

Målrettet tilsyn med råstofgrave - et pilotprojekt med Machine Learning

Specialkonsulent Allan Køtter
Region Hovedstaden

ATV-møde "Perspektivering af digitalisering af data"
den 18. juni 2019

Formål

- Afsøge muligheder for at inddrage ML i vores arbejde med data
 - Adgang til stadig flere datasæt
 - Adgang til avancerede analysemetoder
 - Stigende efterspørgsel på dataunderstøttelse



Projektidé

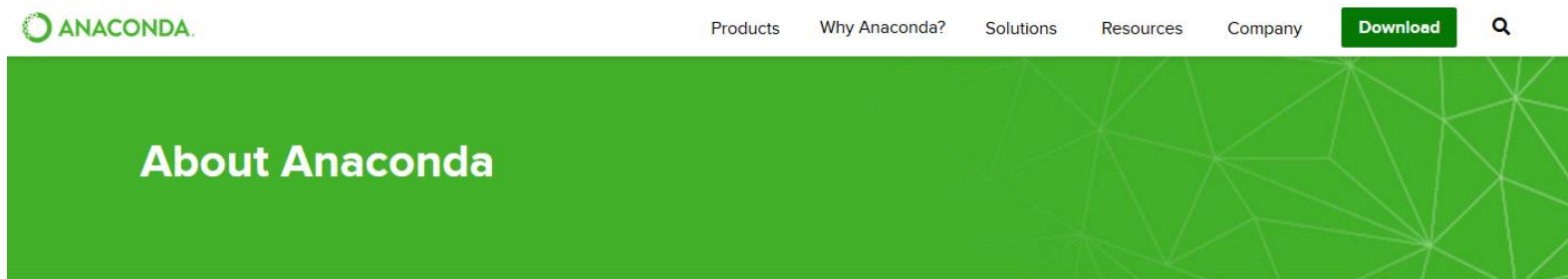
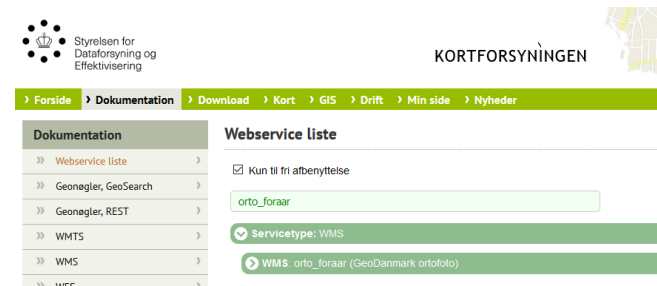


Projektidé



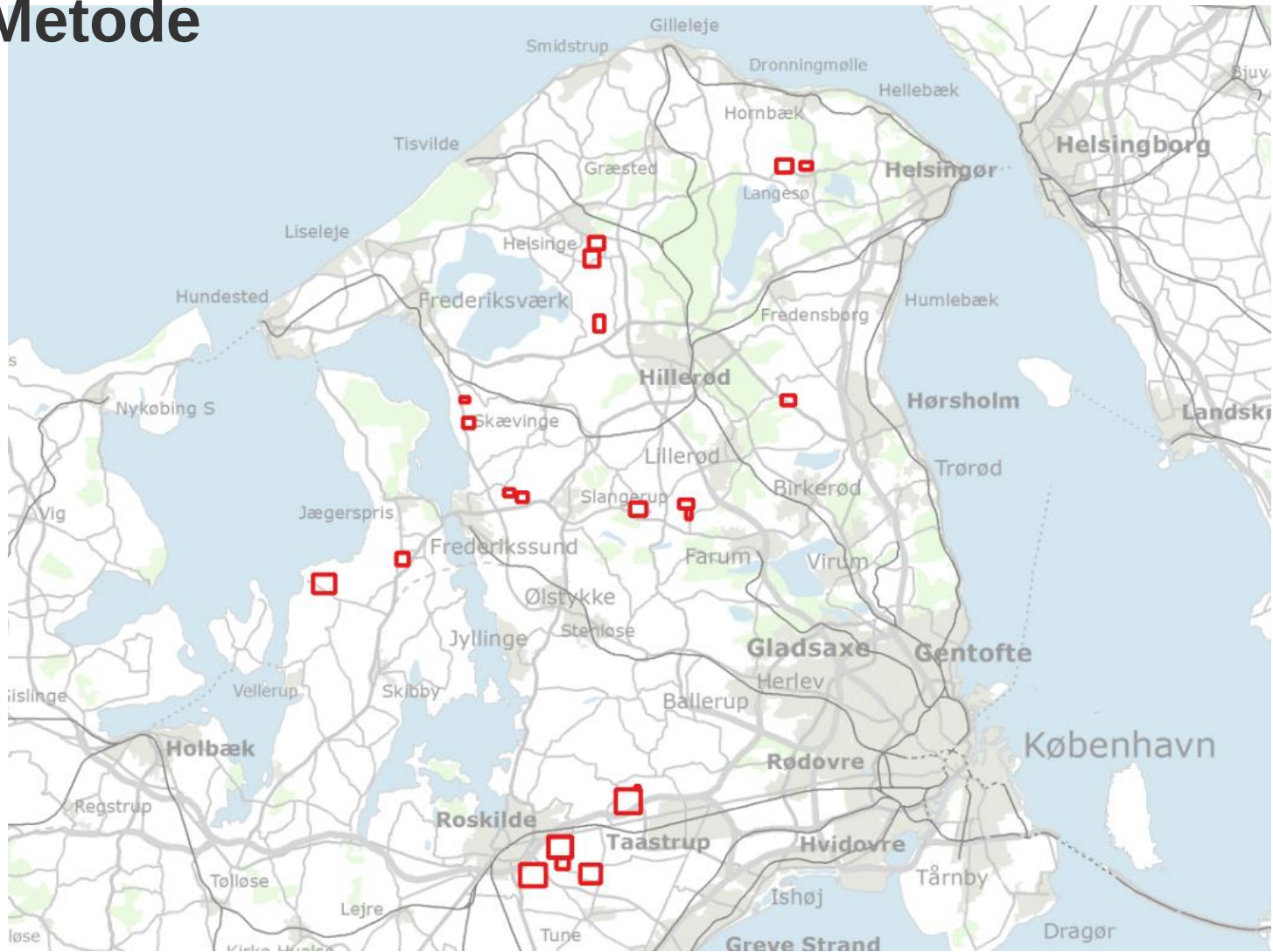
Datakilder og Værktøj

- Kortforsyningens WMS
 - Ortofoto forår (+ NIR)
 - Højdemodel
- Værktøj: Anaconda

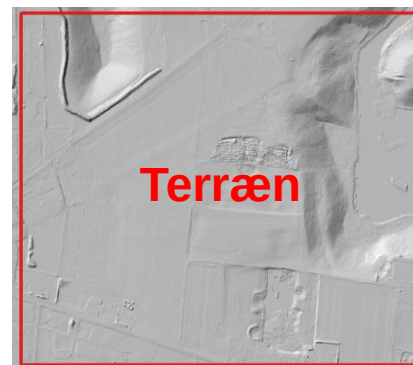
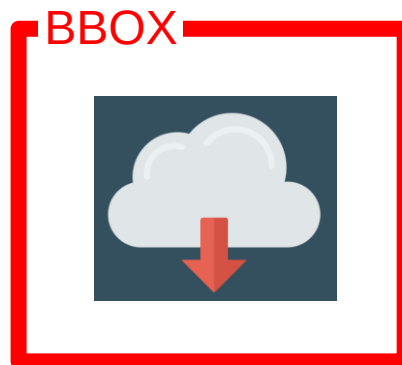


With more than 13 million users, Anaconda is the world's most popular data science platform and the foundation of modern machine learning. Anaconda Enterprise delivers data science and machine learning at speed and scale, unleashing the full potential of our customers' data science and machine learning initiatives.

Metode



Metode



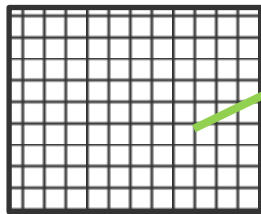
Metode



Pixelbaseret feature extraction

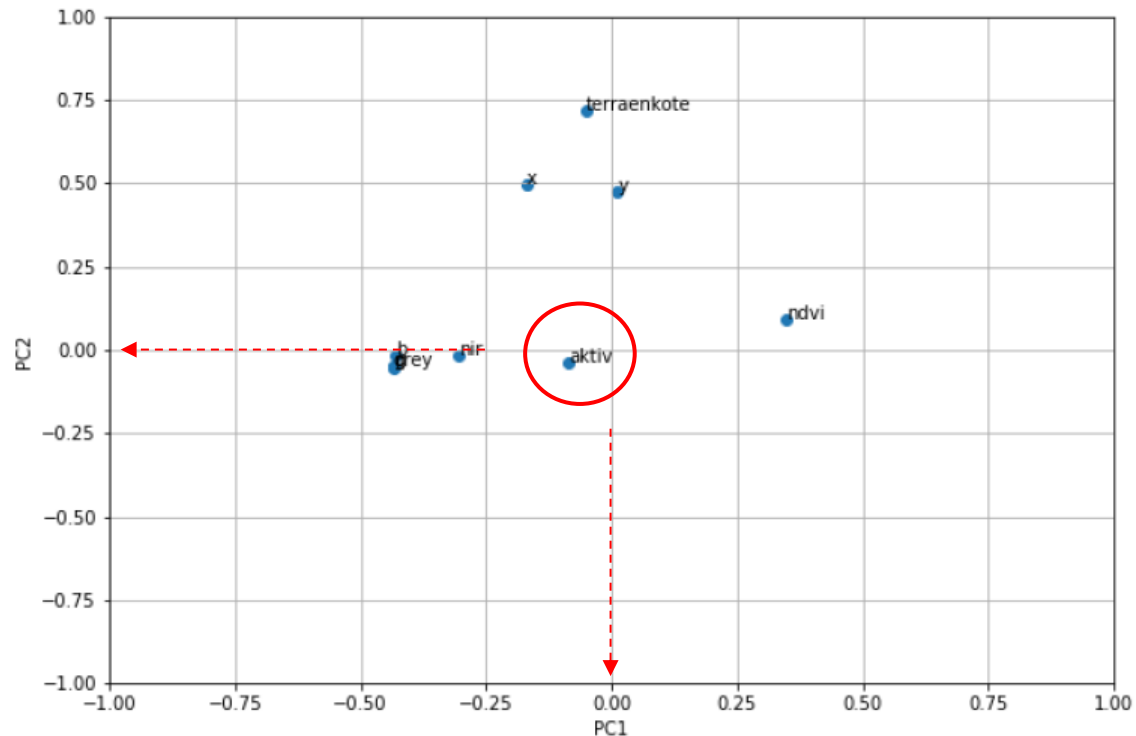


pixel_nr	r_index	c_index	x	y	r	g	b	grey	nir	ndvi	terraenkote	aktiv
0	0	0	697774,5	6169460	127	123	103	125	155	1,08	56,03	0
1	0	1	697775,5	6169460	128	123	100	125	152	1,00	56,03	0
2	0	2	697776,5	6169460	131	126	103	128	158	0,82	56,01	0
3	0	3	697777,5	6169460	121	120	98	120	158	1,61	56,01	0
4	0	4	697778,5	6169460	114	113	89	113	155	3,15	56	0
5	0	5	697779,5	6169460	115	116	96	116	160	2,37	55,96	0
6	0	6	697780,5	6169460	116	117	94	117	160	2,20	55,96	0
7	0	7	697781,5	6169460	115	116	98	116	160	2,37	55,92	0
8	0	8	697782,5	6169460	115	118	98	117	160	2,37	55,87	0
9	0	9	697783,5	6169460	116	118	94	117	160	2,20	55,87	0
10	0	10	697784,5	6169460	116	117	94	117	162	2,09	55,84	1
11	0	11	697785,5	6169460	123	122	98	122	164	1,32	55,84	1
12	0	12	697786,5	6169460	119	120	98	120	165	1,64	55,84	1
...												



R, G, B
+ NIR, Grå, Kote, NDVI, aktiv

PCA – Primary Component Analysis

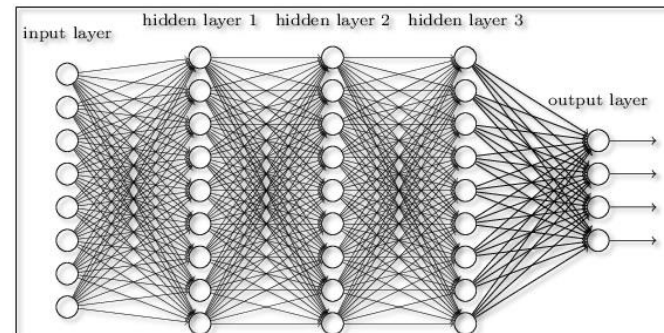


Konklusion på eksplorativ analyse

**Pixelbaseret klassifikation er ikke brugbare
til højopløselige ortofoto!**

→ Objektbaserede metoder

Neurale Netværk



- Har vist sig effektive til klassifikation af højopløselige billeder
- God erfaring med anvendelse til Remote Sensing
- Hastig udvikling inden for de seneste år
 - R-CNN (Girshick et. al. 2014)

Mask R-CNN

Kaiming He Georgia Gkioxari Piotr Dollár Ross Girshick

Facebook AI Research (FAIR)

Abstract

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding-box object detection, and person keypoint detection. Without bells and whistles, Mask R-CNN outperforms all ex-

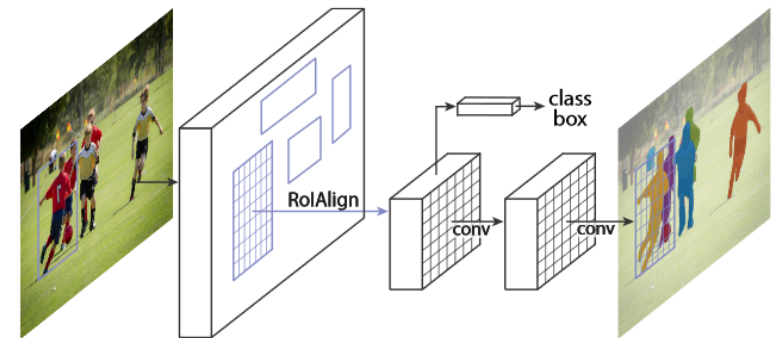


Figure 1. The **Mask R-CNN** framework for instance segmentation.

segmentation, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.¹ Given this, one might expect a complex method is required to achieve good results. However, we show that a surprisingly simple, flexible, and fast system can surpass prior state of the art instance segmentation results.

Region Hovedstaden Center for Regional Udvikling

 [matterport](#) / [Mask_RCNN](#)

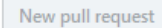
 Watch 535  Star 12,484  Fork 5,514

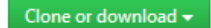
 Code  Issues 897  Pull requests 49  Projects 0  Wiki  Security  Insights

Mask R-CNN for object detection and instance segmentation **on Keras and TensorFlow**

[mask-rcnn](#) [tensorflow](#) [object-detection](#) [instance-segmentation](#) [keras](#)

 203 commits  1 branch  3 releases  40 contributors  View license

Branch: **master** 

 cclauss and waleedka import logging for line 382 	Latest commit 3deaec5 on 9 Mar
 assets	Link to new projects in README 9 months ago
 images	Initial commit 2 years ago
 mrcnn	import logging for line 382 3 months ago
 samples	Fix typos in inspect_data notebook(feature meap -> feature map) 3 months ago
 .gitignore	Update .gitignore last year
 LICENSE	Initial commit 2 years ago
 MANIFEST.in	add setup metadata last year
 README.md	Reversed order (1) and (2) in installation 4 months ago
 requirements.txt	add requirements last year
 setup.cfg	add setup metadata last year
 setup.py	Text updates in setup.py and README. last year

Forsøg med Mask R-CNN

<https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46>

Splash of Color: Instance Segmentation with Mask R-CNN and TensorFlow

Explained by building a color splash filter



Waleed Abdulla

[Follow](#)

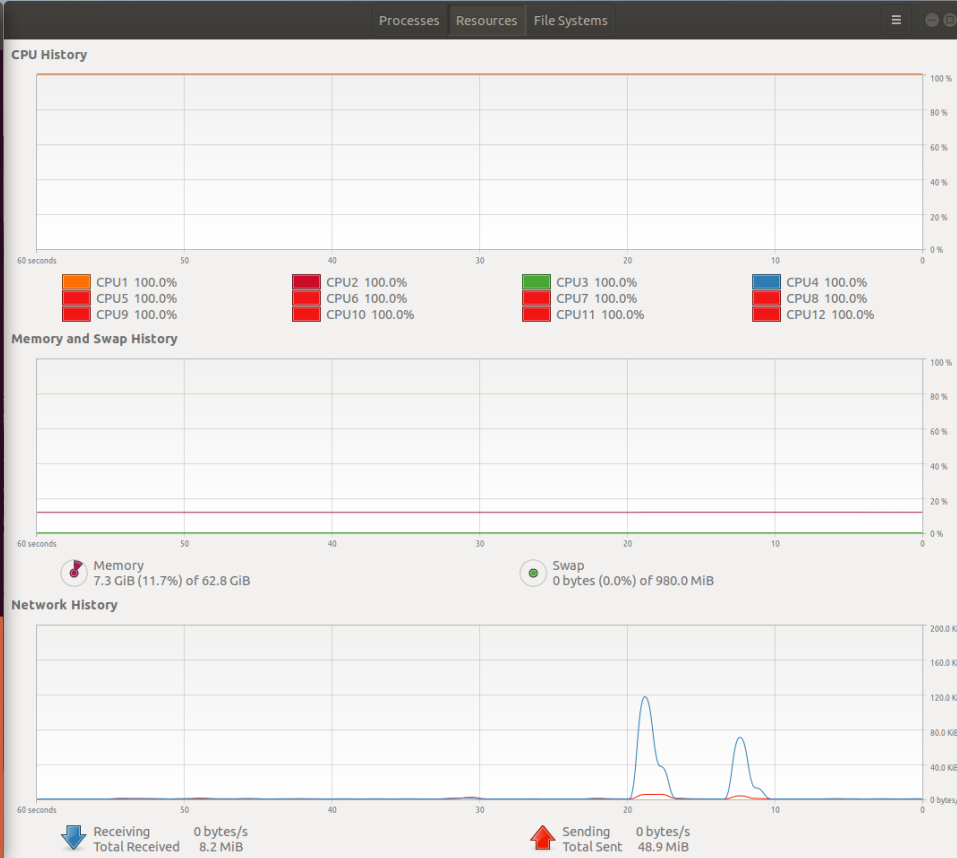
Mar 19, 2018 · 12 min read



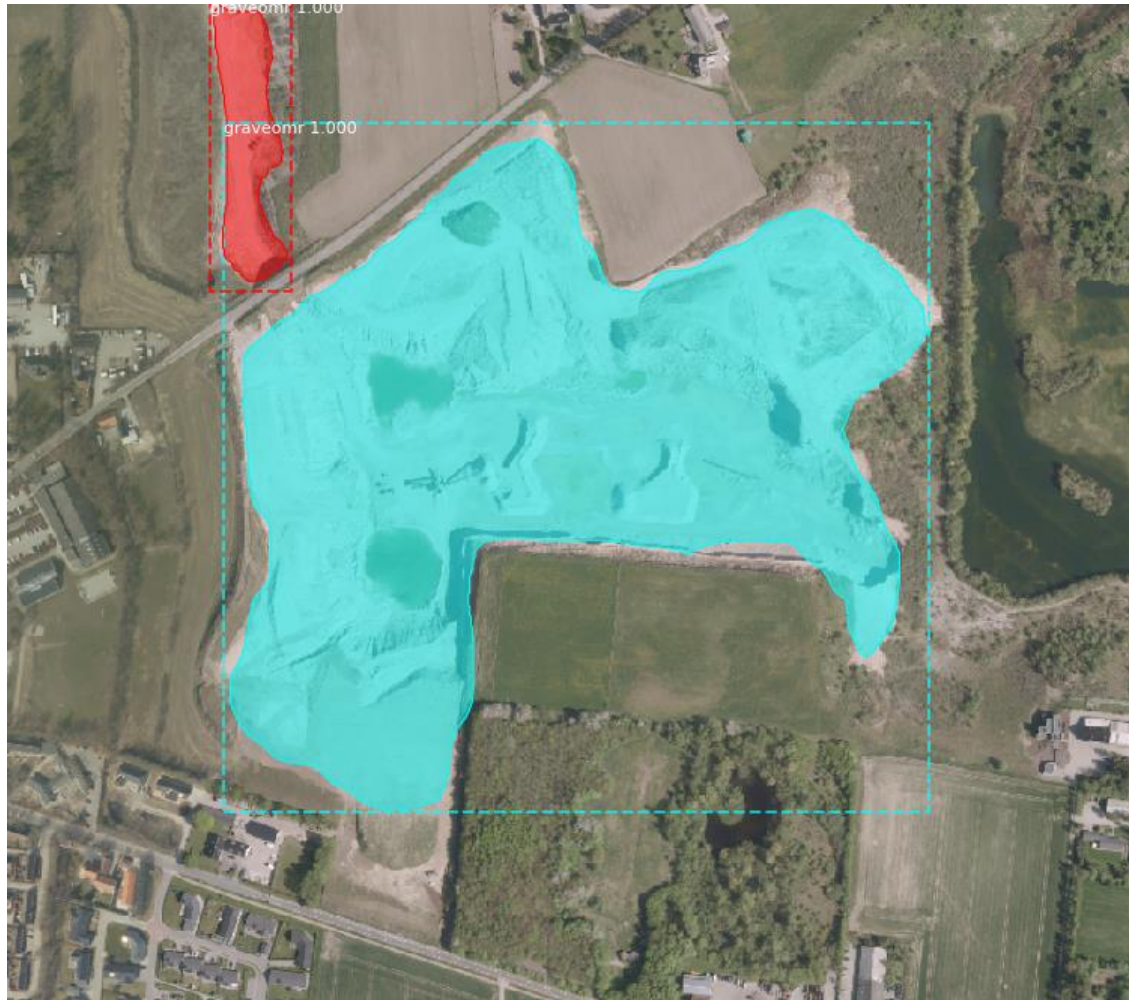
Sample generated by this project

Forsøg med Mask R-CNN

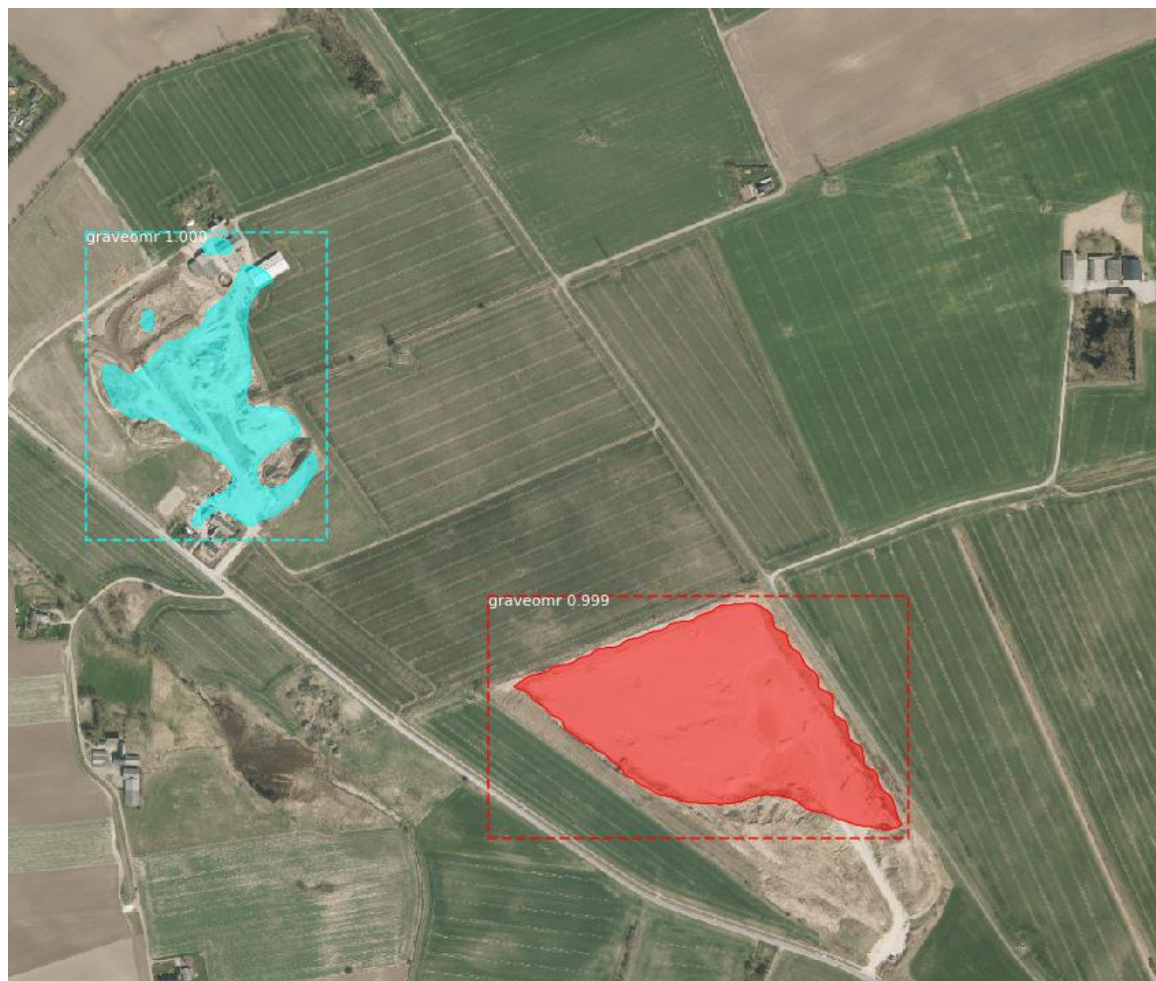
```
ak@WS01: ~/Sync/IT-diplom/2019_1_BD/Projekt
File Edit View Search Terminal Help
rpn_class_raw (Conv2D)
rpn_bbox_pred (Conv2D)
mrcnn_mask_conv1 (TimeDistributed)
mrcnn_mask_bn1 (TimeDistributed)
mrcnn_mask_conv2 (TimeDistributed)
mrcnn_mask_bn2 (TimeDistributed)
mrcnn_class_conv1 (TimeDistributed)
mrcnn_class_bn1 (TimeDistributed)
mrcnn_mask_conv3 (TimeDistributed)
mrcnn_mask_bn3 (TimeDistributed)
mrcnn_class_conv2 (TimeDistributed)
mrcnn_class_bn2 (TimeDistributed)
mrcnn_mask_conv4 (TimeDistributed)
mrcnn_mask_bn4 (TimeDistributed)
mrcnn_bbox_fc (TimeDistributed)
mrcnn_mask_deconv (TimeDistributed)
mrcnn_class_logits (TimeDistributed)
mrcnn_mask (TimeDistributed)
WARNING:tensorflow:From /home/ak/anaconda3/lib/python3.6/site-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
/home/ak/anaconda3/lib/python3.6/site-packages/tensorflow/python/ops/gradients_impl.py:110: UserWarning: Converting sparse IndexedSlices to a dense Tensor of unknown shape. This may consume a large amount of memory.
"Converting sparse IndexedSlices to a dense Tensor of unknown shape."
/home/ak/anaconda3/lib/python3.6/site-packages/keras/engine/training_generator.py:47: UserWarning: Using a generator with 'use_multiprocessing=True' and multiple workers may duplicate your data. Please consider using the 'keras.utils.Sequence' class.
UserWarning('Using a generator with 'use_multiprocessing=True'')
Epoch 1/30
2019-05-18 09:44:13.558353: I tensorflow/stream_executor/dso_loader.cc:152] successfully opened CUDA library libcublas.so.10.0 locally
2019-05-18 09:44:22.361124: W tensorflow/core/common_runtime/bfc_allocator.cc:211] Allocator (GPU_0_bfc) ran out of memory trying to allocate 3.03GiB. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memory were available.
1/100 [.....] - ETA: 7:00:34 - loss: 4.5570 - rpn_class_loss: 0.0657 - rpn_
2/100 [.....] - ETA: 3:28:44 - loss: 4.3747 - rpn_class_loss: 0.0656 - rpn_
3/100 [.....] - ETA: 2:18:06 - loss: 4.1384 - rpn_class_loss: 0.0654 - rpn_
4/100 [.....] - ETA: 1:42:46 - loss: 3.8789 - rpn_class_loss: 0.0651 - rpn_
5/100 [.....] - ETA: 1:21:34 - loss: 3.6187 - rpn_class_loss: 0.0648 - rpn_
6/100 [.....] - ETA: 1:07:26 - loss: 3.3657 - rpn_class_loss: 0.0643 - rpn_
7/100 [=====>] - ETA: 57:20 - loss: 3.1317 - rpn_class_loss: 0.0637 - rpn_
8/100 [=====>] - ETA: 49:45 - loss: 2.9256 - rpn_class_loss: 0.0630 - rpn_
9/100 [=====>] - ETA: 43:52 - loss: 2.7545 - rpn_class_loss: 0.0622 - rpn_
10/100 [=====>] - ETA: 39:09 - loss: 2.6139 - rpn_class_loss: 0.0612 - rpn_
11/100 [=====>] - ETA: 35:17 - loss: 2.4862 - rpn_class_loss: 0.0601 - rpn_
12/100 [=====>] - ETA: 32:04 - loss: 2.3801 - rpn_class_loss: 0.0589 - rpn_
bbox_loss: 0.1253 - mrcnn_class_loss: 0.1348 - mrcnn_bbox_loss: 0.4726 - mrcnn_mask_loss: 1.5886
```



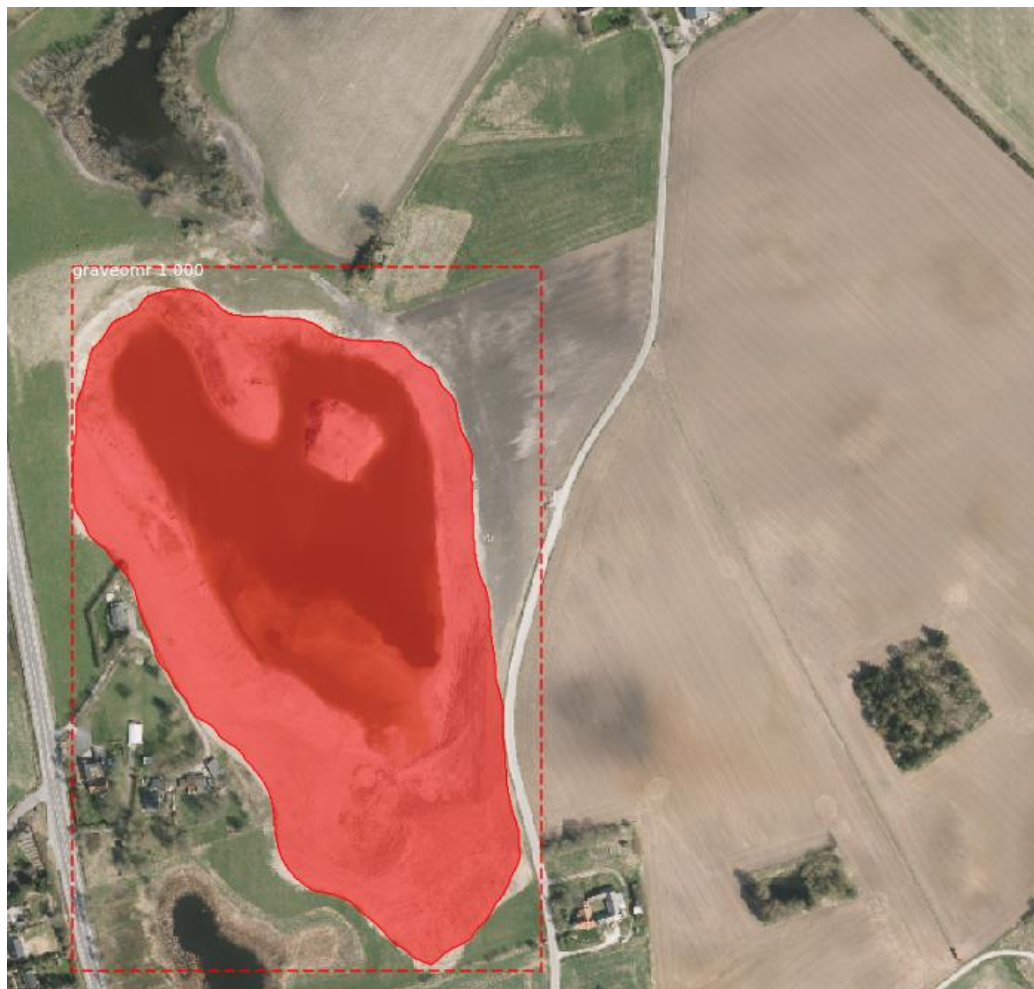
Testkørsel af model



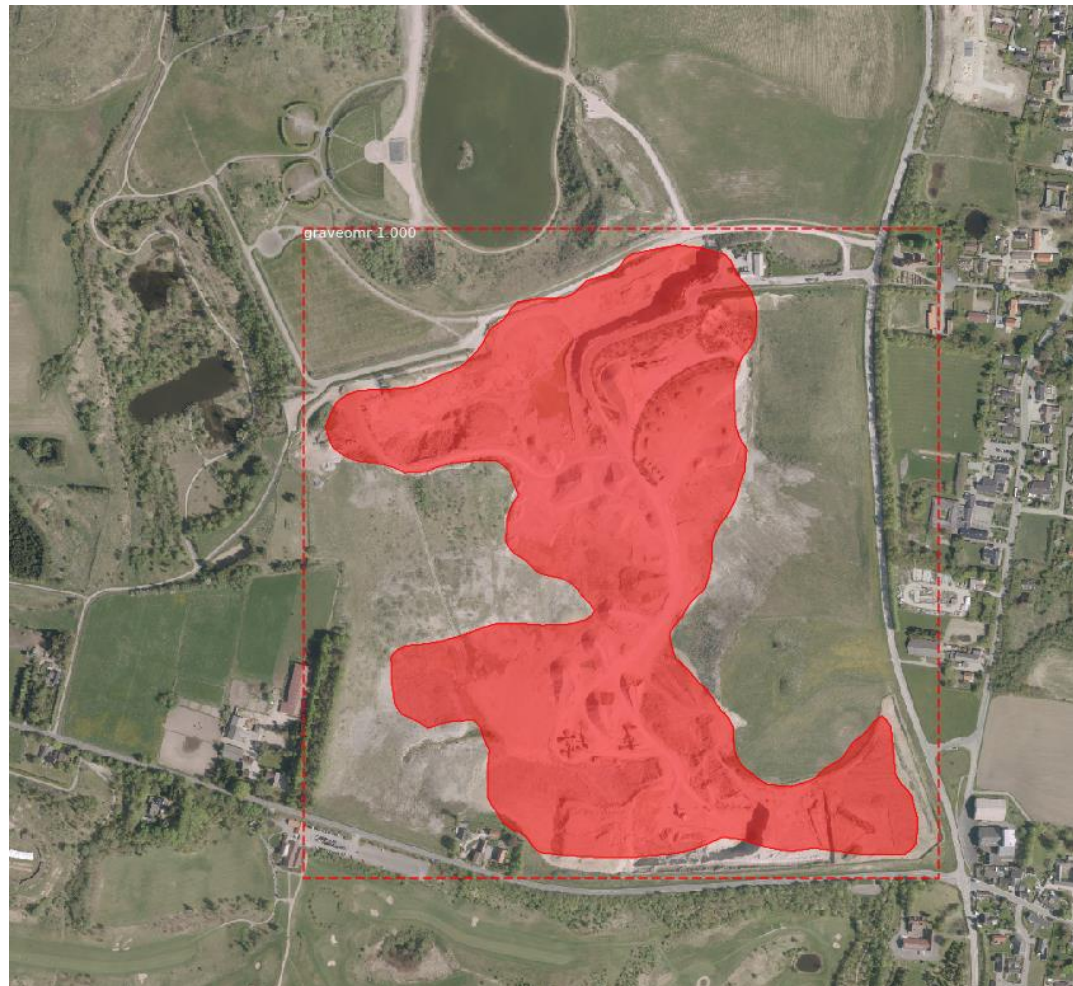
Testkørsel af model



Testkørsel af model



Testkørsel af model



Test af model



Test af model



Test af model



Konklusion

- Det er lykkedes at træne et Neuralt Netværk, til at kunne klassificere og segmentere aktive graveområder ud fra ortofoto.
- Modellen er ikke tilstrækkelig nøjagtig til at kunne bruges i sagsbehandlingen på nuværende tidspunkt.
- Modellens nøjagtighed vurderes at kunne forbedres:
 - Større datasæt
 - Flere parametre

Perspektivering

- Relativt tilgængeligt at komme i gang med avancerede analysemetoder og Machine Learning.
 - Anaconda er et komplet analysemiljø
 - Mange nye algoritmer implementeres i open source distributioner.
 - Omfattende online dokumentation og "communities" omkring open source.
- NB!
 - Computerressourcer → Cloud Computing
 - Implementering og drift af modellen

Tak for opmærksomheden!