Deep Learning with Keras

Shubhabrata Roy, HERE Research Eindhoven
Pycon Ireland | November 06, 2016

Introduction to Deep Learning and Keras

1. What exactly is deep learning ?

2. Why is it generally better than other methods on certain types of data (on image, speech and some other types)?

1. What exactly is deep learning ?

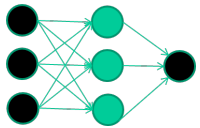
2. Why is it generally better than other methods on certain types of data (on image, speech and some other types)?

1. 'Deep Learning' implies using a neural network with several layers of nodes between input and output

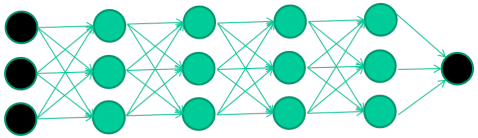
2. The series of layers between input & output perform feature identification and processing in a series of stages, just as our brains do.

Multilayer neural networks have been around for decades. What's actually new?

We have always had good algorithms for learning the weights in networks with 1 hidden layer



But these algorithms are not good at learning the weights for networks with more hidden layers

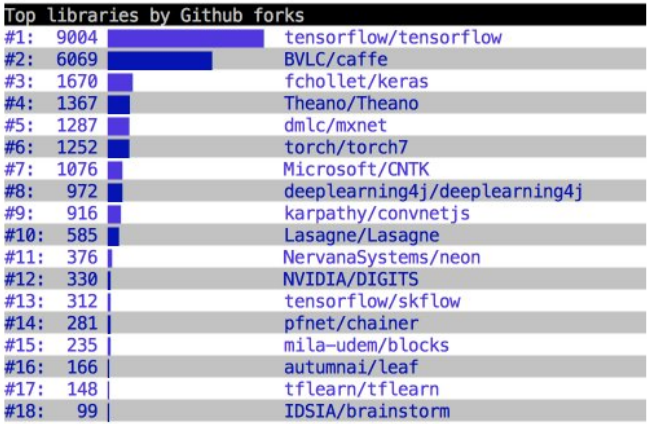


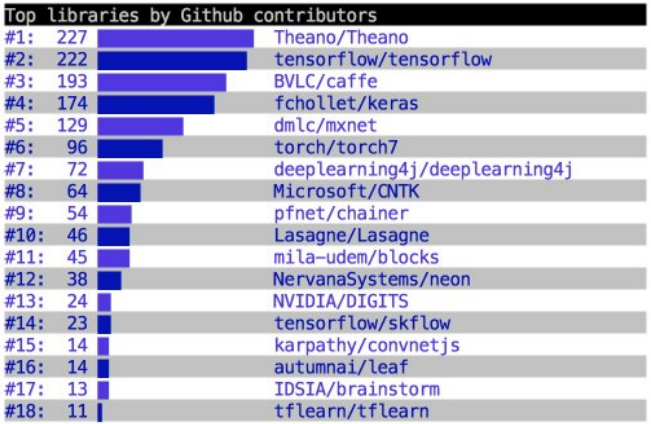
What's new is: **algorithms for training many-layer networks**

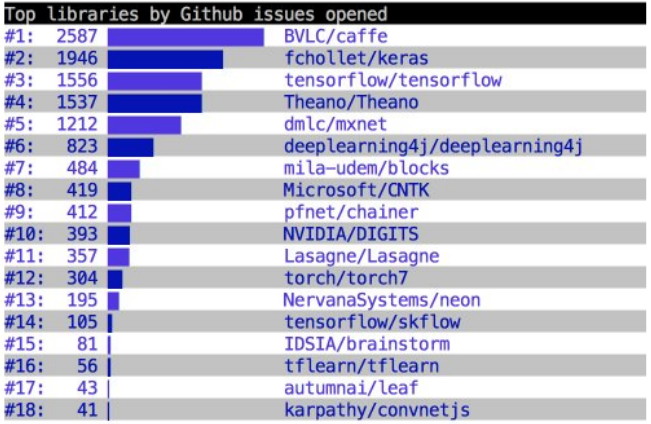
Introduction to Keras

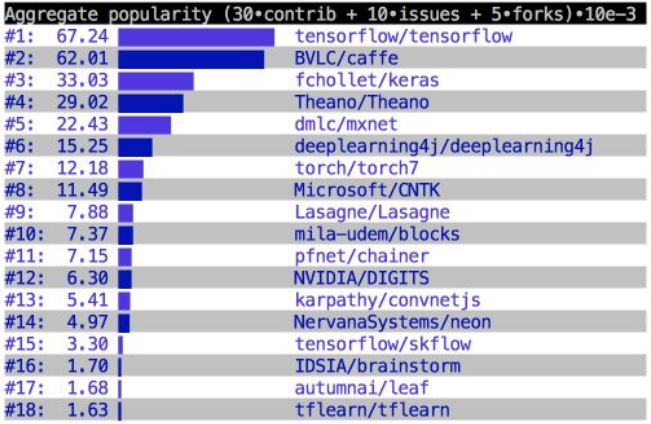
- Developed by Francois Chollet from Google
- A minimalist Python library for deep learning
- Can run on top of Theano or TensorFlow
- Possible to quickly develop and test models
- Runs on both Python 2.7 and 3.5
- Can seamlessly execute on GPUs and CPUs

A bit more about Keras







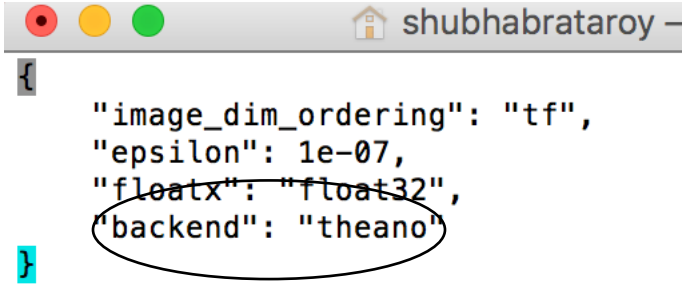


Source: https://twitter.com/fchollet/media

Theano and TensorFlow Backends

- A lightweight API
- Provides a consistent interface to efficient numerical libraries called backends

- Easiest way to choose the backend is editing ~/.keras/keras.json



Building deep learning models with Keras

- **Define the model:** Create a Sequential model and add configured layers
- **Compile the model:** Specify loss function and optimizers and call the compile() function on the model
- **Fit the model:** Train the model on a sample of data by calling the fit() function on the model
- **Make predictions:** Use the model to generate predictions on new data by calling functions such as evaluate() or predict() on the model

Available layers in Keras

Regular Dense/ Multi Layer Perceptron

```
keras.layers.core.Dense(output_dim,
                        init='glorot_uniform',
                        activation='linear',
                        weights=None,
                        W_regularizer=None, b_regularizer=None, activity_regularizer=None,
                        W_constraint=None, b_constraint=None,
                        input_dim=None)
```

Recurrent layer/ LSTM/ GRU etc.

```
keras.layers.recurrent.GRU(output_dim,
        init='glorot_uniform', inner_init='orthogonal',
        activation='sigmoid', inner_activation='hard_sigmoid',
        return_sequences=False,
        go_backwards=False,
        stateful=False,
        input_dim=None, input_length=None)
```

More layers

1-D Convolutional Layers

```
keras.layers.convolutional.Convolution1D(nb_filter, filter_length,
        init='uniform',
        activation='linear',
        weights=None,
        border_mode='valid',
        subsample_length=1,
        W_regularizer=None, b_regularizer=None,
        W_constraint=None, b_constraint=None,
        input_dim=None, input_length=None)
```

2-D Convolutional Layers

```
keras.layers.convolutional.Convolution2D(nb_filter, nb_row, nb_col,
        init='glorot_uniform',
        activation='linear',
        weights=None,
        border_mode='valid',
        subsample=(1, 1),
        W_regularizer=None, b_regularizer=None,
        W_constraint=None,
        dim_ordering='th')
```

Other types of layer include:

- Dropout
- Noise
- Pooling
- Normalization
- Embedding

Activations

- Almost all major activations are available: Sigmoid, tanh, ReLu, softplus, hard sigmoid, linear
- Advanced activations implemented as a layer (after desired neural layer)
- Advanced activations: LeakyReLu, PReLu, ELU, Parametric Softplus, Thresholded linear and Thresholded Relu

Objectives and Optimizers

Objective Functions:

- Error loss: rmse, mse, mae, mape, msle
- Hinge loss: squared hinge, hinge
- Class loss: binary crossentropy, categorical crossentropy

Optimization:

- Provides SGD, Adagrad, Adadelta, Rmsprop and Adam
- All optimizers can be customized via parameters

A brief introduction to ”HERE”

HERE in numbers

200 Countries mapped

7,000+
Employees in 56 countries focused on delivering the world's best map and location services

Millions of changes made to the map **every day**

Today, HERE provides services to nearly all global OEM brands as well as many market leading brands such Microsoft, Amazon, Facebook etc.

30+
Years of experience transforming mapping technology

4 of 5
In-car navigation systems in Europe and N America use HERE maps

We are making sense of the world through the lens of location.

We help people achieve better outcomes –
whether it's a driver moving to their destination safely,
a municipality managing its infrastructure smartly or
a business optimizing the utility of its assets.

here

A history of transforming potential into meaningful products

1985

Navigation Technologies formed

1994

1st map for in-car nav

1st map for Web (1995)

2004

1st map for ADAS

1st map for phone

1st map for Adaptive Cruise Control (2005)

2007

Community mapping

Offline maps for mobile

2009

1st map for Predictive Cruise Control

High precision data collection and map-building technology (2010)

Use of sensor data for map building

2011

1st pure location cloud

2014

Augmented Reality technology

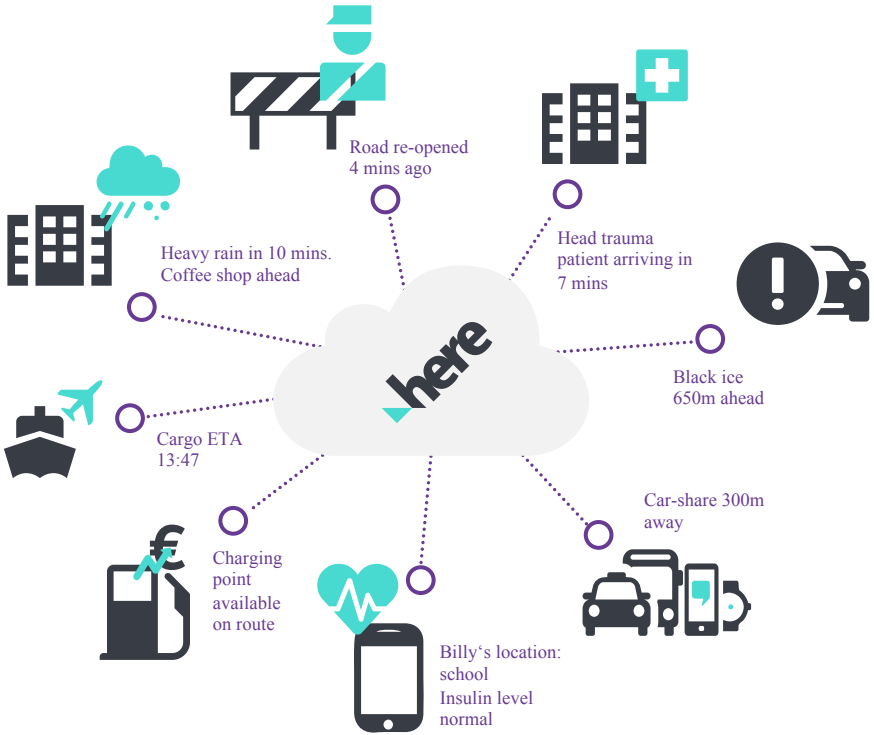
2015

HERE gets new investors: Audi, BMW, Daimler

First commercial cloud service for automated vehicles

HERE **Open Location Platform**

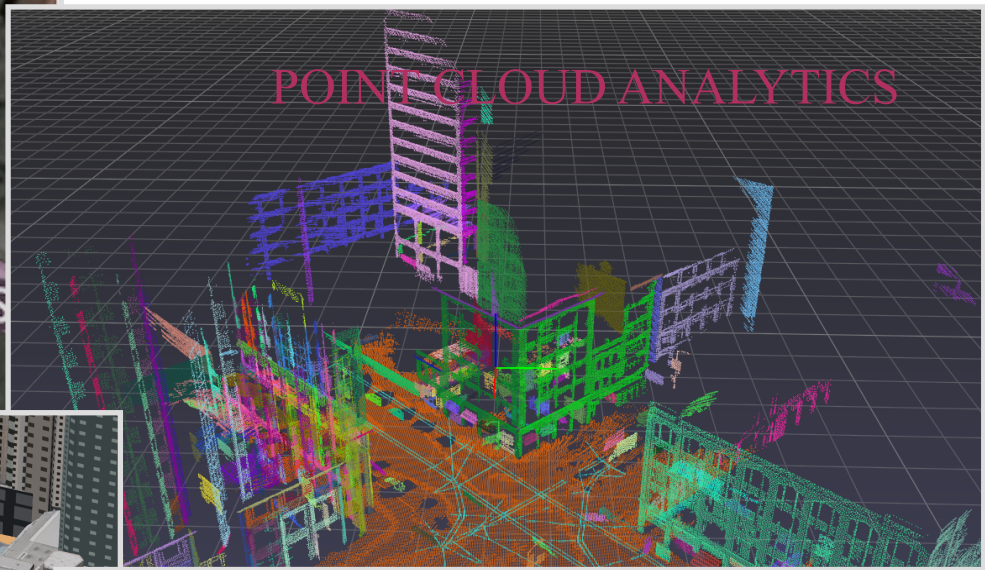
The power of location



Research @ HERE

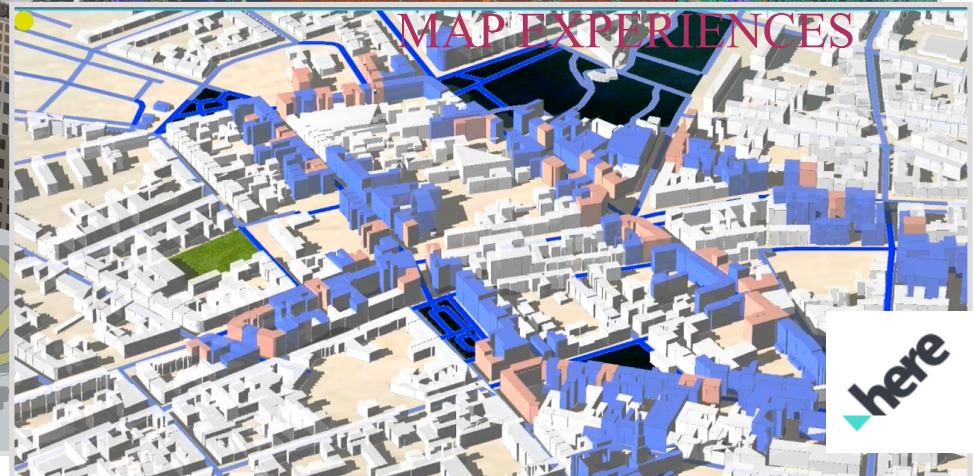

here

SPATIAL TRAJECTORY ANALYTICS

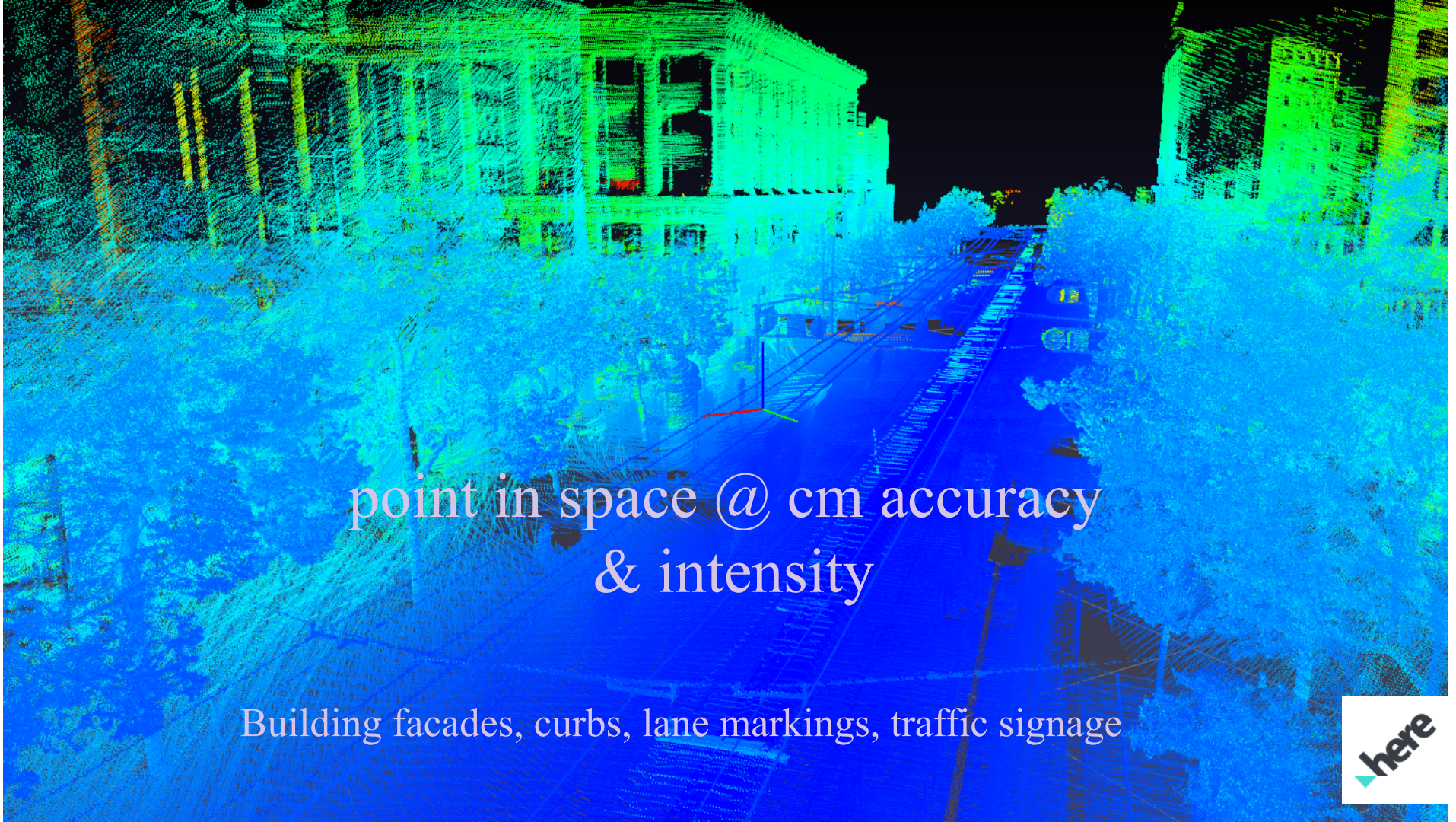


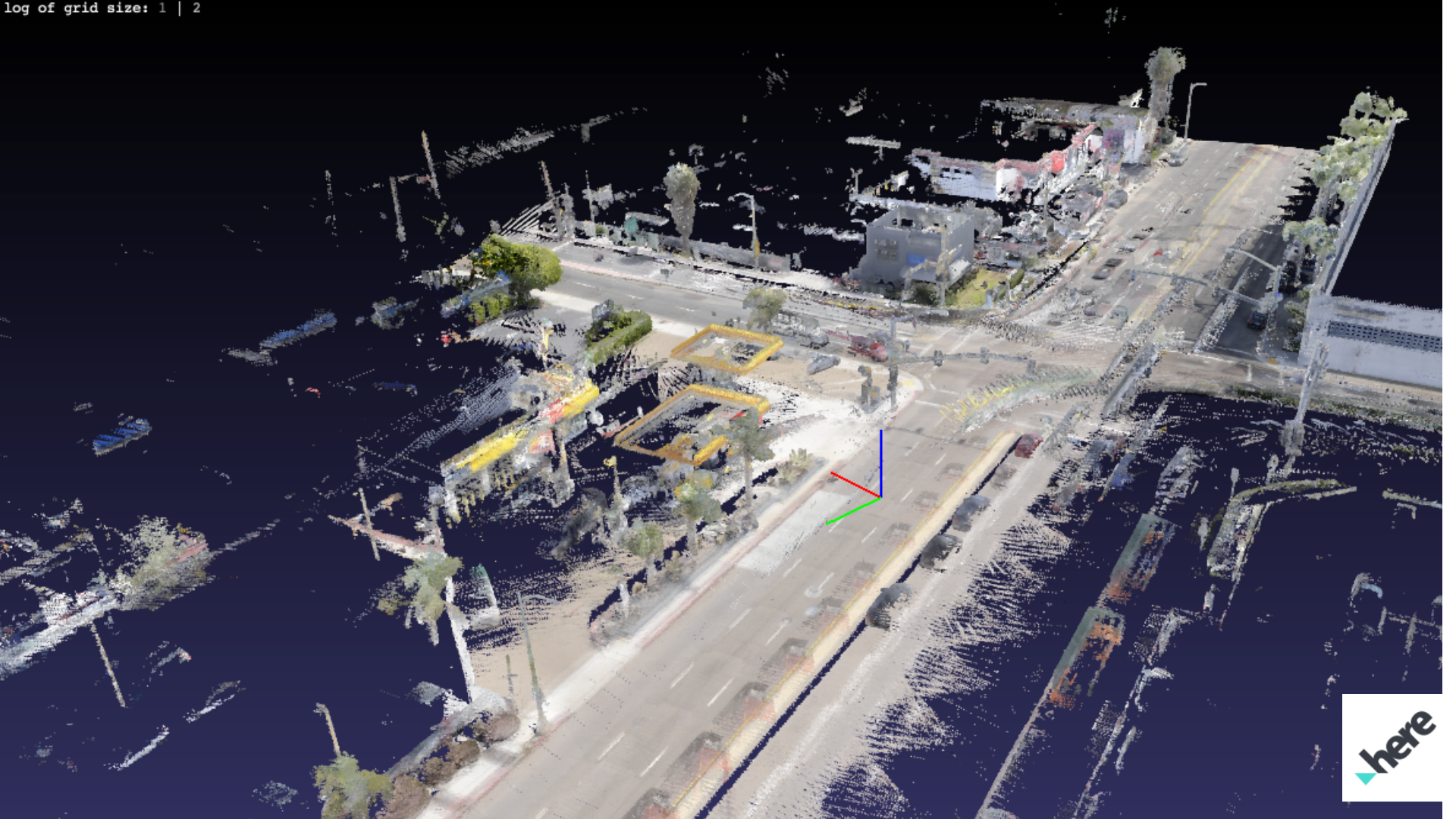
POINT CLOUD ANALYTICS

NEXT GENERATION MAP



MAP EXPERIENCES


point in space @ cm accuracy
& intensity
Building facades, curbs, lane markings, traffic signage
here

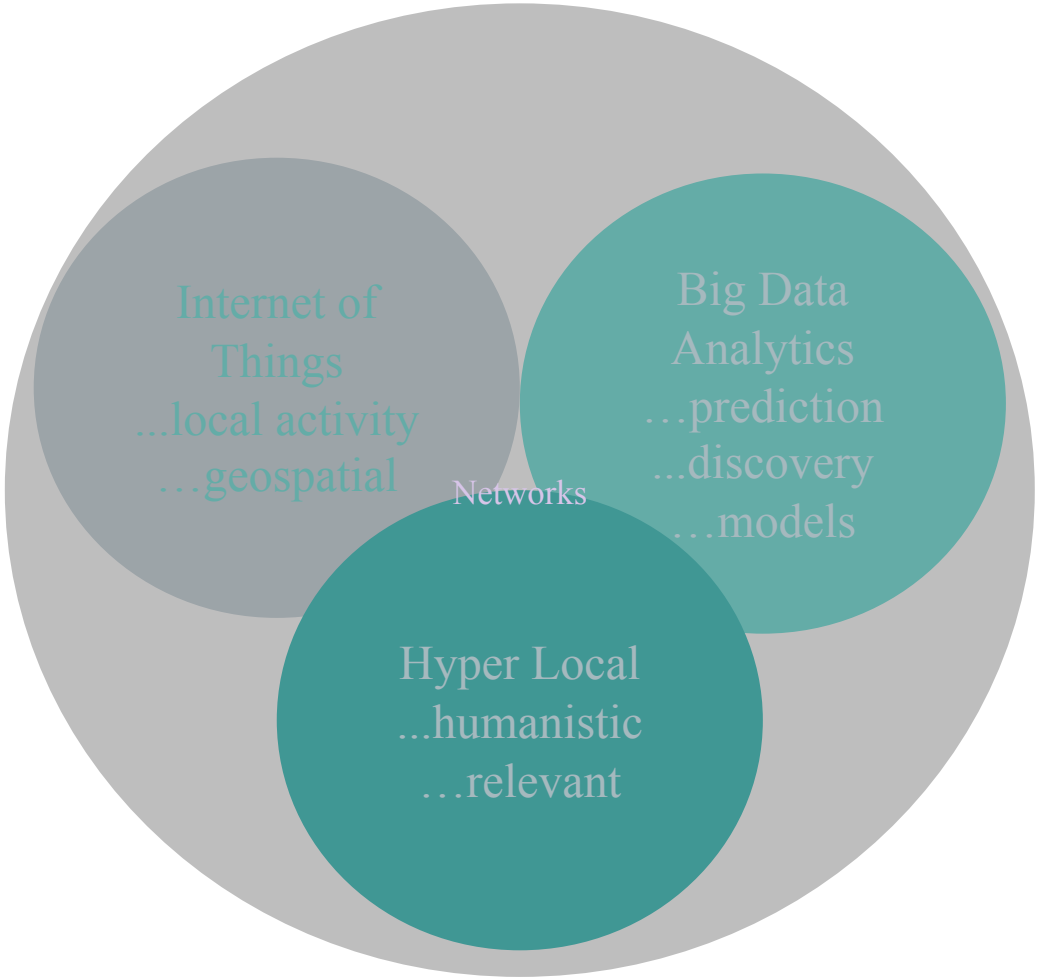

log of grid size: 1 | 2


Pervasive, data-driven,
Cloud-enabled
‘Internet of Things’.

We are using multiple devices everyday

Machine generated
Human generated
Structured
Unstructured

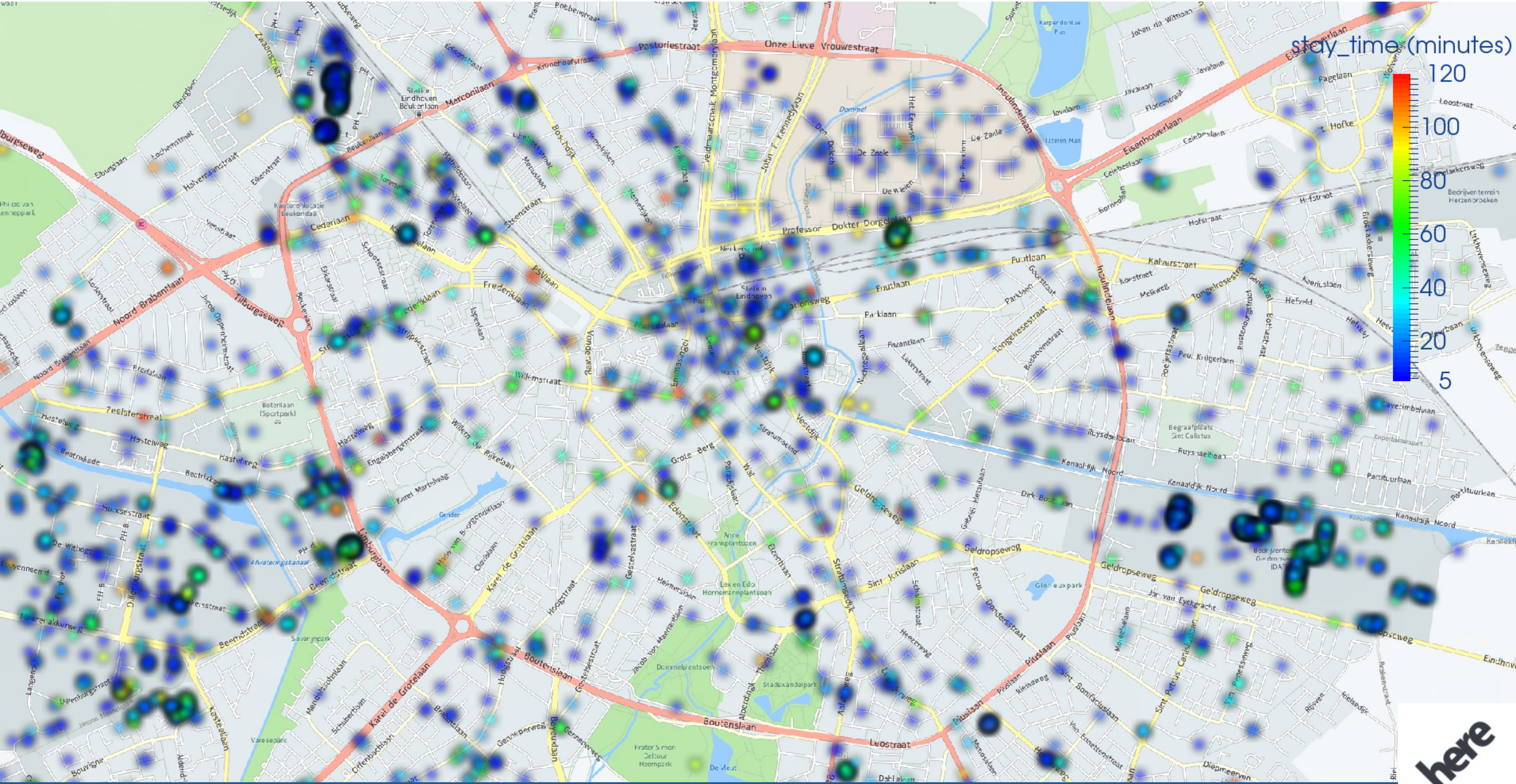
Velocity
Volume
Variety
Variability
Veracity
Visualization
Value $$$$


Internet of
Things
...local activity
…geospatial
Big Data
Analytics
…prediction
...discovery
…models
Networks
Hyper Local
...humanistic
…relevant

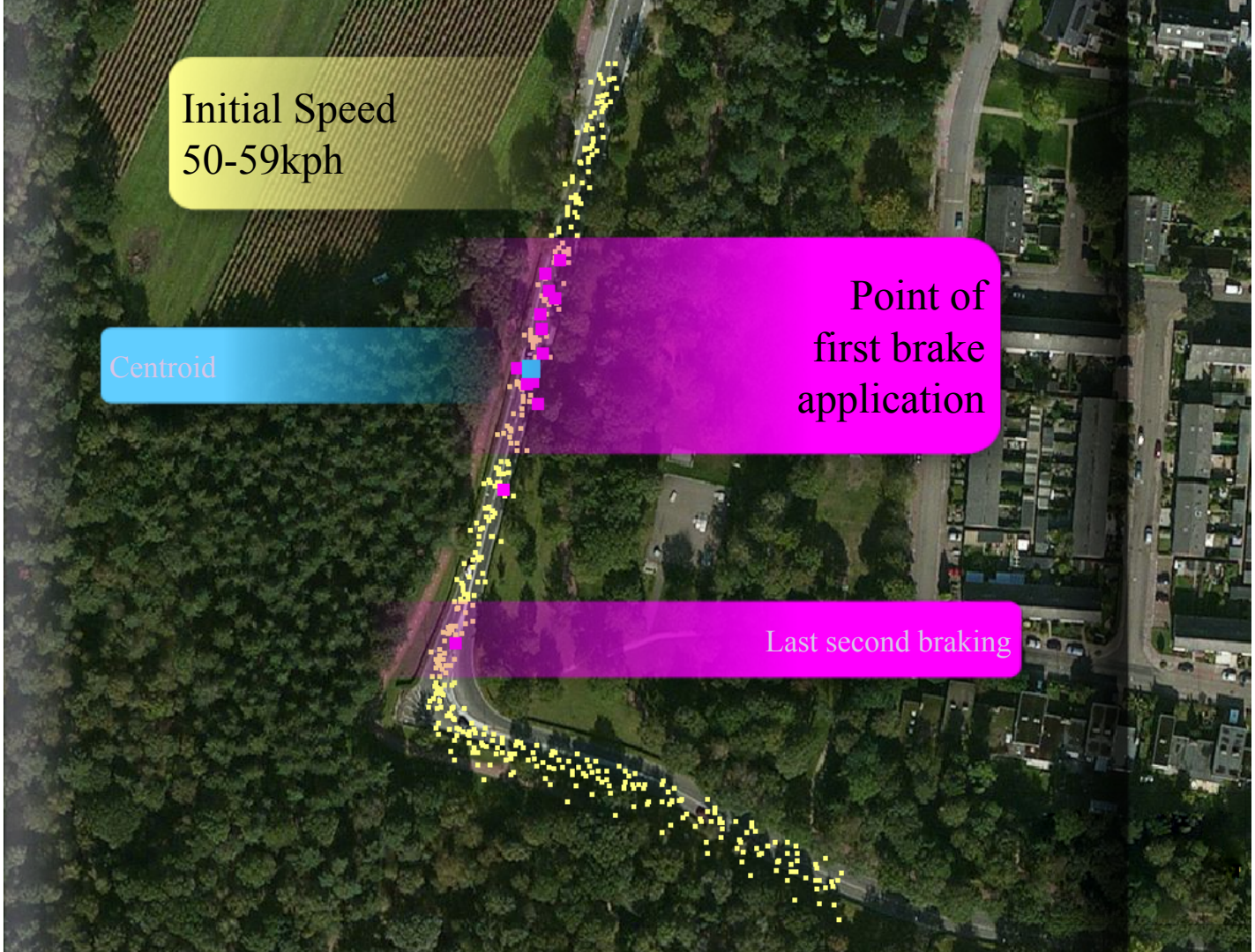
Maps are morphing into something new…

…companions, advisors
….assistants

….devices are listening
…reporting, tracking



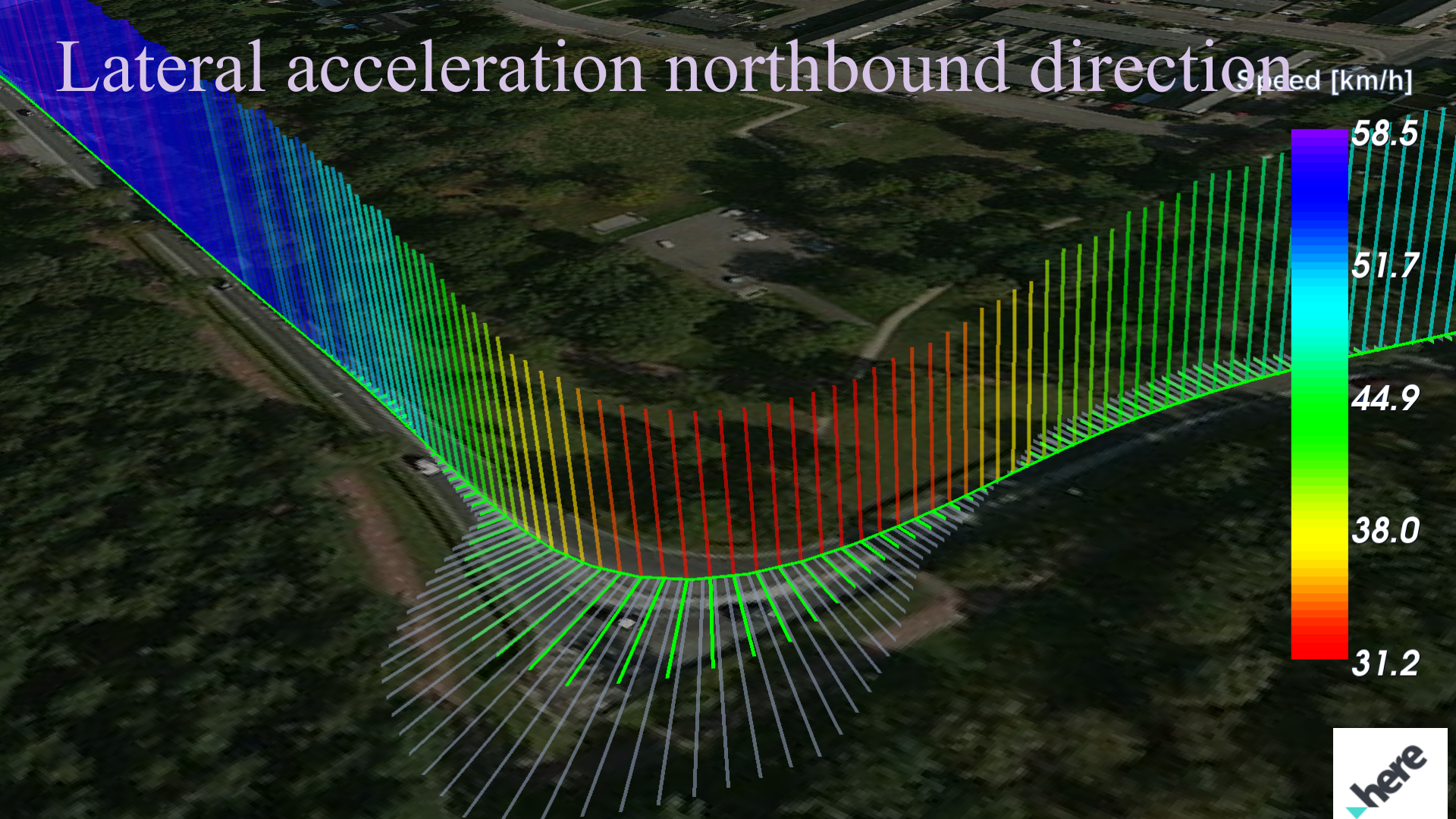
HERE Research


Initial Speed
50-59kph
Point of
first brake
application
Centroid
Last second braking

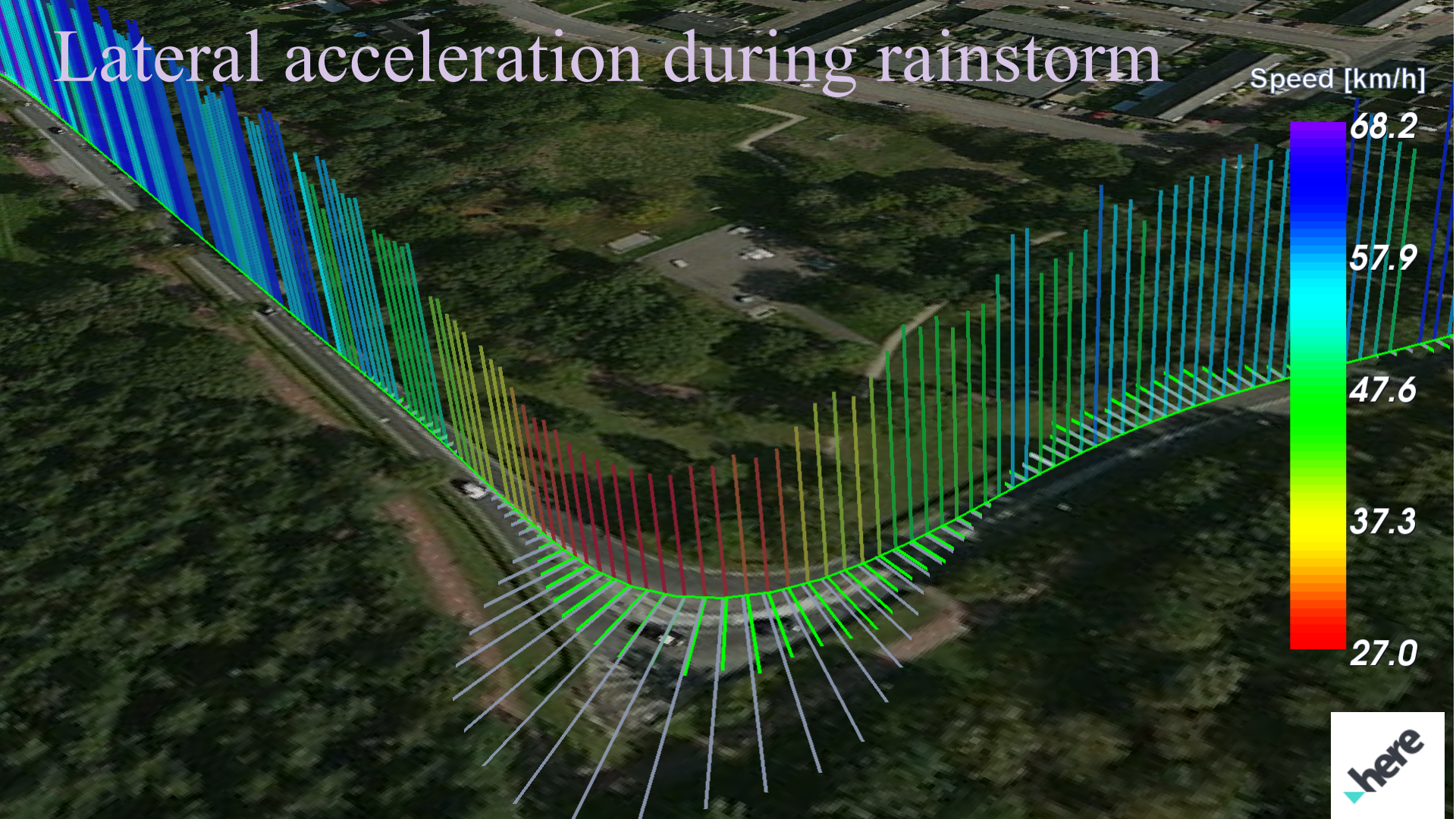

Normal Behavior
First Brake Location
SPEED (km/h)
70
60
40
20


SPEED (km/h)
70
60
40
20
Abnormal Behavior
First Brake
Location
here

Lateral acceleration northbound direction



Lateral acceleration during rainstorm



Use Cases with Deep Learning

Image classification

Objective:
Vehicle type identification based on images

Application:
Object identification for autonomous vehicles

Framework:

Image_Recognition_with_Keras

Sequence Prediction

Objective:

Predict number of rental bike over a time window

Application:

- Managing resources (bikes) to meet future demands
- Update points of interest of a map when someone is looking to rent a bike (e.g. likely to run out of available bikes in an hour)

Framework:

Sequence_Prediction_with_Keras


here