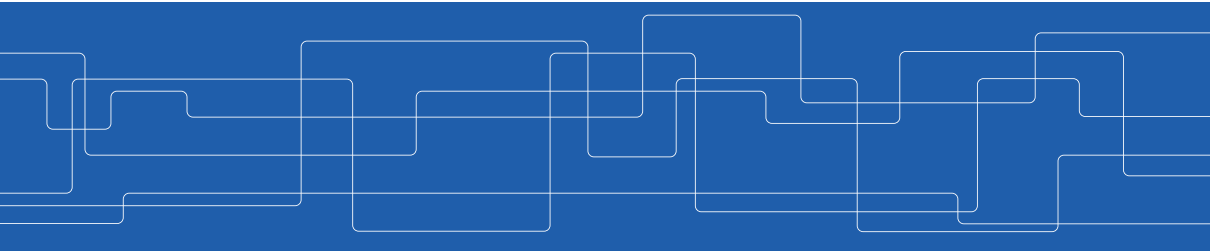
BLACKROCK

Investigation of deep learning approaches for overhead imagery analysis

Joar Gruneau



How do we detect vehicles?

Figure: A overhead image with a resolution of 30 cm/pixel

Previous techniques

Earlier methods

- Sliding window techniques
- Aggregating pixels into super pixels before analysing
- Two stage segmentation networks
- Fast region-based convolutional neural networks (fast R-CNN)

Two stage segmentation networks

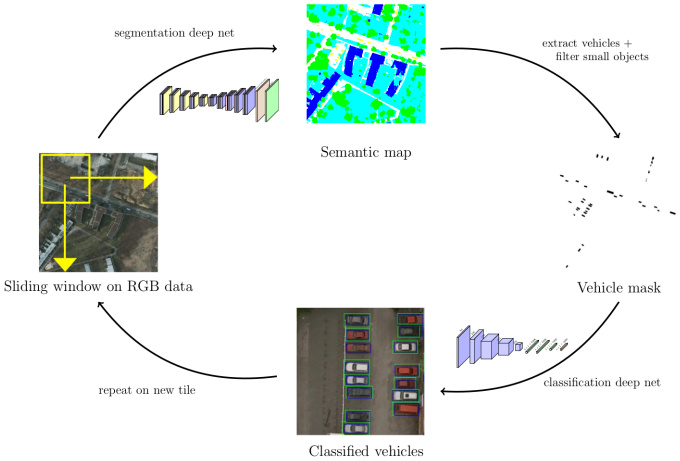


Figure: The Segment Before you Detect network structure

Fast R-CNN

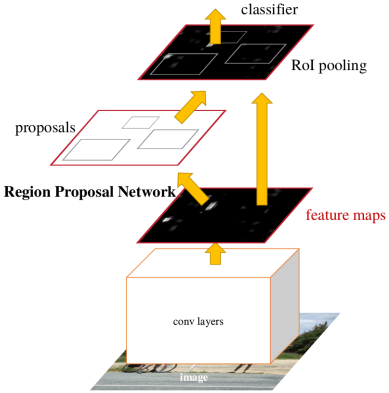


Figure: The two stage fast R-CNN

Previous techniques

Drawbacks

- Segment Before you Detect is slow due complex structure with two deep networks
- Fast R-CNN can only predict some predefined ratios of bounding boxes which affects the precision

The goal

Extract vehicles directly from the segmentation map

- Superior speed compared to Segment Before you Detect methods
- A more general approach compared to fast R-CNN since arbitrary size bounding boxes can be predicted
- The final network will be more simple considering only one stage is needed.

Proposed Model

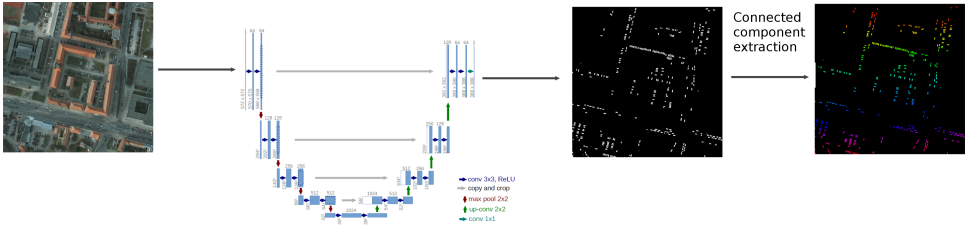


Figure: The architecture of the proposed model

KTH VETENSKAP OCH KONST BLACKROCK

The problem

The normal cross entropy loss does not enforce object separation sufficiently

- Small separations between vehicles are likely to be overlooked

Find a loss function that enforces object separation

- Adding an adversarial loss term
- Weighting the cross entropy loss based on vehicle separation

Adding an adversarial loss term

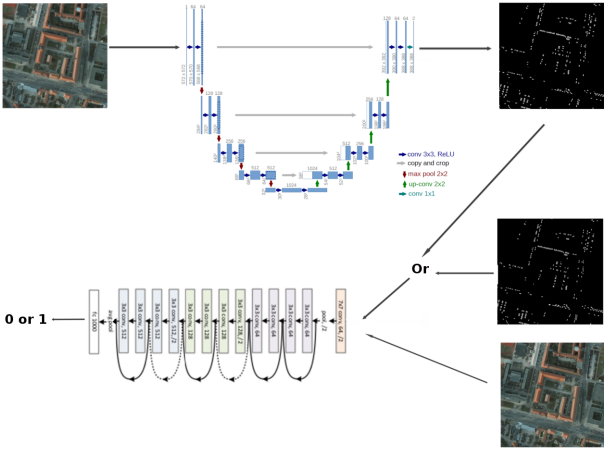


Figure: The generative adversarial network architecture

Adding an adversarial loss term

Builds on a counterfeit game between the semantic and discriminating network

$$\ell_{bce}(\hat{y}, y) = -(y * ln(\hat{y}) + (1 - y) * ln(1 - \hat{y}))$$

$$\ell_{pxl}(\hat{y}, y) = -\sum_{x=1}^{H*W} y(x) * log(\hat{y}(x))$$

$$\mathcal{L}(G) = \ell_{bce}(D(x, G(x)), 1) + \lambda \ell_{pxl}(G(x), y)$$

$$\mathcal{L}(D) = \ell_{bce}(D(x, G(x)), 0) + \ell_{bce}(D(x, y), 1)$$

KTH VETENSKAP OCH KONST BLACKROCK

Weighting the cross entropy loss

Enforce the network to learn small separations between vehicles

$$\ell_{pxl}(\hat{y}, y) = -\sum_{x=0}^{H*W} \omega(x) * y(x) * log(\hat{y}(x))$$

$$\omega(x) = \omega_c + \omega_0 * epx(\frac{(-d_1(x) - d_2(x))^2}{2\sigma^2})$$

$$\omega_c = 2,\ \omega_0 = 10,\ \sigma = 3$$

Weighting the cross entropy loss

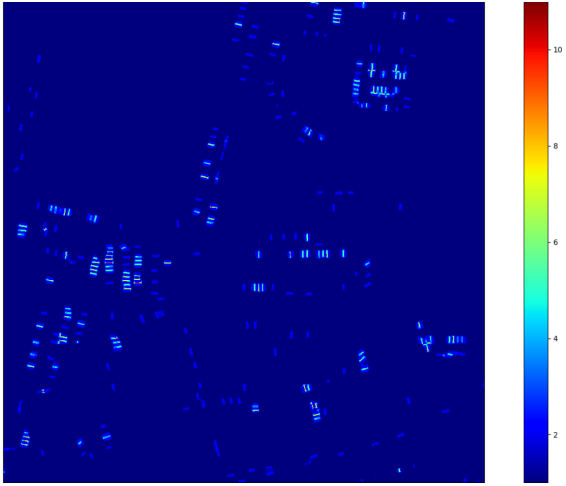


Figure: The pixel wise weight map used to enforce seperations between vehicles

The datasets

The model will be evaluated on two datasets

- The ISPRS Potsdam dataset
  - A segmentation dataset
  - 5 cm/pixel resolution
  - $\approx$ 98 % of all pixels belong to the background class
  - Images will be down sampled to 30 cm/pixel to match resolution of commercial satellite images.
  - 24 images for training and validation, 14 for testing.
- The Vedai dataset
  - A vehicle detection dataset
  - Vehicles are marked with bounding boxes
  - 25 cm/pixel resolution
  - Severe class imbalance $\approx$ 99.3 % of all pixels belong to the background class
  - 927 images for training, 100 for validation, 240 for testing.

The Potsdam dataset

Figure: The left image shows the input image and the right image the ground truth segmentation

Evaluation metrics

A vehicle is defined as detected if the intersect over union of its predicted area and its ground truth area is greater than 0.5

$$IoU = \frac{A_{pred} \cap A_{true}}{A_{pred} \cup A_{true}} > 0.5$$

$$Precision = \frac{True\ positive}{True\ Positive + False\ poitive}$$

$$Recall = \frac{True\ positive}{True\ Positive + False\ negative}$$

$$F1\ score = \frac{2 * Recall * Precision}{Recall + Precision}$$

Results: Adding a adversarial loss term

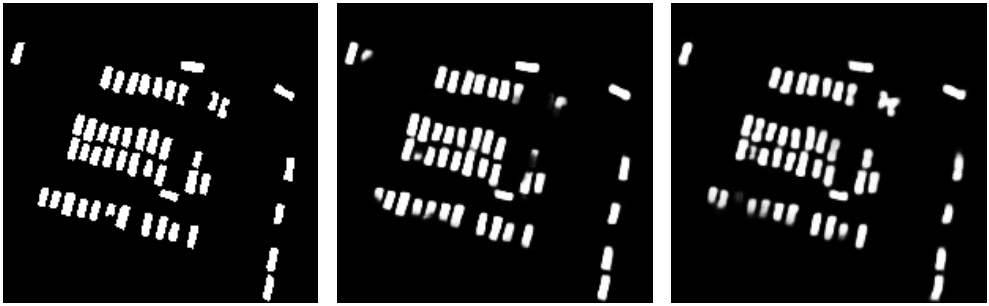


Figure: The the ground truth segmentation, the soft prediction without and with an adversarial loss

Results: Adding a adversarial loss term

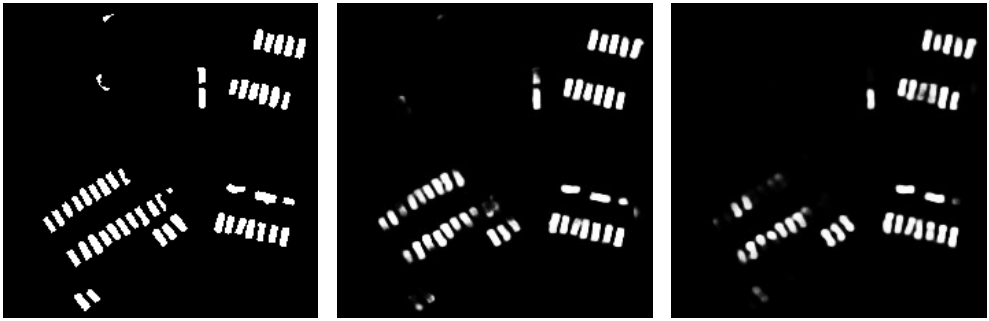


Figure: The the ground truth segmentation, the soft prediction without and with an adversarial loss

Results: Adding a adversarial loss term

The network with an adversarial loss was much more unstable, harder to train and did not increase the F1 pixel wise score on the validation dataset.

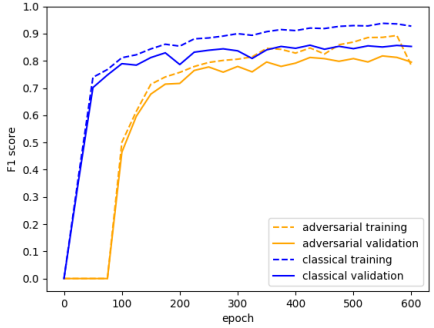


Figure: The F1 pixel wise score with and without an adversarial loss

Results: Weighting the loss function

Figure: The ground truth segmentation, the hard prediction with and without a weighted loss function

Results: Weighting the loss function

Figure: The ground truth segmentation, the hard prediction with and without a weighted loss function

Results: Weighting the loss function

Weighting the loss function enforced better object separation while obtaining comparable pixel wise F1 score.

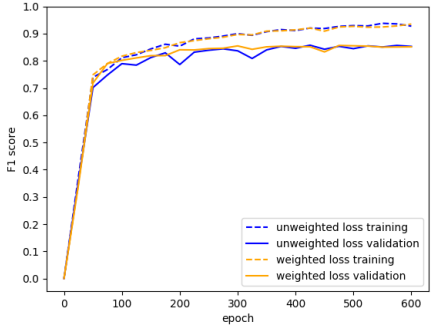


Figure: The F1 pixel wise score with and without loss weighting

Final results on the Potsdam dataset

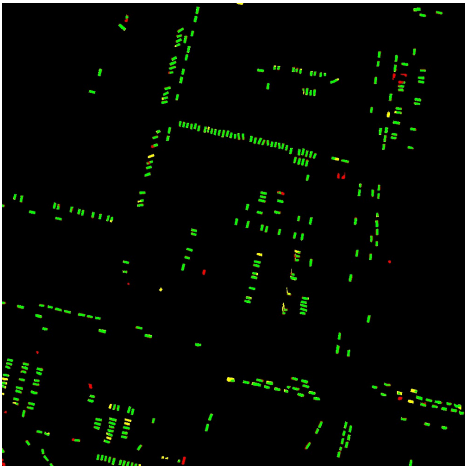


Figure: The input and output segmentation. Green is true positives, yellow are false negatives and red are false positives.

Final results on the Potsdam dataset

Vehicle detection

- The proposed model obtains a pixel wise F1 score 0.851
- The proposed model obtains a vehicle wise F1 score 0.811
- Mean evaluation time per image is 0.19 seconds = 2 seconds per square kilometre

Final results on the Potsdam dataset

Counting cars

- The connected components of the predicted segmentation and the ground truth segmentation were extracted and counted
- Mean prediction error of 1.67 %

Tile #	2_11	2_12	7_9	7_10	7_11	7_12
Number of vehicles	107	123	304	250	346	346
Predicted number of vehicles	108	126	310	251	352	337
Prediction error in %	0.9	2.4	1.9	0.4	1.7	2.7

Comparison with earlier work on Potsdam

Model	Proposed Model	SBD
Resolution used	**30 cm/pixel**	12.5 cm/pixel
Pixel wise F1 score	0.851	**0.884**
Vehicle wise F1 score	**0.811**	0.773
Mean prediction error counting cars	**1.67 %**	3.57 %
Evaluation time per image *	**0.19 seconds**	28.19 seconds

Table: Shows the comparison between the proposed model and the Segment before you Detect (SBD) model [1] on the Potsdam dataset. ***** The SBD model was evaluated on a Tesla K20 which can at maximum perform $3.52 * 10^{12}$ 32 bit floating point operations per second. The proposed model was evaluated on a Tesla K80 which can perform at maximum $8.74 * 10^{12}$ 32 bit floating point operations per second. Therefore the evaluation time on the SBD model was multiplied with $3.52/8.74 \approx 0.4027$ to make a fair comparison.

Comparison with earlier work on Vedai

Model	Detection time per image *	Vehicle wise F1 score
Faster R-CNN (Z&F)	0.1998	0.212
Faster R-CNN (VGG-16)	0.2248	0.225
Fast R-CNN (VGG-16)	3.1465	0.224
CCNN	0.2736	0.305
Proposed Model	**0.0616**	**0.542**

Table: Shows the comparison between the proposed model and the Faster R-CNN (Z&F), Faster R-CNN (VGG-16), Fast R-CNN (VGG-16) [2] and the Cascaded Convolutional Neural Networks (CCNN) [3] on the Vedai dataset. * The other models were evaluated by [3] on a Titan X which can at maximum perform $11 * 10^{12}$ 32 bit floating point operations per second. The proposed model was evaluated on a Tesla K80 which can perform at maximum $8.74 * 10^{12}$ 32 bit floating point operations per second. Therefore the evaluation time on the compared models were multiplied with $11/8.74 \approx 1.2586$ to make a fair comparison.

Final conclusions

Proposed model characteristics

- The proposed model has a computational time which is only a fraction of the SBD model while also obtaining higher vehicle wise F1 score and lower counting prediction error.
- The proposed model can use much lower resolution images which makes it a viable method for detecting and counting cars in satellite images.
- The proposed model outperforms the R-CNN models with a significant margin while also obtaining a slightly lower computational time.
- The proposed model only need a few images for training which means that building new datasets for training is a low cost operation.
- The objects need to have a separation for connected component extraction to work. This can however be solved by introducing an "artificial" separation between touching objects.

Nicolas Audebert, Bertrand Le Saux, and Sbastien Lefvre. "On the usability of deep networks for object-based image analysis". In: *arXiv:1609.06845 [cs]* (Sept. 2016). URL: http://arxiv.org/abs/1609.06845 (visited on 03/09/2018).

Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". en. In: *Computer Vision ECCV 2014*. Lecture Notes in Computer Science. Springer, Cham, Sept. 2014, pp. 818–833. ISBN: 978-3-319-10589-5 978-3-319-10590-1. DOI: 10.1007/978-3-319-10590-1_53. URL: https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53 (visited on 04/19/2018).

Jiandan Zhong, Tao Lei, and Guangle Yao. "Robust Vehicle Detection in Aerial Images Based on Cascaded Convolutional Neural Networks". en. In: *Sensors* 17.12 (Nov. 2017), p. 2720. DOI: 10.3390/s17122720. URL: http://www.mdpi.com/1424-8220/17/12/2720 (visited on 04/19/2018).