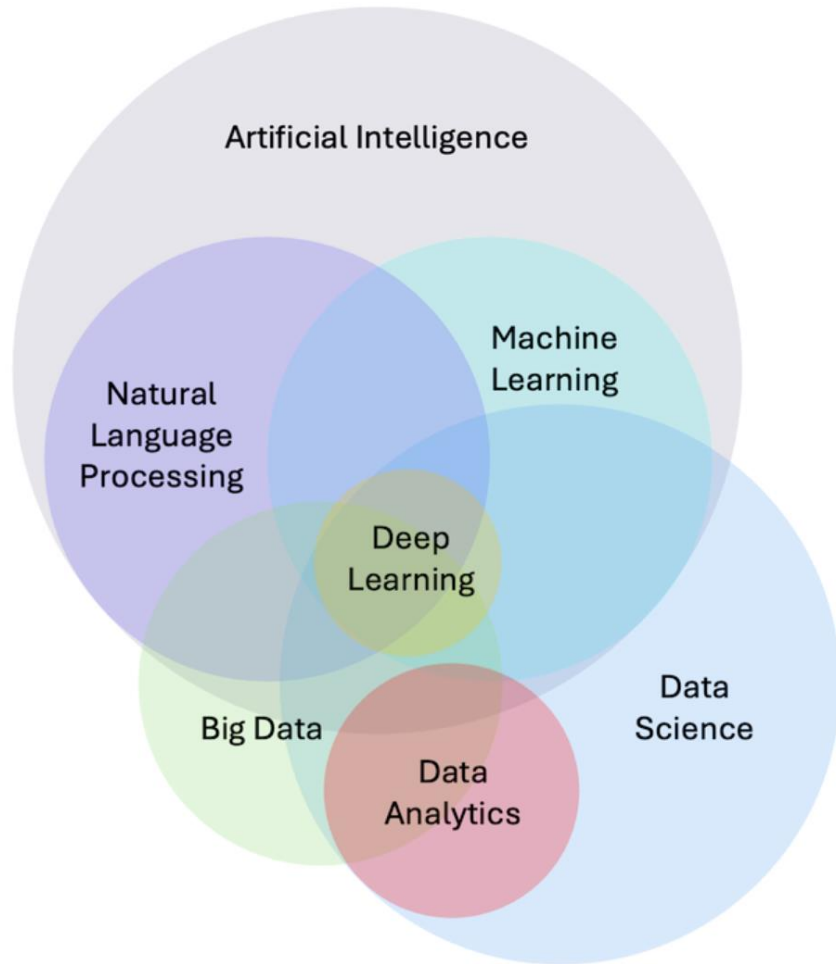
Accelerated Simulations for Engineers

Artificiële Intelligentie en de Tygron Watermodule.

15.00 – 15.35: Zaal 2. Water

- **Theorie Artificiële Intelligentie**
 - Wat verstaan we onder AI, wat gebruiken we in Tygron?
- **Wat kan de Tygron gebruiker hiermee**
 - Solution Looking For A Problem.....?
- **Zelf aan de slag?**

Theorie Artificiële Intelligentie



Deep Learning

Wat is het?

Bruikbare patronen uit data halen

Hoe?

Door gebruik van neurale netwerken

Euhm, wat betekent dat praktisch?

Door Python + Pytorch / TensorFlow

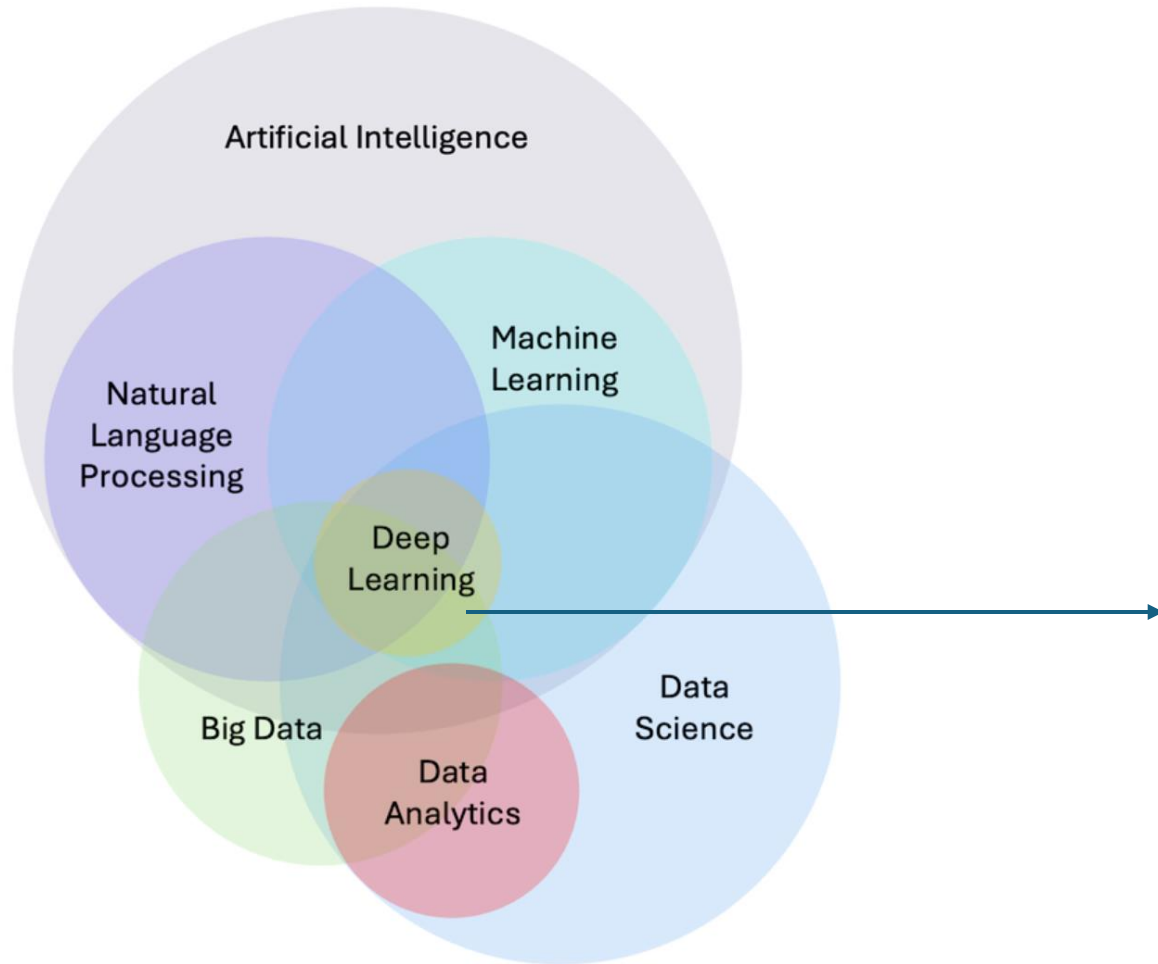


<https://www.clearlead.ai/blog/unified-intelligence-ai-ml-nlp-and-data-science>

TYGRON PLATFORM

Accelerated Simulations for Engineers

Theorie Artificiële Intelligentie



CNN (Convolutional Neural Network)



<https://www.clearlead.ai/blog/unified-intelligence-ai-ml-nlp-and-data-science>

Theorie Artificiële Intelligentie

Computer vision



5	7	12
28	9	3
55	92	61

R

17	42	2
25	58	31
38	63	71

G

51	71	22
89	99	34
57	92	61

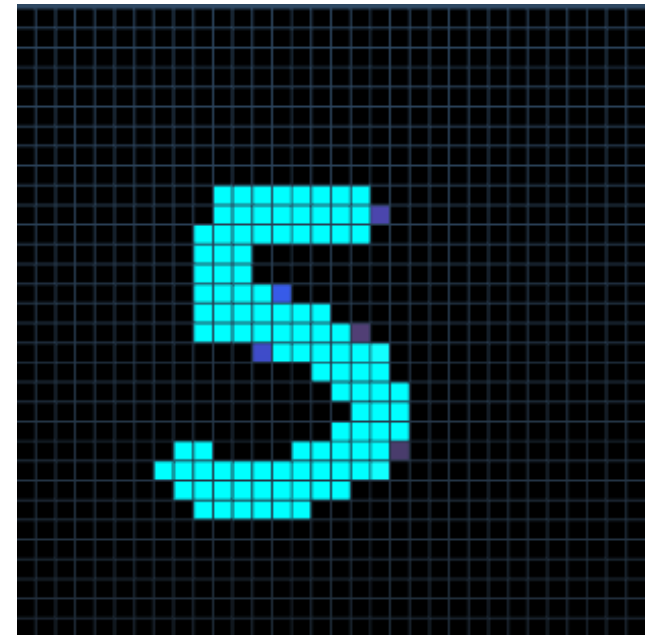
B

Classificatie:

- Oog = 95%
- Neus = 2%
- Oor = 4 %
- Knikker = 66 %

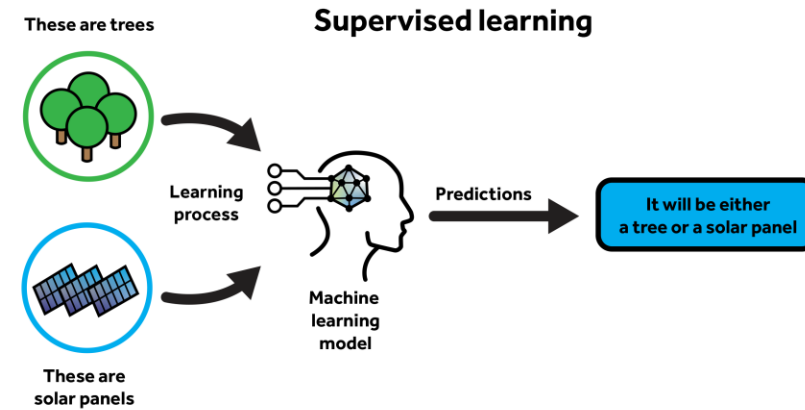
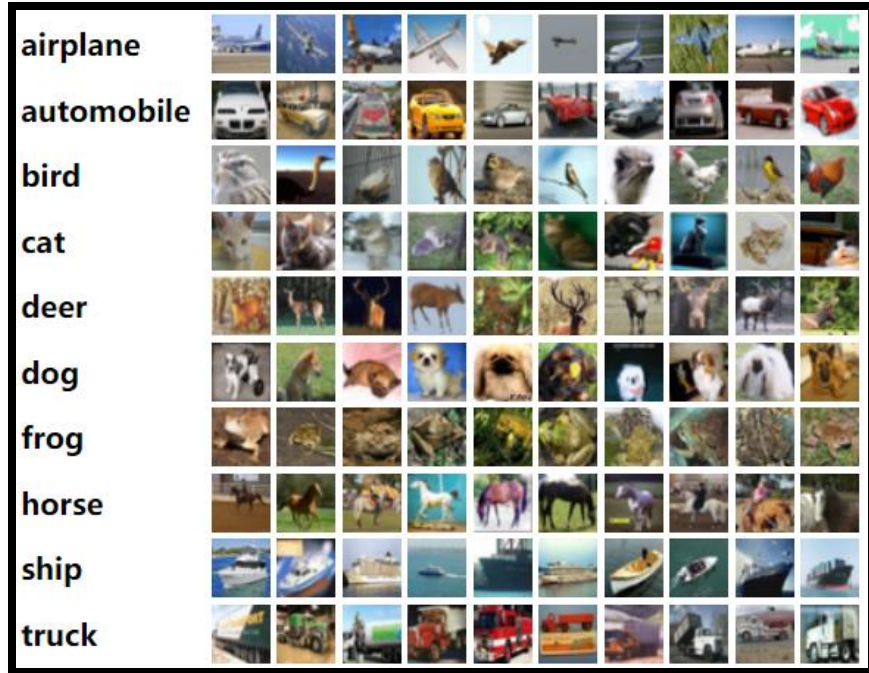
Theorie Artificiële Intelligentie

Het herkennen van een patroon in een afbeelding. Dus wat wij doen, als we kijken naar een afbeelding.... Dan zien wij hier het getal 5.



Theorie Artificiële Intelligentie

CIFAR-10

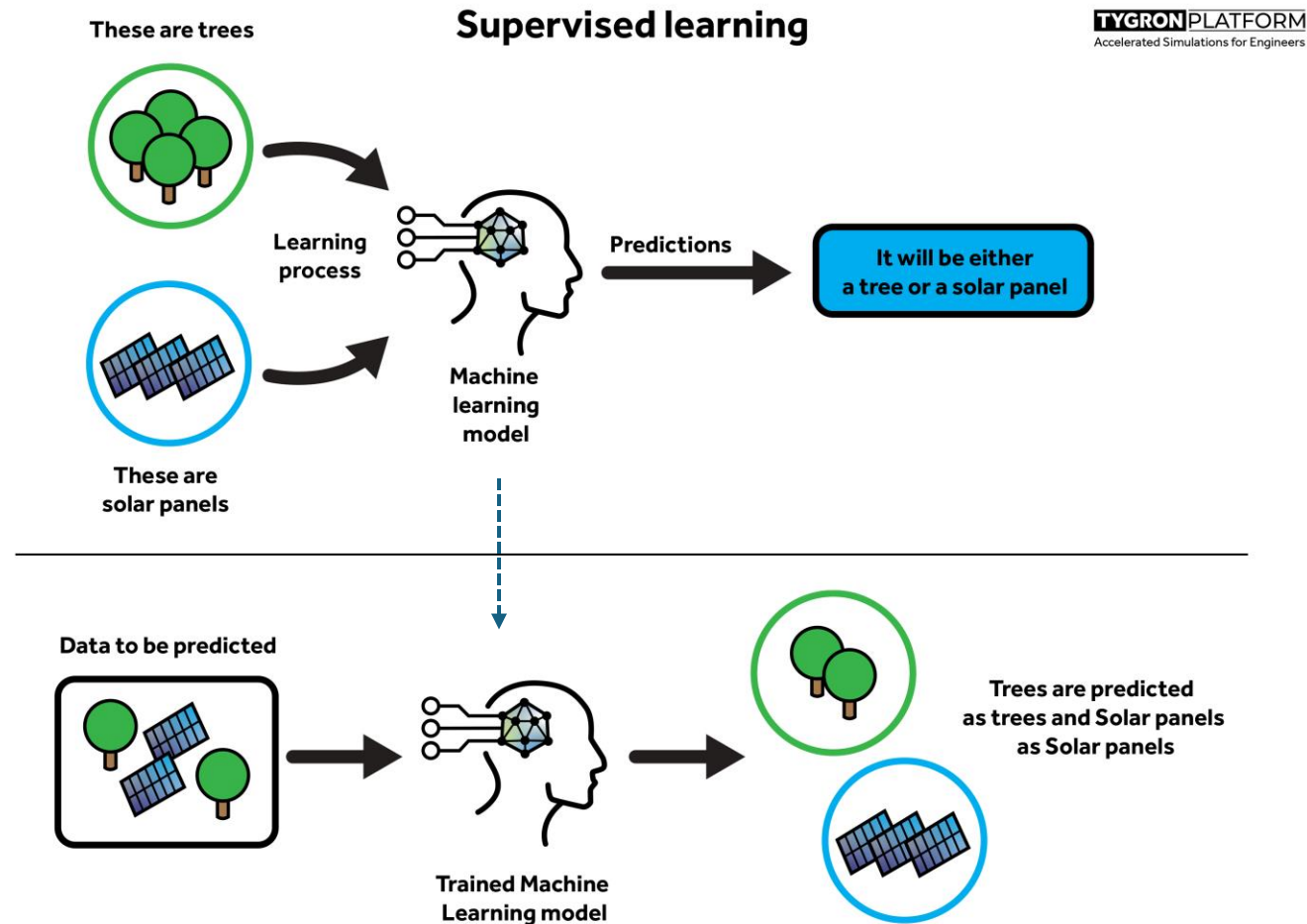


TYGRON PLATFORM
Accelerated Simulations for Engineers



Hugging Face (<https://huggingface.co/>)

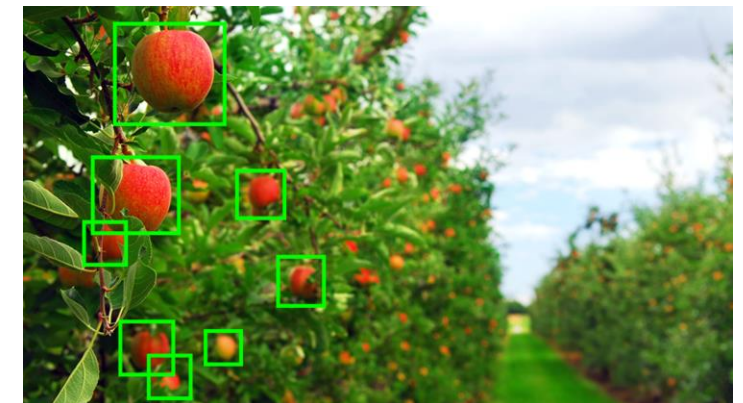
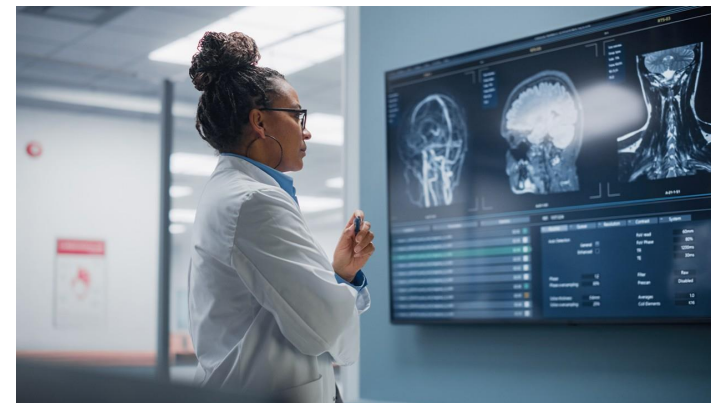
Theorie Artificiële Intelligentie



Theorie Artificiële Intelligentie

Toepassingen

- Voorraad beheer
- Gewasmonitoring
- Kwaliteitscontrole tijdens productie proces
- Atonoom rijden
- Medische beeldanalyse
- Tekstherkenning
- Sportanalyse voor het bijhouden van statistieken
-



Bron: <https://www.augmentedstartups.com/blog/unleash-your-inner-superhero-top-10-real-world-computer-vision-projects>

Wat kan de Tygron gebruiker hiermee?

Solution looking for a problem?



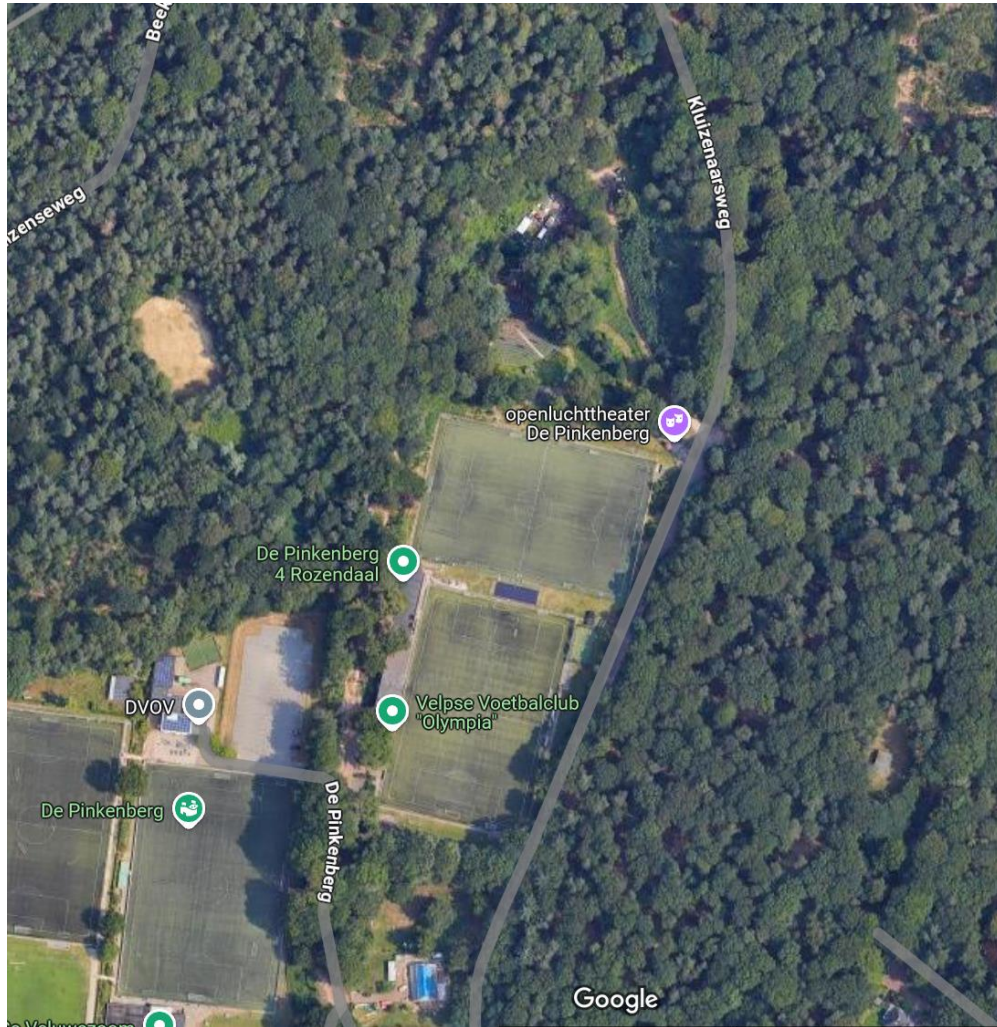
Bron: Dall-e

Zeker, hier is een afbeelding voor "Solution Looking For A Problem":

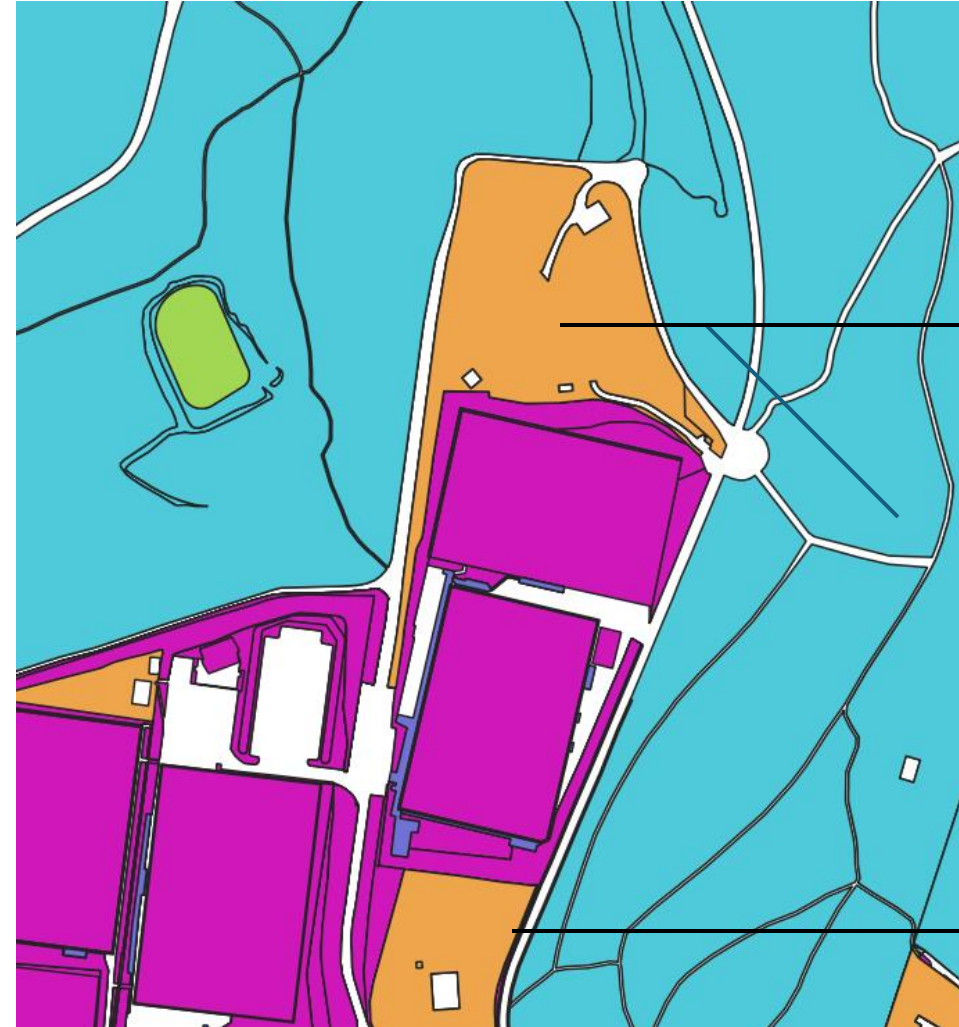


Bron: Gemini

Wat kan de Tygron gebruiker hiermee?

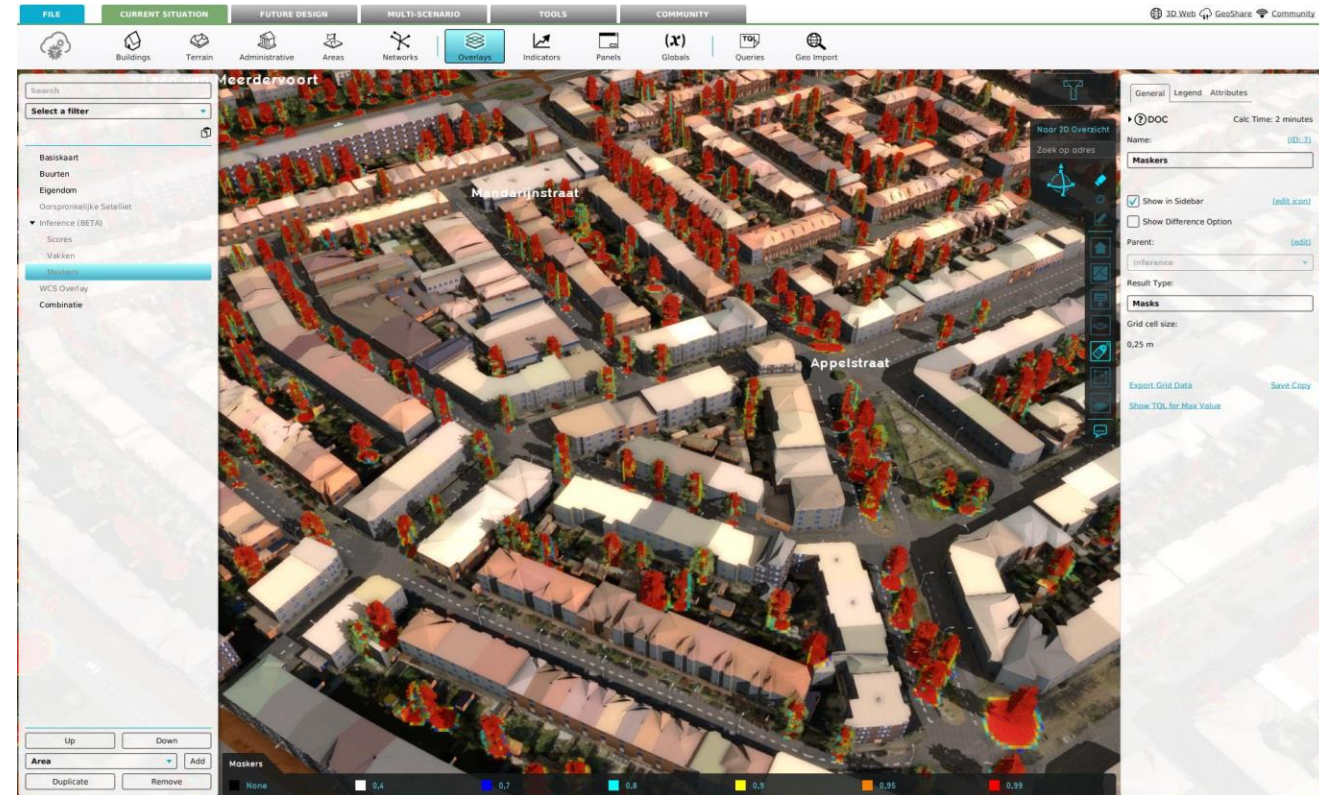


Bron: Google Maps



Bron: BGT onbegroeid terrein en begroeid terrein

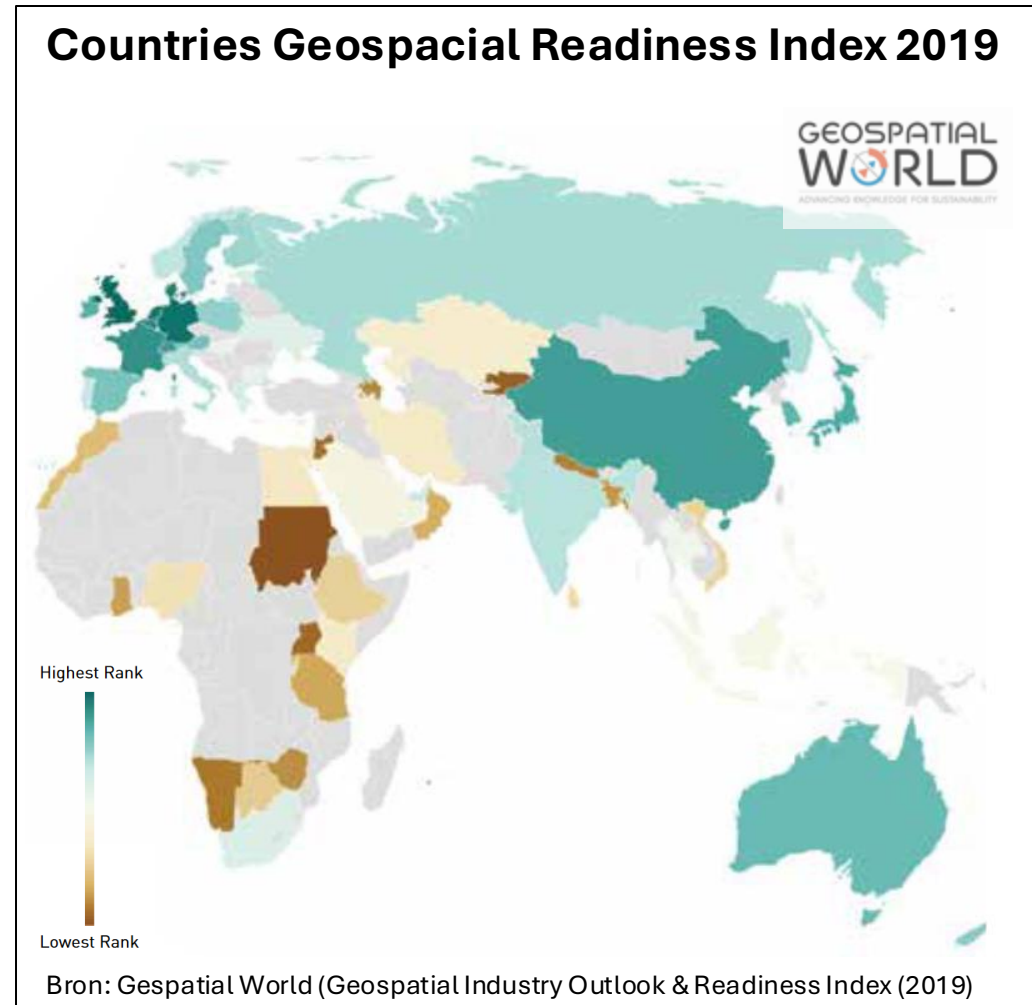
Wat kan de Tygron gebruiker hiermee?



Wat kan de Tygron gebruiker hiermee?

Herkennen van objecten in de openbare ruimte

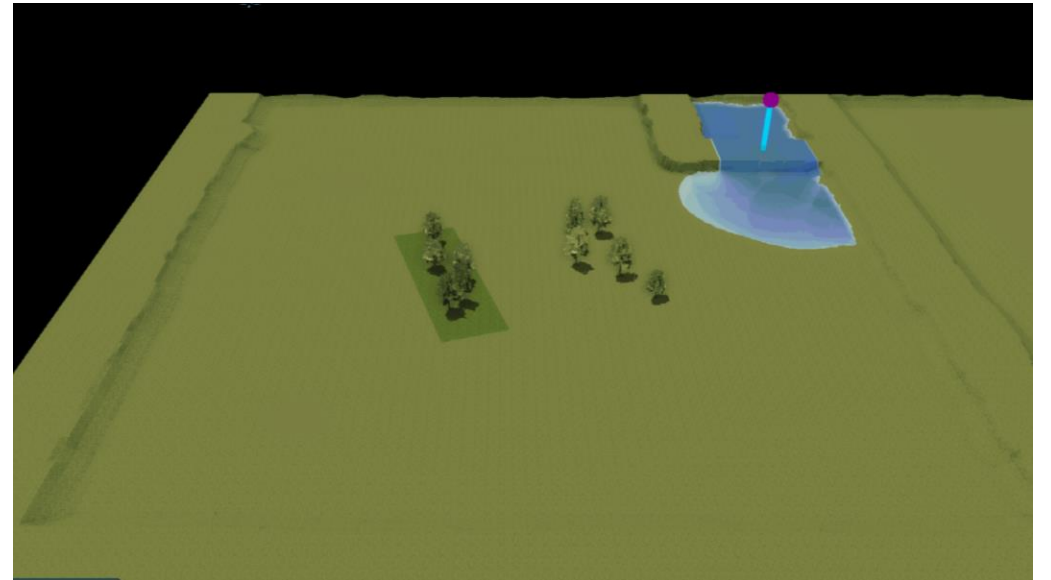
- Waar staan bomen? Wat zijn dat voor bomen?
- Wat is de grens van een heidegebied?
- Waar wordt een C-watergang onderbroken?
- Wat is de waterkwaliteit?
- Groeien er bomen hoger dan 5 meter op een dijk?
- Zijn er wellen zichtbaar langs een dijk?
- Weinig data beschikbaar over de inrichting van de openbare ruimte (buitenland)



Wat kan de Tygron gebruiker hiermee?

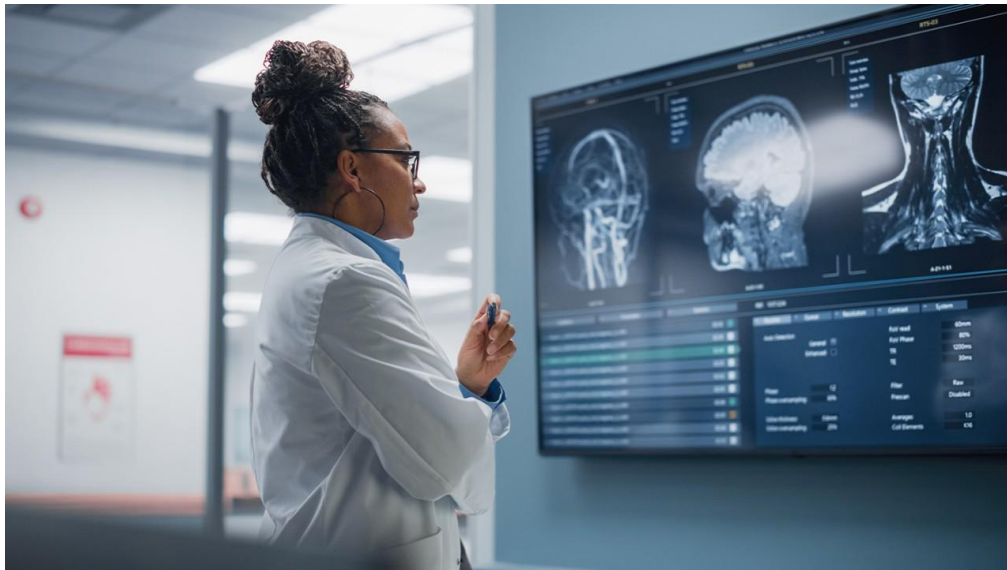
Betere inschatting van de parameters bij berekeningen.

- Infiltratiewaarden
- Evaporatiewaarden
- Manningwaarde
- Berging in microreliëf
- Oppervlakkige afstroming langs obstakels
- ...

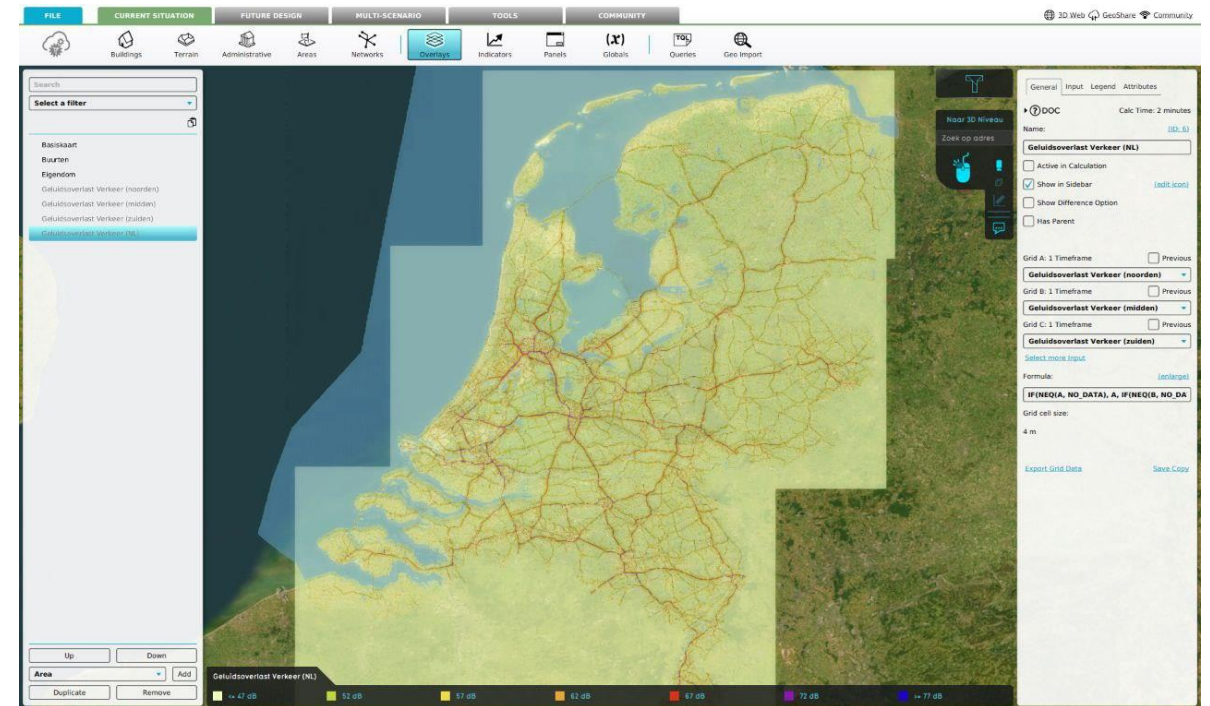


Wat kan de Tygron gebruiker hiermee?

- Resultaten van je berekening controleren.
 - Fouten in het model opzoeken – waar stroomt water waar je het niet zou verwachten?

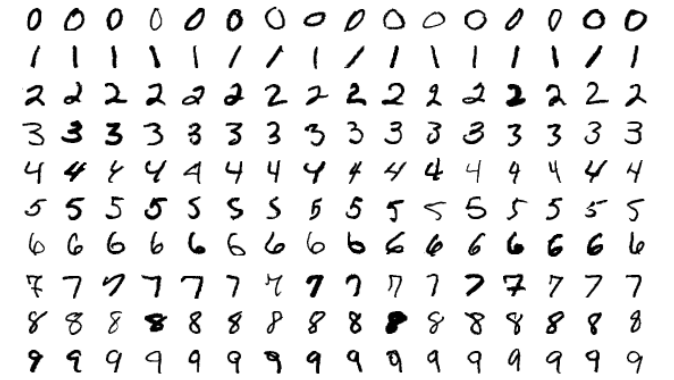


Bron: <https://www.augmentedstartups.com/blog/unleash-your-inner-superhero-top-10-real-world-computer-vision-projects>

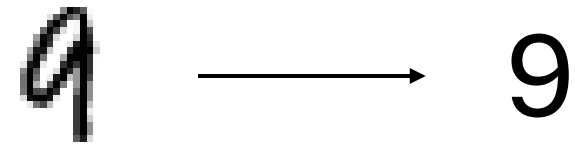


Bron: <https://simnl.tygron.com/web/3dmap.html?token=18468b5aMyZyZCz9gXRZJaQHfZz2qwud>

Zelf aan de slag



MNIST-dataset



```
1 # import tensorflow and keras (tf.keras not "vanilla" Keras)
import tensorflow as tf
from tensorflow import keras

2 # get data
(train_images, train_labels), (test_images, test_labels) = \
keras.datasets.mnist.load_data()

3 # setup model
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation=tf.nn.relu),
    keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer=tf.train.AdamOptimizer(),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

4 # train model
model.fit(train_images, train_labels, epochs=5)

5 # evaluate
test_loss, test_acc = model.evaluate(test_images, test_labels)
print('test accuracy:', test_acc)

6 # make predictions
predictions = model.predict(test_images)
```

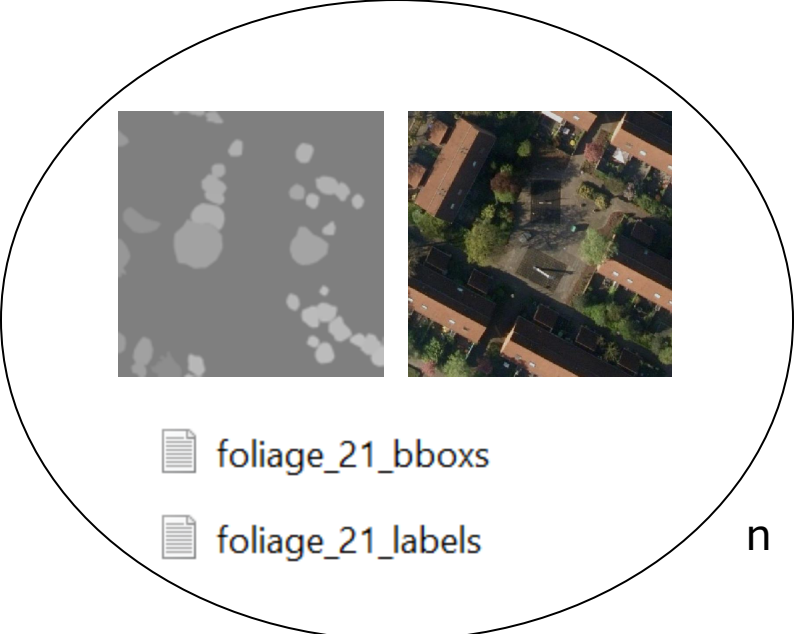

Zelf aan de slag

1 = boom



Zelf aan de slag

Dataset met gelabelde objecten



foliage_21_bboxes

foliage_21_labels

n



Python



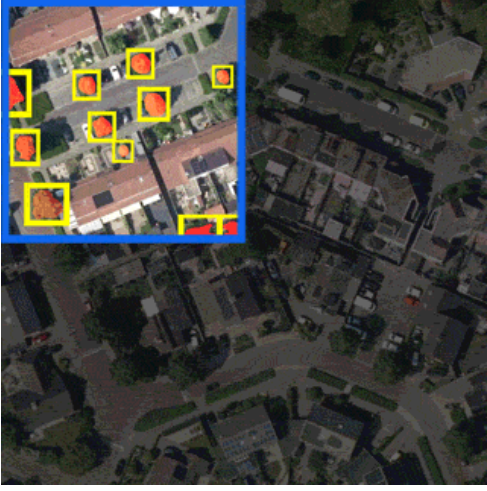
PyTorch



ONNX

Inference

TYGRON PLATFORM
Accelerated Simulations for Engineers



Tot slot



Feedback en vragen



Ward van Laatum

Custom Solutions and Support team

t. +31 6 86872006

e. Ward@Tygron.com