Investigation of deep learning approaches for overhead imagery analysis

JOAR GRUNEAU

Master in Computer Science
Date: May 2, 2018
Supervisor: Kevin Smith
Examiner: Danica Kragic
Swedish title: Utredning av djupinlärnings metoder för satellit och flygbilder
School of Electrical Engineering and Computer Science

Abstract

Analysis of overhead imagery has a great potential to produce real time data cost effectively. This can be an important foundation for decision-making for businesses and politics. Every day a massive amount of new satellite imagery is produced. To fully take advantage of these data volumes a computationally efficient pipeline is required for the analysis. This thesis proposes a pipeline which outperforms the Segment Before you Detect network [6] and different types of fast region based convolutional neural networks [61] with a large margin in a fraction of the time. The model obtains a prediction error for counting cars of 1.67% on the Potsdam dataset and increases the vehicle wise F1 score on the Vedai dataset from 0.305 reported by [61] to 0.542. This thesis also shows that it is possible to outperform the Segment Before you Detect network in less than 1% of the time on car counting and vehicle detection while also using less than half of the resolution. This makes the proposed model a viable solution for large scale satellite imagery analysis.

Sammanfattning

Analys av flyg och satellit bilder har stor potential att kostnadseffektivt producera data i realtid för beslutsfattande för företag och politik. Varje dag produceras massiva mängder nya satellitbilder. För att fullt kunna utnyttja dessa datamängder krävs en beräkningseffektivt nätverk för analysen. Denna avhandling föreslår ett nätverk som överträffar Segmentet Before you Detect nätverket [6] och olika typer av snabbt regionsbaserade convolutional neurala nätverk [61] med en stor marginal på en bråkdel av tiden. Den föreslagna modellen erhåller ett prediktionsfel för att räkna bilar på 1,67% på Potsdam datasetet och ökar F1-poängen for fordons detektion på Vedai datasetet från 0.305 rapporterat av [61] till 0.542. Denna avhandling visar också att det är möjligt att överträffa Segment Before you Detect nätverket på mindre än 1 % av tiden på bilräkning och fordonsdetektering samtidigt som den föreslagna modellen använder mindre än hälften av upplösningen. Detta gör den föreslagna modellen till en attraktiv lösning för storskalig satellitbildanalys.

Contents

Chapter 1

Introduction

Convolutional neural networks (CNN) have had great success for computer vision tasks [40, 57, 43, 21]. The success is possible in part thanks to graphical processing units (GPUs) and large scale human annotated datasets which the networks can learn from. CNN have progressed from single object detection in images [20] to multiple object detection and bounding box prediction [24]. CNN networks have also had great success in different segmentation task [10]. There is a similar trend in segmentation where we are moving from the easier task of semantic segmentation to the more complex task of instance segmentation. In semantic segmentation every pixel is mapped to a class and in instance segmentation the different instances of objects are separated detected for each class. It is not trivial to construct a loss function for segmentation which enforces the desired properties into the segmentation network. Much work has gone into this problem which has resulted in several different types of loss functions [36, 55, 31]. Often the loss function fails to enforce important properties such as spatial contiguity in the segmentation maps [25] or proper spatial separation of objects [5]. Conditional Markov random fields (CRFs) have been a popular post processing step to ensure spatial contiguity in segmentation maps [59]. Ronneberger *et al.* [36] also shows that object separation can be enforced by weighting the pixel wise loss function based on the proximity to nearby objects.

A new popular network for image to image translations are generative adversarial networks. The network consists of a generator network which performs the image translation and a discriminative network

which aims to learn the loss function to differentiate the generated samples from the ground truth ones [12]. These type of networks have had great success on image to image translation tasks and are able to produce much more artistically pleasing mappings then networks without the adversarial loss [22, 18]. The success comes from the general approach where the network can learn its own loss function which has proven beneficial for many tasks where a effective loss function is hard to express. These types of networks have also been applied to image segmentation and has shown to give an increased performance [25, 46]. GANs have proven to be especially successful on small dataset such as medical segmentation where the human annotations usually are costly due to the required medical expertise needed to create correct annotations [46, 53, 54, 35, 3].

In today's information society data is becoming more and more valuable. The data is used as a foundation to do business decisions as well as political decisions daily. The life time of new data is also decreasing and new data is considered old and unrepresentative much faster than before. Thanks to the large number of satellites and the low cost to produce high resolution satellite images, analysis of satellite images can be a useful tool to obtain real time data cost effectively. To mention a few applications it can be used for traffic flow monitoring [37, 28] vegetation monitoring [50, 8], urban area monitoring [27], water reserve capacity monitoring, generating new maps [18] and even to detect endangered whales [30]. Investors are continuously competing to obtain an edge over their competitors and turn a profit. Here satellite imagery analysis can be used to obtain fresh information before it reaches the market. For example, if we continuously can count the number of vehicles outside a marketplace we can more accurately predict how many customers that are visiting the marketplace and therefore make more accurate predictions about the markets earnings before those earnings are released to the public market.

Much research has been performed investigating object detection and more specifically vehicle detection in overhead imagery [2, 17, 5, 32, 62, 6, 38]. However object detection in overhead images has proven to be a troublesome area. The objects of interest are usually very small compared to the image and there can be multiple objects within im-

age. This causes naive classification networks to achieve bad performance if the entire image is fed in at once [2]. To combat this some form of segmentation is usually done and the image is fed into the network in patches. Earlier methods fed explicit image patches through the CNN using sliding window techniques[17]. This achieved good performance but at a great computational cost since redundant computation of low-level filters for overlapping patches had to be performed [25]. To combat this different forms of segmentation algorithms were used such as the mean-shift-algorithm which drastically decreased the number of patches which had to be fed through the network [2].

This thesis investigate how to define a loss function which enforces good vehicles separation directly into the segmentation network for overhead imagery. Two different approaches are investigated, weighting the loss based on vehicle separation and adding an adversarial loss term. In both cases the desired outcome is to force the network to pay closer attention to segmentation around nearby objects. The different instances of vehicles can then be extracted directly from the segmentation maps by connected component extraction. This pipeline would achieve superior speed compared to earlier more modern methods which usually uses a two stage approach since only a single stage network would be needed at training and testing.

An example of these types of two stage detection networks is the Segment Before You Detect (SBD) [5] and the fast region-based convolutional network (fast R-CNN) [34, 11] or mask region-based convolutional network (mask R-CNN). The SBD pipeline uses a segmentation network to find the vehicle patches. The patches are then extracted and fed into a classification network to predict vehicles and find bounding boxes. The fast R-CNN or mask R-CNN [16] achieves superior speed compared to the SBD pipeline. These networks uses a second stage region proposal network (RPN) to compute the bounding box predictions on internal convolutional feature maps. This approach minimizes computational time since layers can be shared. However these networks can only predict some predefined ratio of bounding boxes and the two stage network adds complexity both at training and test time.

1.1 Research Question

In this thesis we will investigate if it is possible to construct a loss function which enforces vehicle separation directly into the segmentation network for satellite imagery. The goal is to enforce good enough vehicle separation so instances of vehicles can be directly found by connected component extraction in the segmentation map. The proposed models will be compared to the network trained with the standard cross entropy loss to measure improvements. The metrics for evaluation will be pixel wise F1 score and vehicle wise F1 score as well as visual inspection of the segmentation maps. The final model will be compared with earlier work such as the SBD and fast R-CNN on the Potsdam and Vedai datasets on the metrics pixel wise F1 score, vehicle wise F1 score, prediction error counting cars and evaluation time.

Chapter 2

Background

2.1 Generative adversarial networks

Goodfellow *et al.* [12] first proposed the generative adversarial network (GAN). The network consists of two parts, a generator and a discriminator. The generator's task is to generate samples from some data distribution. The discriminators task is to differentiate these generated samples from the true samples. This results in a counter fitting game where the generator continuously tries to produce better generated data to fool the discriminator and the discriminator is forced to become better at differentiating these generated samples from the true samples.

A common solution to try to force the generator to generate samples from the entire distribution is to input a noise vector into the generator [33]. Since this thesis focuses on segmentation where a deterministic mapping from the image to the segmentation map is desired no noise vector will be fed into the generator.

2.1.1 Unconditional generative adversarial networks

Unconditional GANs are the simplest form of GANs. Here the discriminator does not observe the input to the generator. This means that the discriminator will learn a loss function which does not depend on the generators input [18]. We first define the binary cross entropy loss.

$$\ell_{bce}(\hat{y}, y) = -(yln(\hat{y}) + (1-y)ln(1-\hat{y})) \tag{2.1}$$

Here $\hat{y}$ is the prediction and y is the ground truth. The loss function for a unconditional GAN can then be described as.

$$\mathcal{L}(G, D) = -(\ell_{bce}(D(y), 1) + \ell_{bce}(D(G(x)), 0)) \tag{2.2}$$

Here D is the discriminating network, G is the generating network, y ground truth sample and x is some input which should be translated by the generator to look like it comes from the ground truth distribution. G tries to minimize this function and D tries to maximize it. Hence we get a minimax game

$$G^* = argmin_G[max_D[\mathcal{L}(G, D)]] \tag{2.3}$$

2.1.2 Conditional generative adversarial networks

A conditional generative adversarial network (cGAN) was proposed by [26]. By letting the discriminator observe the input to the generator we can condition the loss function the discriminator learns on this input. This is of great importance here since we are not just trying to generate any segmentation maps but segmentation maps corresponding to the input image. The objective function will in this case be given by the below equation.

$$\mathcal{L}(G, D) = -(\ell_{bce}(D(x, y), 1) + \ell_{bce}(D(x, G(x)), 0)) \tag{2.4}$$

It has been shown that a multi term loss function can improve the quality of the generator [29, 18]. For image to image mappings a $\mathcal{L}_1$ or $\mathcal{L}_2$ loss is usually used. However for image segmentation a cross entropy loss is a better option to enforce the generator to assign a high probability to the correct class for each pixel. The pixel wise cross entropy loss is given below.

$$\ell_{pxl}(\hat{y}, y) = -\sum_{x=1}^{H*W} y(x) * log(\hat{y}(x)) \tag{2.5}$$

Here $y(x)$ is the one hot encoding for pixel x and $\hat{y}(x)$ is the predicted probabilities. H and W is the height and width of the image. The discriminator's objective is unchanged but the generator now has to fool the discriminator as well as minimizing the distance to the ground truth in respect to the cross entropy.

$$G^* = argmin_G[max_D[\mathcal{L}(G, D)] + \lambda \ell_{pxl}(G)] \tag{2.6}$$

Here λ is a constant which controls the importance of the second loss term.

2.1.3 Training generative adversarial networks

Generative adversarial networks are notoriously difficult to train. The minimax game formulated in 2.3 relies on both networks being of equal strength. If the generative network is too strong the discriminative network will not be able to differentiate between the generated and the true samples and will not provide an effective loss for the generator. If the discriminative network is too strong it will be able to differentiate between the generated samples and the true samples very effectively no matter what the generator does. The gradients will then be very small, and the generative network will have a problem with vanishing gradients. In practice, it is very hard to achieve this balance but some common tricks is making the generative or discriminative networks more or less complex as well as training them at different amount on steps each iteration.

A collection of tips for stabilizing the GAN game has been gathered by [9]. The minimax game becomes more unstable with sparse gradients. This means that leaky Relu should be used instead of Relu for the activation function. For down sampling mean pooling should be used instead of max pooling. For up sampling a 2D transposed convolution or PixelShuffle [42] can be used. The collection also recommends smoothing labels so that the discriminator has to learn a more effective loss function than to simply differentiate between the generated and ground truth labels by checking if they are continuous or not.

2.2 Segmentation networks

For the generative part of a GAN a segmentation network is needed. This network takes a image as an input and produces segmentation maps. A fully convolutional network (FCN) was first proposed Shelhamer and Long *et al.* [41]. A FCN is a CNN without any fully connected layers. A network with fully connected layers must have a specific input size on the image while a FCN network can take inputs of any size. The key insight made by the authors were that fully connected layers can be viewed as convolutions with kernels that cover their entire input region. Hence, a CNN with fully connected layers can be viewed as a FCN since it takes patches from a image of any size and outputs a spatial output map when the patches are aggregated. While

the resulting maps are equivalent the computational cost for the FCN is greatly reduced. This is because no overlapping regions between patches has to be computed. This makes these networks ideal for generating dense output maps such as for image segmentation.

Ronneberger *et al.* [36] builds on the advancements of the FCN to propose a new type of segmentation network. The U-Net uses a encoder decoder structure with skip connections from bottleneck layers to up sampled layers. These skip connections are crucial for segmentation tasks as the initial feature maps maintain low-level features which need to be properly exploited for accurate segmentation at later stages. The network has been shown to produce high accuracy results even on small sized datasets [45, 36, 18, 53, 54].

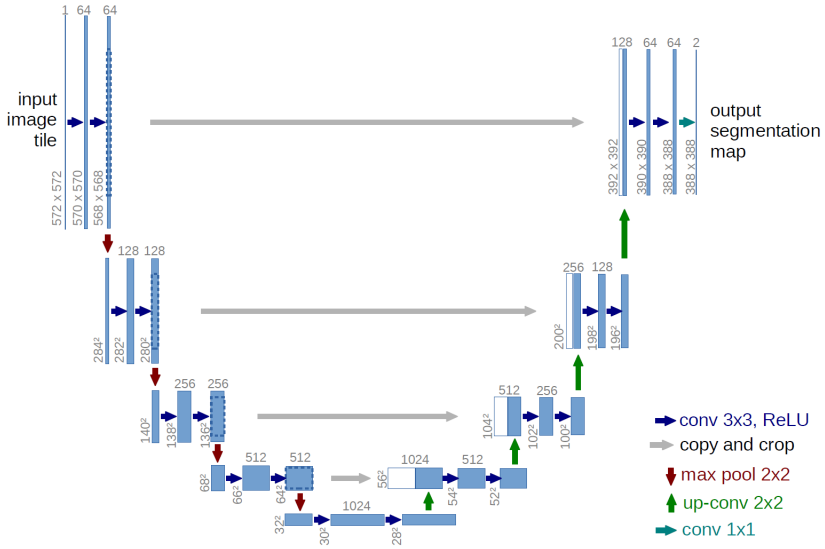

Figure 2.1: Shows the U-Net architecture proposed by Ronneberger *et al.* [36].

Notice that the size of the input image and the output prediction in 2.1 is different. This is because the U-Net uses unpadded convolutions so at every convolutional layer one pixel is lost at every side. To obtain predictions in the borders of an image the context is extrapolated by mirroring the image [23]. It is also important that we choose an initial image size so the activation maps length is even through all layers of the network. [36]. In theory the U-Net can handle images of any size but in practice we have memory limitations for the GPU. Large input images

is therefore split up into patches. The segmentation on the patches can then be directly stitched together without any overlapping because the net uses unpadded convolutions.

2.3 Classification networks

For the discriminative part of the GAN a classification network is needed. This takes a input image and a set of segmentation maps and decides if the segmentation maps are artificial or ground truth. There are several high preforming classification networks such as the VGG networks [44] and the ResNet networks [14].

2.3.1 VGG Networks

The popular 16 layer VGG16 or the 19 layers VGG19 have performed very well on a wide variety of tasks such as classification [44]. They have been used inside Fast Region-based Convolutional Networks (fast R-CNN) [34, 11] to generate object activation maps for the regional proposal network (RPN). Fast R-CNN have been able to do object detection and bounding box prediction at a fraction of the time of earlier networks. They have also been used as a encoder in the SegNet architecture which has been a very popular segmentation network [7].

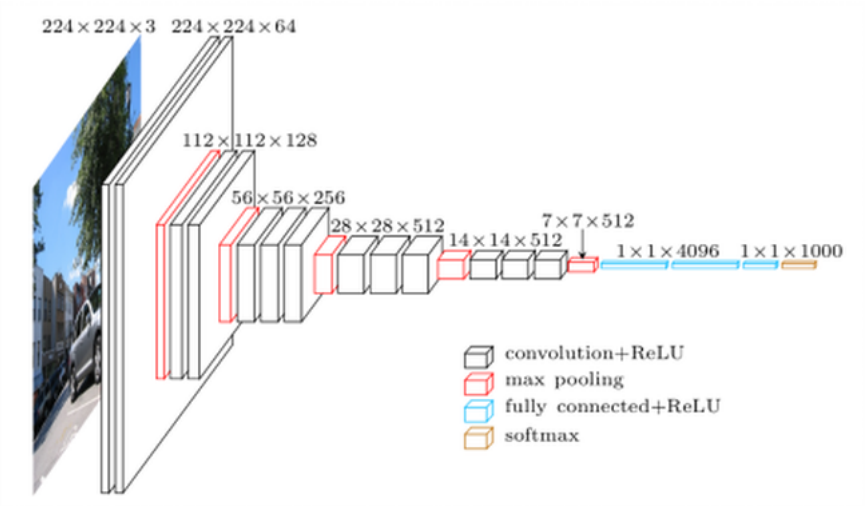

Figure 2.2: The VGG16 architecture proposed by [44]

The VGG16 network takes inputs of images with size 224×224 and

produces a class wise prediction. The network is very simple, yet powerful and uses only 3×3 convolutions and 2×2 max pooling layers with stride 2.

2.3.2 ResNet networks

ResNet is another widely popular classification network. It has performed very well on classification tasks [15, 49] It has also been shown that wider residual networks are more memory efficient while still obtaining comparable performance [52, 56].

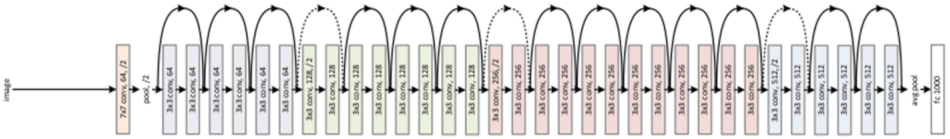

Figure 2.3: The 34 layer deep ResNet architecture [44]

The success behind the ResNet lies in the residual blocks. A residual block is a two or more convolutional layers with a identity short-cut connection in between. The residual block can be defined as.

$$y = F(x, W) + x \tag{2.7}$$

The benefit of this is that this guarantees that the gradients will flow through the entire network. Therefore, it is possible to build and train very deep residual networks by stacking hundreds of residual blocks. This also allows the network to itself decide the required depth since it can learn to set the weight of the final residual blocks to zero if they are not needed [14]. Hence, we get a very flexible network where we easily can adapt the depth to the task at hand.

2.4 Weight functions and dealing with severely imbalanced datasets

Due to the nature of overhead images the objects of interest are usually small compared to the entire image. This causes the dataset to be imbalanced since most pixels will belong to the background class. A naive segmentation network could then obtain good accuracy by only

predicting everything to the background class. To combat this there are several techniques. The most straight forward is to use a weighted cross entropy loss where every term is weighted depending on the class frequency or predicted frequency [48]. Below is the weighted cross entropy loss for a two class segmentation problem.

$$\ell_{wce}(\hat{y}, y) = -\omega_c y log(\hat{y}) + (1 - y) log(1 - \hat{y}) \tag{2.8}$$

Here y is the true probability for the foreground class, $\hat{y}$ the predicted probability for the foreground class and ω_c the class weights for the foreground class depending on the frequency which can be defined in many different ways such as.

$$w_c = \frac{W * H - \sum_{x=0}^{W*H} \hat{y}_{c=1}(x)}{\sum_{x=0}^{W*H} \hat{y}_{c=1}(x)} \tag{2.9}$$

Here W and H are the width and height of the image and $\hat{y}_{c=1}(x)$ is the predicted probability that pixel x belongs to class one. Another class weighting technique i.

$$w_c = \frac{\sum_{n=0}^{N} \sum_{x=0}^{W*H} y_{c=0}(x)}{\sum_{n=0}^{N} \sum_{x=0}^{W*H} y_{c=1}(x)} \tag{2.10}$$

$\sum_{n=0}^{N} \sum_{x=0}^{W*H} y_{c=0}(x)$ is the sum over all of the background pixels in the training dataset. This is divided by the sum of the foreground pixels to obtain the class frequency ratio. Another popular approach is to minimize the intersect over union loss [55, 31]. The intersect over union loss for a two class segmentation task is given below.

$$\ell_{IoU}(\hat{y}, y) = 1 - \frac{\sum_{x=0}^{W*H} y(x) * \hat{y}(x)}{\sum_{x=0}^{W*H} y(x) + \sum_{x=0}^{W*H} \hat{y}(x) - \sum_{W*H} y(x) * \hat{y}(x)} \tag{2.11}$$

Here $\hat{y}(x)$ is the discrete class prediction for pixel x.

To ensure good object separation Ronneberger *et al.* [36] proposes to scale the pixel wise loss based on the proximity to the closest two foreground objects according.

$$\ell_{pxl}(\hat{y}, y) = -\sum_{x=0}^{H*W} \omega(x) * y(x) * log(\hat{y}(x)) \tag{2.12}$$

Here $\omega(x)$ is the weighting term for pixel x, y is the one hot encoding, $\hat{y}(x)$ is the discrete prediction and H and W is the height and width of the segmentation map. The weighting factor $\omega(x)$ is computed as following.

$$\omega(x) = \omega_c + \omega_0 * epx(\frac{(-d_1(x) - d_2(x))^2}{2\sigma^2}) \tag{2.13}$$

Here $\omega_c(x)$ depends on the class frequency and could be computed as 2.9 or 2.10. The distances $d_1(x)$ and $d_2(x)$ is the distance to the nearest and second nearest foreground object. The constant σ determines how fast the penalty should decay with increasing distance and ω_0 is a coefficient that determines the importance of the object separation penalty.

2.5 Earlier work

In most cases the segmentation networks needs some post processing to improve the accuracy of the segmentation maps. Conditional random markov fields CRF have been very successfully to enforce spatial contiguity in the output maps [4, 25]. There have been work that used mean field inference expressed as a recurrent convolutional networks to do CRF like post processing [39, 60]. Luc *et al.* [25] proposed a adversarial segmentation network to enforce higher order potentials without being limited to a single class. Instead, of directly enforcing these higher order potentials in a CRF model as post processing the goal was to enforce them in the generator directly with adversarial training. This technique also has the benefit of lower complexity since at test time only the generator will be used.

The generators task was to produce segmentation maps for the C classes. One initial concern was that the discriminator would trivially be able to differentiate the generated segmentation maps from the ground truth by only examining if they were continuous or discrete. To combat this a scaling method was proposed where the ground truth segmentation maps were processed so that a mass of τ where placed on the correct label but were otherwise made as similar as possible to the generated maps (in regard to KL divergence). The scaling method showed no improvement over the basic method with no preprocessing.

Son and Jung *et al.* [45] showed that a U-Net combined with an adversarial loss could achieve state of the art performance for retinal vessel segmentation in fundoscopic images. The team investigated several types on adversarial networks proposed in [18] such as image-GAN, patch-GAN and pixel-GAN. For Image-GAN the discriminator make a decision on a image level if the image is generated or not. For patch-GAN the images are split into patches and the discriminator analyses each individually. The result is the aggregated result from all patches. For pixel-GAN the discriminator makes it decision on pixel per pixel level. The team found that a image-GAN together with a cross entropy term preformed the best and outperformed the non adversarial segmentation network trained only with the cross entropy loss by a small margin.

Ronneberger *et al.* [36] showed that a U-Net could learn small border pixels in cell segmentation by weighting the loss as 2.12. The loss enforced the network to learn small border and outperformed the second best network on the ISBI cell tracking challenge 2015 with a large margin.

Chapter 3

Method

3.1 The segmentation network

3.1.1 Network architecture

For the segmentation network a U-Net proposed by [36] was used. Different number of filters in the initial layer was tried but it was concluded that more than 16 filters did not give a better performance and only resulted in a greater computational cost and overfitting. Dropout did not manage to solve the overfitting problem for the more complex networks so a U-Net with 16 initial filters and no dropout was chosen as the segmentation network for the following experiments. To produce data augmentation the training image were randomly rotated 0, 90 180 or 270 degrees as well as randomly mirrored left to right and up to down.

To extract vehicles from the segmentation map connected component extraction is used. Pixels can be defined to be connected by four or eight way connectivity. In four way connectivity only the above, below and two sideways neighbours are examined for connectivity but in eight way connectivity the diagonal neighbours are examined as well. Since all pixels in a vehicle should be four way connected with each other only use this mode will be used. The most simple and intuitive method of extracting connected components in an image is the two pass algorithm [47] which is a O(n) time algorithm. The below image shows the proposed pipeline at testing time for counting and detecting vehicles.

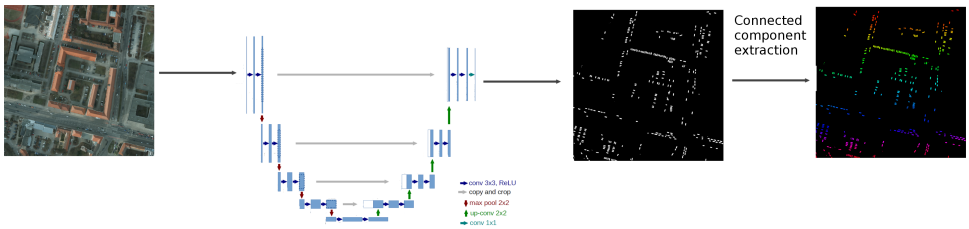

Figure 3.1: Shows the proposed vehicle detection and counting pipeline at test time.

3.1.2 Border mirroring

Since the U-Net uses unpadded convolutions the output segmentation is smaller than the input image. Therefore, borders were added to the input image to give the output segmentation the desired size. Information in the borders where extrapolated by mirroring. The below image shows the result of mirroring borders. The mirrored borders are shaded for clarity and the lighter are is the covered area for the output segmentation map.

Figure 3.2: Shows a training image with mirrored borders from the Potsdam dataset. The shaded area will be lost due to unpaded convolutions.

3.2 Weighting the cross entropy loss

Using the weighting definition 2.12 the cross entropy loss was weighted based on object separation. Initially ω_c was set to reflect the class imbalance as defined in (2.9) or (2.10). This led the model to achieve a good recall but a low precision and F1 score. The values $\omega_c = 2$, $\omega_0 = 10$ and $\sigma = 3$ was found to give a good F1 score while also enforcing better object separation. Below is the binary labels and the corresponding weight map for the given values.

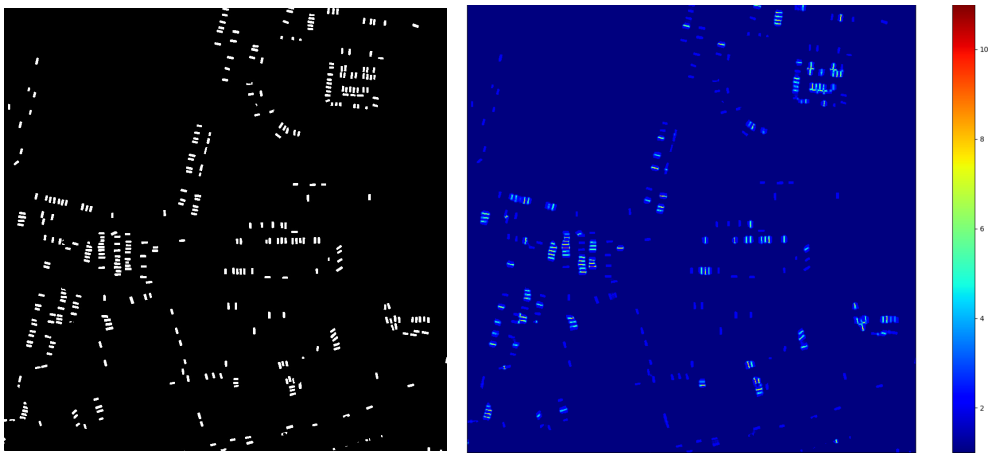

Figure 3.3: Shows the ground truth labels and the pixel weight map for a training image.

3.3 Adding an adversarial loss term

For the discriminative network a ResNet will be used. The ResNet is chosen over the VGG due to its slightly better performance and computational cost [19] as well as its flexible architecture which makes it easy to change the depth or the input size. The input to the discriminative network will be the satellite image concatenated with the ground truth or generated segmentation as showed in the image below.

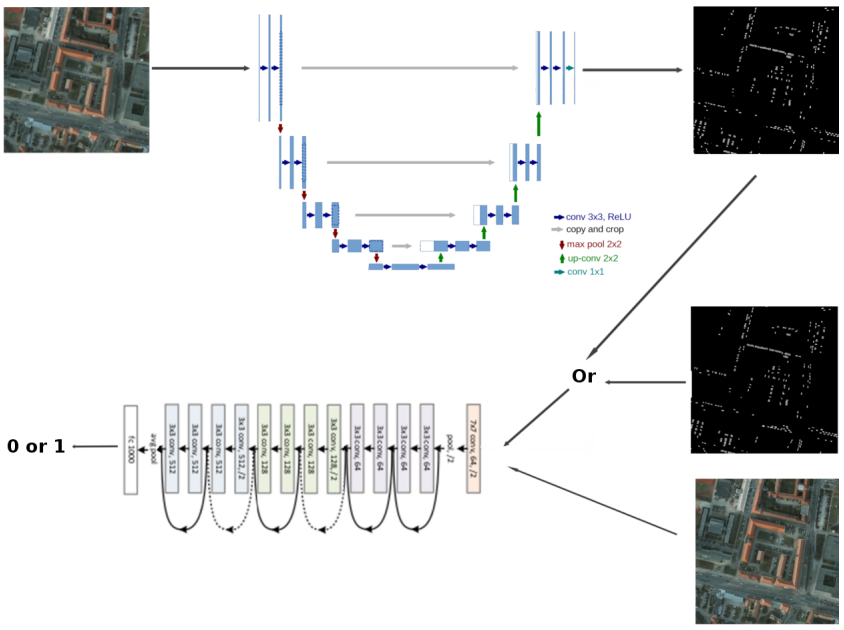

Figure 3.4: Shows the sarchitecture of the segmentation network with an adversarial loss at training time.

A patch-GAN structure will be used where the discriminator only has a view of 100×100 pixels at once. The loss for the discriminator will be the average loss over all patches. The reason for this is that a patch-GAN with a smaller window was more stable during training compared to a patch-GAN with a larger window or an image-GAN. This also made it easy to perform batch normalization over the batch of patches extracted from one image.

Following the example of [13, 25] the term $-\ell_{bce}(D(G(x), y), 0)$ is replaced with $+\ell_{bce}(D(G(x), y), 1)$ in equation 2.4. Hence instead of minimizing the probability of the discriminative network to predict the generated map to be synthetic we maximize the probability of predicting the generated map as ground truth. The reason for this is that it leads a stronger gradient for the discriminator when making predictions on ground truth and generated maps. The objective function for the GAN then becomes.

$$\mathcal{L}_{bce}(G, D) = \lambda(\ell_{bce}(D(x, G(x)), 1) - \ell_{bce}(D(x, y), 1)) \tag{3.1}$$

Finally we define our minimax game as.

$$G^* = argmin_G[\ell_{pxl}(G(x), y) + max_D[\lambda \mathcal{L}(G, D)]] \tag{3.2}$$

Here we set $\lambda = 0.01$ since this value gave a low importance to the adversarial loss early in the training but a larger importance at later stages when the cross entropy loss had decreased significantly. This led to faster convergence of the network while training. The model converged a lot slower with a larger value on λ since the adversarial network would place too much importance on one specific difference early in training and completely disregard other differences. For example, the discriminator might learn the average size of a vehicle and solely base its decision on this. The generator would then be able to decrease its loss significantly by only chopping up objects into vehicle sized object and disregarding the nature of the object.

Since [25] showed that there were no significant disadvantage of concatenating the generated maps with the satellite image in the basic manner opposed to the product or scaling method this was the initial approach. However in our experiments the discriminative network quickly learned to spot the difference between the discrete ground truth labels and the continuous generated labels and spent all of it's time teaching the generative network to draw discrete boundaries and almost completely disregarded the initial segmentation task. To force the discriminative network to learn a more productive loss function the ground truth labels were first smoothed before being fed into the discriminator. The smoothing were performed so that the ground truth labels should have the same smoothness in border pixels as the generated segmentation maps without an adversarial loss. This forced the discriminative network to learn a more productive loss function and greatly improved performance.

Initially both models were updated at each step. However it turned out that the network complexities poorly matched each other and one network always ended up outperforming the other. For the adversarial networks a ResNet with depth 14, 18, 34, 50 and 101 was tried but for all the configurations one of the networks drastically outperformed the other and training diverged as a result. To more easily be able to monitor the learning of the two networks the loss functions where changed

to.

$$\mathcal{L}(G) = \ell_{pxl}(G(x), y) + \lambda \ell_{bce}(D(x, G(x)), 1) \tag{3.3}$$

$$\mathcal{L}(D) = \ell_{bce}(D(x, G(x)), 0) + \ell_{bce}(D(x, y), 1) \tag{3.4}$$

Here both networks tries to minimize its respective loss function. An alternating training regime described below could now be used.

```
for epoch in epochs:
  update_generative_network
  if epoch % check_discriminative_network == 0:
    while discriminative_loss > cutoff:
      update_discriminative_network
```

Here % is the modulus operator and $check\_discriminator = 20$ and $cutoff = 1.0$. Since we stop training the discriminative network when its loss goes below the cutoff value the discriminative network is not able to significantly outperform the generative network. A cutoff value of 1.0 indicates that the discriminative network makes a correct prediction approximately 60% of the time.

Chapter 4

Result

4.1 The datasets

The datasets chosen for evaluation are the ISPRS Potsdam segmentation dataset and the Vedai vehicle detection dataset. The Potsdam dataset was chosen since it is over urban areas with high car densities. This corresponds well with the application of interest which is counting cars in car parks which usually have densities of vehicles as well. The Potsdam images also had a very high resolution which allowed them to be down sampled to match satellite resolution easily. The Vedai dataset was chosen since this is a well known vehicle detection dataset. This allowed the proposed model to be evaluated against previous well known models. The Vedai dataset also has a very low car density which lets us test generality of the proposed model for vehicle detection over different settings.

4.1.1 The ISPRS Potsdam semantic dataset

The ISPRS Potsdam dataset is a two dimensional semantic segmentation dataset [1]. The dataset has six classes, impervious surfaces, buildings, low, vegetation, trees, cars and clutter/background. The images are are of the TIFF format and has 6000×6000 size with a resolution of 5 cm per pixel. There are 24 images for training and validation and 14 for testing. The images are rgb or infra-red together with rgb. There is also a height data channel. The ground truth segmentation for the test images are not released and the segmentation maps have to be sent to ISPRS for evaluation. Below is an example of the training data from

the ISPRS Potsdam segmentation dataset.

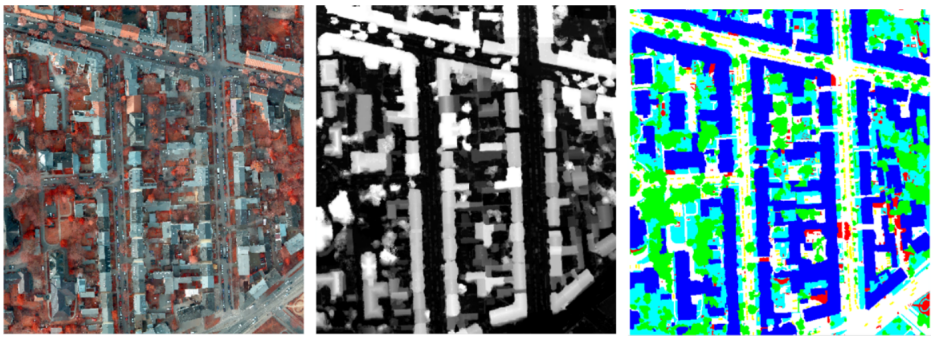

Figure 4.1: An example image from the Potsdam dataset with the rgb channel, the height data channel and the ground truth segmentation map.

The evaluation metric for the individual classes is pixel wise F1 score and the overall performance is measured by pixel accuracy.

$$Precision = \frac{True\ positive}{True\ Positive + False\ poitive} \tag{4.1}$$

$$Recall = \frac{True\ positive}{True\ Positive + False\ negative} \tag{4.2}$$

$$F1\ score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{4.3}$$

This work focuses on detecting vehicles in low cost images so the images will be down sampled to a size of 1000×1000 pixels which corresponds a resolution of 30 cm per pixel. This is done to match the resolution of commercial satellites such as DigitalGlobes WorldView 3 and 4 [51]. One important factor to keep in mind is that this is a multi class segmentation dataset and not a vehicles detection dataset. In several of the images it is possible to spot vehicles through trees without leaves but these are not segmented as vehicles but as trees. This makes it a harder challenge for the segmentation network since it must learn to differentiate between very similar objects.

Figure 4.2: Figure shows the ground truth labels and the pixel weight map for a training image

4.1.2 The VEDAI dataset

The VEDAI dataset [32] consists of 9 different classes, these classes and the number of objects are given in the table below.

Classes	Number
Car	1340
Pick-up	950
Truck	300
Plane	47
Boat	170
Camping car	390
Tractor	190
Vans	100
Other	100

To make the results comparable with the extensive research of different methods done by [62] the classes plane, boat tractor van and other were removed due to scarcity of data. The annotations for each target consists on the centre coordinates of the target, the angle of the center line of the bounding box as well as the corners of the bounding box. The bounding box fits the target closely so no extra information is given on the sides of the target. The evaluation metrics on the VEDAI dataset are precision recall and F1-score:

The dataset was divided into three parts where 927 images were used for training, 100 for validation and 240 for testing. The dataset was transformed into a semantic dataset by assigning the pixels of the bounding boxes for remaining classes to the foreground class and the rest of the pixels to the background class. A small ”artificial” separation was introduced around each bounding box so that all bounding boxes would be separated in the ground truth segmentation.

The dataset is severely imbalanced and around 99.3 % of all pixels belongs to the background class. To deal with this the training data set was first filtered to make it more balanced. From the 927 training images only those which contained five or more vehicles where selected, this resulted in 166 images. Those images where used for the initial part of training and when the network started overfitting the network was finally trained on all of the training images. This training procedure removed instabilities early on in training where the network otherwise where inclined to predict everything to the background class and set all weights to zero.

4.2 Weighting the loss function

The weighted cross entropy loss enforced much better object separation on the validation dataset. Below are cherry picked examples from high vehicle density areas in the validation images.

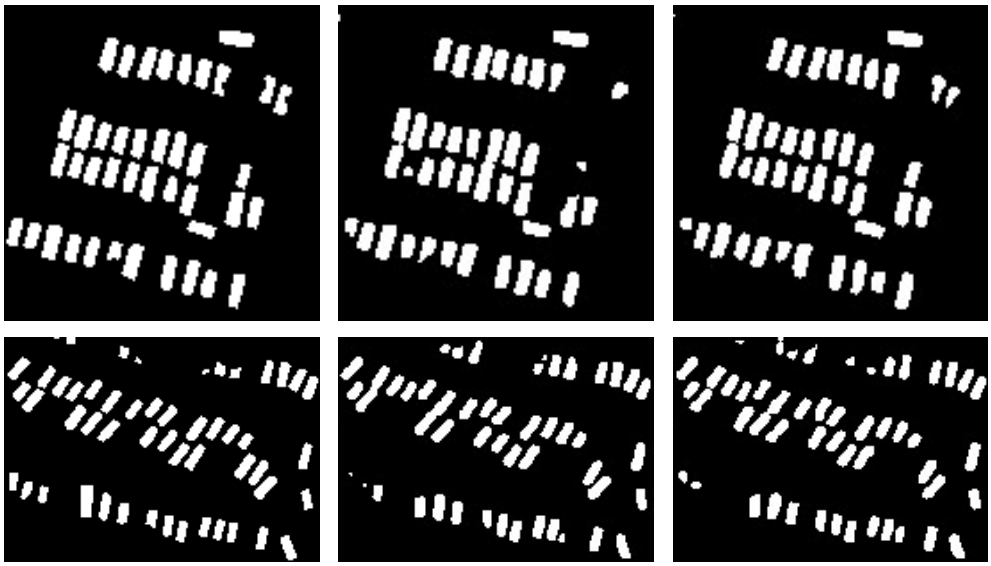

Figure 4.3: Shows the ground truth segmentation, the discrete prediction without a weighted loss and the discrete prediction with a weighted loss. The weighted loss has clearly enforced better segmentation around nearby vehicles.

The weighted loss achieved better object separation while still obtaining comparable pixel wise F1 score. Below is the pixel wise F1 score on the training and validation set during training.

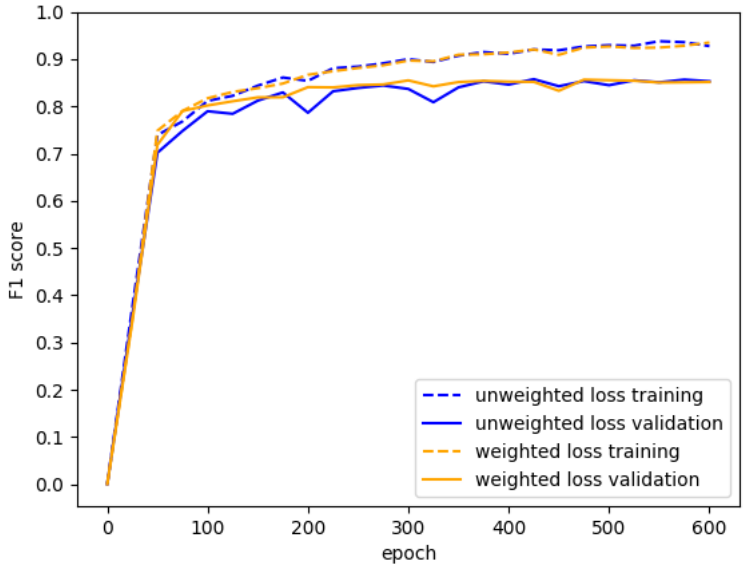

Figure 4.4: Shows the pixel wise F1 score of the network with the weighted and unweighted loss for the training and validation dataset.

The proposed model's car counting capabilities where also evaluated on the validation dataset. Here we define the true number of cars as the number of connected components in the ground truth segmentation map. The predicted number of cars is the number of connected components in the predicted segmentation map.

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

4.3 Adding an adversarial loss term

Since the GAN game becomes unstable with sparse gradients the Relu activation function was therefore replaced with a leaky Relu activation in both the generative and discriminative network and max pooling was replaced by mean pooling in the generative network. The U-NET already uses 2D transposed convolution for up sampling so this was kept as in the original network. This changes greatly stabilized the training. However I still managed to train a GAN using the origin U-Net with sparse gradients with a carefully chosen learning rate decay.

This training was very unstable and even a small change in the learning rate would cause the training to diverge.

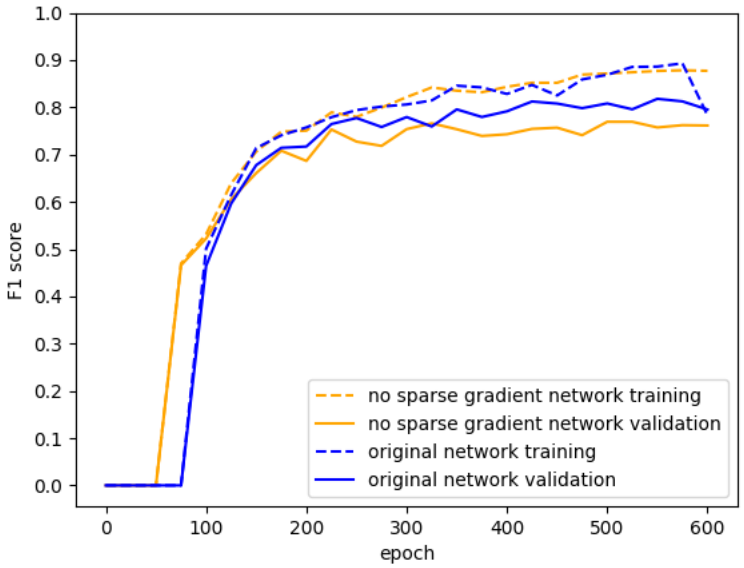

Figure 4.5: The pixel wise F1 score for the for the training and validation dataset with and without sparse gradients functions.

Both networks had a similar performance on the training data but the original network over fitted less and performed better on the validation data. Hence the original U-Net network was chosen as the generative network for the following comparisons although it was harder to train.

The segmented images looks very similar on a large scale but if we look closely there are differences between the two networks. In the below images we displays cherry picked high density areas of vehicles. The segmentation maps are displayed as continuous values where the brightness of each pixel is the probability of that pixel belonging to a car. This is done to make better comparisons between the two networks. To obtain a discrete prediction an argmax between this layer and the background class can be preformed.

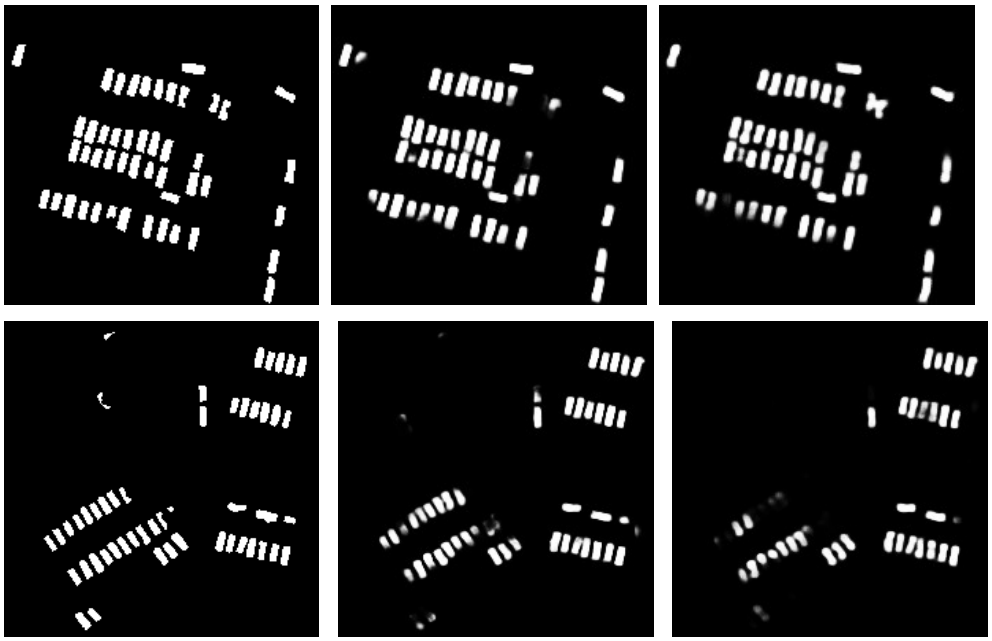

Figure 4.6: Shows the ground truth segmentation, the continuous prediction without and with and adversarial loss.

The adversarial network enforced more assertive predictions and cars were more likely to be fully segmented or completely left out. However the segmented objects seemed to be more loosely conditioned on the input image then for the network without an adversarial loss. Other differences were that the GAN network was more inclined to chop up false positives into vehicle like objects. It also had a tenancy to separate long vehicles such as trucks into smaller car like parts. The adversarial segmentation maps looks worse by visual inspection and the adversarial loss did not improve the pixel wise F1 score on the validation dataset.

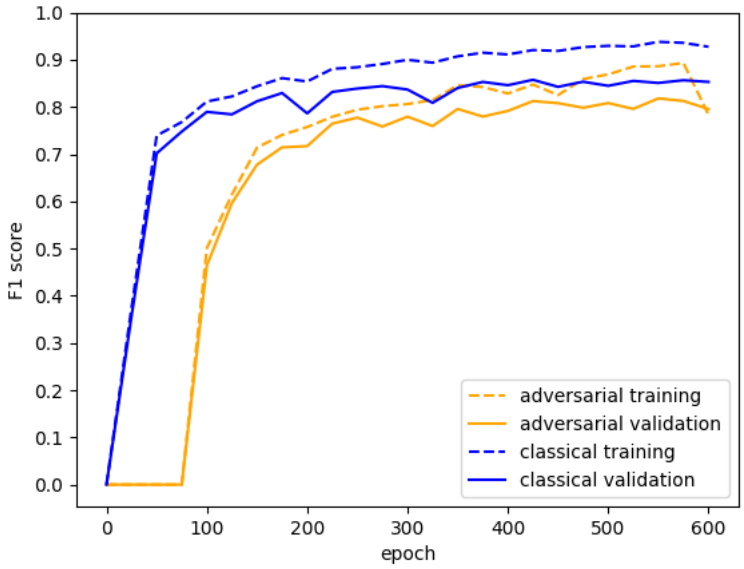

Figure 4.7: Shows the pixel wise F1 score with and without an adversarial loss term on the validation dataset.

4.4 Comparison with earlier work

The network with the weighted cross entropy loss was chosen to be compared with previous models since this had the best performance on the validation dataset. Below is the comparison with the proposed model and the SBD model on the Potsdam dataset.

Comparison on the ISPRS Potsdam dataset

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 4.1: Shows the comparison between the proposed model and the Segment before you Detect (SBD) model [6] 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 wich 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. The evaluation time should therefore not be regarded as exact but as an indication of the speed difference between the two models.

The proposed model was also evaluated on the Vedai dataset using the 512×512 resolution. The proposed model was compared with the Faster R-CNN (Z&F), Faster R-CNN (VGG-16), Fast R-CNN (VGG-16) [58] and the Cascaded Convolutional Neural Networks (CCNN) [61].

Comparison on the Vedai dataset

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 4.2: Shows the comparison between the proposed model and the Faster R-CNN (Z&F), Faster R-CNN (VGG-16), Fast R-CNN (VGG-16) [58] and the Cascaded Convolutional Neural Networks (CCNN) [61] on the Vedai dataset. * The other models were evaluated by [61] 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 wich 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. The evaluation time should therefore not be regarded as exact but as an indication of the speed difference between the two models.

Chapter 5

Discussion

This thesis proposes a vehicle segmentation and detection pipeline capable of detecting and counting vehicles in satellite resolution images. The proposed model outperforms the Segment Before you Detect (SBD) pipeline [6] on the vehicle wise F1 score and prediction error by a significant margin while using less than half of the resolution for input images. The proposed model achieves this while obtaining a computational time which is less than 1% of the SBD network's computational time. The proposed model almost doubles the vehicle wise F1 score on the Vedai dataset compared to Faster R-CNN (Z&F), Faster R-CNN (VGG-16), Fast R-CNN (VGG-16) [58] and the Cascaded Convolutional Neural Networks (CCNN) [61]. It achieves this performance with a computational time which is less than a third of the second fastest model.

The pipeline needs a very small amount of training images. Only 18 training images where used to obtain a vehicle detection F1 score of 0.811 and a counting car prediction error of 1.67 %. This ensures that constructing a dataset is a low cost operation which makes it a viable option for commercial analysis of satellite imagery. It can be argued that one of the restrictions of the proposed pipeline is that only separated objects can be detected due to the nature of connected component extraction. However, by introducing "artificial" separations between touching objects on the Vedai dataset it is shown that the proposed pipeline still can learn to separate touching objects and outperforms earlier methods.

Bibliography

[1] *2D Semantic Labeling - ISPRS*. URL: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html (visited on 02/07/2018).

[2] Nassim Ammour et al. "Deep Learning Approach for Car Detection in UAV Imagery". en. In: *Remote Sensing* 9.4 (Mar. 2017), p. 312. DOI: 10.3390/rs9040312. URL: http://www.mdpi.com/2072-4292/9/4/312 (visited on 02/07/2018).

[3] Assaf Arbelle and Tammy Riklin Raviv. "Microscopy Cell Segmentation via Adversarial Neural Networks". In: *arXiv:1709.05860 [cs]* (Sept. 2017). URL: http://arxiv.org/abs/1709.05860 (visited on 02/07/2018).

[4] Anurag Arnab et al. "Higher Order Conditional Random Fields in Deep Neural Networks". In: *arXiv:1511.08119 [cs]* (Nov. 2015). URL: http://arxiv.org/abs/1511.08119 (visited on 02/20/2018).

[5] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images". en. In: *Remote Sensing* 9.4 (Apr. 2017), p. 368. DOI: 10.3390/rs9040368. URL: http://www.mdpi.com/2072-4292/9/4/368 (visited on 02/07/2018).

[6] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. "On the usability of deep networks for object-based image analysis". In: *arXiv:1609.06845 [cs]* (Sept. 2016). arXiv: 1609.06845. URL: http://arxiv.org/abs/1609.06845 (visited on 04/27/2018).

[7] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *arXiv:1511.00561 [cs]* (Nov. 2015). URL: http://arxiv.org/abs/1511.00561 (visited on 03/08/2018).

[8] J. A. J. Berni et al. "Thermal and Narrowband Multispectral Remote Sensing for Vegetation Monitoring From an Unmanned Aerial Vehicle". In: *IEEE Transactions on Geoscience and Remote Sensing* 47.3 (Mar. 2009), pp. 722–738. ISSN: 0196-2892. DOI: 10.1109/TGRS.2008.2010457.

[9] Soumith Chintala. *ganhacks: starter from "How to Train a GAN?" at NIPS2016*. original-date: 2016-12-09T16:09:27Z. Apr. 2018. URL: https://github.com/soumith/ganhacks (visited on 04/04/2018).

[10] Alberto Garcia-Garcia et al. "A Review on Deep Learning Techniques Applied to Semantic Segmentation". In: *arXiv:1704.06857 [cs]* (Apr. 2017). URL: http://arxiv.org/abs/1704.06857 (visited on 03/09/2018).

[11] Ross Girshick. "Fast R-CNN". In: *arXiv:1504.08083 [cs]* (Apr. 2015). URL: http://arxiv.org/abs/1504.08083 (visited on 03/08/2018).

[12] Ian Goodfellow. "NIPS 2016 Tutorial: Generative Adversarial Networks". In: *arXiv:1701.00160 [cs]* (Dec. 2016). URL: http://arxiv.org/abs/1701.00160 (visited on 02/07/2018).

[13] Ian J. Goodfellow et al. "Generative Adversarial Networks". In: *arXiv:1406.2661 [cs, stat]* (June 2014). URL: http://arxiv.org/abs/1406.2661 (visited on 02/20/2018).

[14] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *arXiv:1512.03385 [cs]* (Dec. 2015). URL: http://arxiv.org/abs/1512.03385 (visited on 03/08/2018).

[15] Kaiming He et al. "Identity Mappings in Deep Residual Networks". In: *arXiv:1603.05027 [cs]* (Mar. 2016). URL: http://arxiv.org/abs/1603.05027 (visited on 03/08/2018).

[16] Kaiming He et al. "Mask R-CNN". In: *arXiv:1703.06870 [cs]* (Mar. 2017). URL: http://arxiv.org/abs/1703.06870 (visited on 03/09/2018).

[17] Ashley C. Holt et al. "Object-based detection and classification of Vehicles from high-resolution aerial photography". English. In: *Photogrammetric Engineering and Remote Sensing* 75.7 (July 2009), pp. 871–880. ISSN: 0099-1112. URL: https://iths.pure.elsevier.com/en/publications/object-based-detection-and-classification-of-vehicles-from-high-r (visited on 02/07/2018).

[18] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: *arXiv:1611.07004 [cs]* (Nov. 2016). URL: http://arxiv.org/abs/1611.07004 (visited on 02/18/2018).

[19] Justin Johnson. *cnn-benchmarks: Benchmarks for popular CNN models*. original-date: 2016-07-13T06:46:20Z. Apr. 2018. URL: https://github.com/jcjohnson/cnn-benchmarks (visited on 04/27/2018).

[20] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[22] Christian Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *arXiv:1609.04802 [cs, stat]* (Sept. 2016). URL: http://arxiv.org/abs/1609.04802 (visited on 03/09/2018).

[23] Ruirui Li et al. "DeepUNet: A Deep Fully Convolutional Network for Pixel-level Sea-Land Segmentation". In: *arXiv:1709.00201 [cs]* (Sept. 2017). URL: http://arxiv.org/abs/1709.00201 (visited on 03/08/2018).

[24] Tsung-Yi Lin et al. "Microsoft COCO: Common Objects in Context". In: *arXiv:1405.0312 [cs]* (May 2014). URL: http://arxiv.org/abs/1405.0312 (visited on 03/09/2018).

[25] Pauline Luc et al. "Semantic Segmentation using Adversarial Networks". In: *arXiv:1611.08408 [cs]* (Nov. 2016). URL: http://arxiv.org/abs/1611.08408 (visited on 02/07/2018).

[26] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *arXiv:1411.1784 [cs, stat]* (Nov. 2014). URL: http://arxiv.org/abs/1411.1784 (visited on 02/18/2018).

[27] T. Moranduzzo, M. L. Mekhalfi, and F. Melgani. "LBP-based multiclass classification method for UAV imagery". In: *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. July 2015, pp. 2362–2365. DOI: 10.1109/IGARSS.2015.7326283.

[28] T. Moranduzzo and F. Melgani. "Automatic Car Counting Method for Unmanned Aerial Vehicle Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 52.3 (Mar. 2014), pp. 1635–1647. ISSN: 0196-2892. DOI: 10.1109/TGRS.2013.2253108.

[29] Deepak Pathak et al. "Context Encoders: Feature Learning by Inpainting". In: *arXiv:1604.07379 [cs]* (Apr. 2016). URL: http://arxiv.org/abs/1604.07379 (visited on 02/19/2018).

[30] Andrei Polzounov et al. "Right whale recognition using convolutional neural networks". In: *arXiv:1604.05605 [cs]* (Apr. 2016). URL: http://arxiv.org/abs/1604.05605 (visited on 03/09/2018).

[31] Md Atiqur Rahman and Yang Wang. "Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation". In: *Advances in Visual Computing*. Ed. by George Bebis et al. Cham: Springer International Publishing, 2016, pp. 234–244. ISBN: 978-3-319-50835-1.

[32] Sébastien Razakarivony and Frédéric Jurie. "Vehicle Detection in Aerial Imagery : A small target detection benchmark". In: *Journal of Visual Communication and Image Representation, Elsevier* (Mar. 2015). URL: https://hal.archives-ouvertes.fr/hal-01122605 (visited on 02/07/2018).

[33] Scott Reed et al. "Generative Adversarial Text to Image Synthesis". en. In: (May 2016). URL: https://arxiv.org/abs/1605.05396 (visited on 02/18/2018).

[34] Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *arXiv:1506.01497 [cs]* (June 2015). URL: http://arxiv.org/abs/1506.01497 (visited on 03/08/2018).

[35] Mina Rezaei et al. "Conditional Adversarial Network for Semantic Segmentation of Brain Tumor". In: *arXiv:1708.05227 [cs]* (Aug. 2017). URL: http://arxiv.org/abs/1708.05227 (visited on 02/19/2018).

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *arXiv:1505.04597 [cs]* (May 2015). URL: http://arxiv.org/abs/1505.04597 (visited on 02/19/2018).

[37] M. H. O. Ruhe, C. Dalaff, and R. D. Kuhne. "Traffic monitoring and traffic flow measurement by remote sensing systems". In: *Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems*. Vol. 1. Oct. 2003, 760–764 vol.1. DOI: 10.1109/ITSC.2003.1252053.

[38] W. Sakla, G. Konjevod, and T. N. Mundhenk. "Deep Multi-modal Vehicle Detection in Aerial ISR Imagery". In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Mar. 2017, pp. 916–923. DOI: 10.1109/WACV.2017.107.

[39] Alexander G. Schwing and Raquel Urtasun. "Fully Connected Deep Structured Networks". In: *arXiv:1503.02351 [cs]* (Mar. 2015). URL: http://arxiv.org/abs/1503.02351 (visited on 02/20/2018).

[40] Pierre Sermanet et al. "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks". In: *arXiv:1312.6229 [cs]* (Dec. 2013). URL: http://arxiv.org/abs/1312.6229 (visited on 03/09/2018).

[41] Evan Shelhamer, Jonathan Long, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *arXiv:1605.06211 [cs]* (May 2016). URL: http://arxiv.org/abs/1605.06211 (visited on 02/19/2018).

[42] Wenzhe Shi et al. "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network". In: *arXiv:1609.05158 [cs, stat]* (Sept. 2016). arXiv: 1609.05158. URL: http://arxiv.org/abs/1609.05158 (visited on 04/26/2018).

[43] Karen Simonyan and Andrew Zisserman. "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *arXiv:1406.2199 [cs]* (June 2014). URL: http://arxiv.org/abs/1406.2199 (visited on 03/09/2018).

[44] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv:1409.1556 [cs]* (Sept. 2014). URL: http://arxiv.org/abs/1409.1556 (visited on 03/08/2018).

[45] Jaemin Son, Sang Jun Park, and Kyu-Hwan Jung. "Retinal Vessel Segmentation in Fundoscopic Images with Generative Adversarial Networks". In: *arXiv:1706.09318 [cs]* (June 2017). URL: http://arxiv.org/abs/1706.09318 (visited on 02/07/2018).

[46] Nasim Souly, Concetto Spampinato, and Mubarak Shah. "Semi and Weakly Supervised Semantic Segmentation Using Generative Adversarial Network". In: *arXiv:1703.09695 [cs]* (Mar. 2017). URL: http://arxiv.org/abs/1703.09695 (visited on 03/09/2018).

[47] George Stockman and Linda G. Shapiro. *Computer Vision*. 1st. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN: 0-13-030796-3.

[48] Carole H. Sudre et al. "Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations". In: *arXiv:1707.03237 [cs]* 10553 (2017), pp. 240–248. DOI: 10.1007/978-3-319-67558-9_28. URL: http://arxiv.org/abs/1707.03237 (visited on 03/08/2018).

[49] Christian Szegedy et al. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *arXiv:1602.07261 [cs]* (Feb. 2016). URL: http://arxiv.org/abs/1602.07261 (visited on 03/08/2018).

[50] K. Uto et al. "Characterization of Rice Paddies by a UAV-Mounted Miniature Hyperspectral Sensor System". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 6.2 (Apr. 2013), pp. 851–860. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2013.2250921.

[51] *WorldView-3 Satellite Sensor | Satellite Imaging Corp*. URL: https://www.satimagingcorp.com/satellite-sensors/worldview-3/ (visited on 04/16/2018).

[52] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. "Wider or Deeper: Revisiting the ResNet Model for Visual Recognition". In: *arXiv:1611.10080 [cs]* (Nov. 2016). URL: http://arxiv.org/abs/1611.10080 (visited on 03/08/2018).

[53] Yuan Xue et al. "SegAN: Adversarial Network with Multi-scale L_1 Loss for Medical Image Segmentation". In: *arXiv:1706.01805 [cs]* (June 2017). URL: http://arxiv.org/abs/1706.01805 (visited on 03/09/2018).

[54] Dong Yang et al. "Automatic Liver Segmentation Using an Adversarial Image-to-Image Network". In: *arXiv:1707.08037 [cs]* (July 2017). URL: http://arxiv.org/abs/1707.08037 (visited on 02/19/2018).

[55] Jiahui Yu et al. "UnitBox: An Advanced Object Detection Network". In: *arXiv:1608.01471 [cs]* (2016), pp. 516–520. DOI: 10.1145/2964284.2967274. URL: http://arxiv.org/abs/1608.01471 (visited on 03/08/2018).

[56] Sergey Zagoruyko and Nikos Komodakis. "Wide Residual Networks". In: *arXiv:1605.07146 [cs]* (May 2016). URL: http://arxiv.org/abs/1605.07146 (visited on 03/08/2018).

[57] Matthew D. Zeiler and Rob Fergus. "Visualizing and Understanding Convolutional Networks". In: *arXiv:1311.2901 [cs]* (Nov. 2013). arXiv: 1311.2901. URL: http://arxiv.org/abs/1311.2901 (visited on 04/17/2018).

[58] 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).

[59] Y. Zhao et al. "Classification of High Spatial Resolution Imagery Using Improved Gaussian Markov Random-Field-Based Texture Features". In: *IEEE Transactions on Geoscience and Remote Sensing* 45.5 (May 2007), pp. 1458–1468. ISSN: 0196-2892. DOI: 10.1109/TGRS.2007.892602.

[60] Shuai Zheng et al. "Conditional Random Fields as Recurrent Neural Networks". In: *arXiv:1502.03240 [cs]* (Dec. 2015), pp. 1529–1537. DOI: 10.1109/ICCV.2015.179. URL: http://arxiv.org/abs/1502.03240 (visited on 02/20/2018).

[61] 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).

[62] Jiandan Zhong, Tao Lei, and Guangle Yao. "Robust Vehicle Detection in Aerial Images Based on Cascaded Convolutional Neural Networks". In: *Sensors (Basel, Switzerland)* 17.12 (Nov. 2017). ISSN: 1424-8220. DOI: 10.3390/s17122720. URL: https://www.

ncbi . nlm . nih . gov / pmc / articles / PMC5751529/ (visited on 02/18/2018).