This is the English title

OSQUAR STUDENT

Master in Computer Science
Date: April 19, 2018
Supervisor: Lotta Larsson
Examiner: Lennart Bladgren
Swedish title: Detta är den svenska översättningen av titeln
School of Electrical Engineering and Computer Science

Abstract

English abstract goes here.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Sammanfattning

Träutensilierna i ett tryckeri äro ingalunda en oviktig faktor, för trevnadens, ordningens och ekonomiens upprätthållande, och dock är det icke sällan som sorgliga erfarenheter göras på grund af det oförstånd med hvilket kaster, formbräden och regaler tillverkas och försäljas Kaster som äro dåligt hopkomna och af otillräckligt.

Contents

Chapter 1

Introduction

Convolutionla neural networks (CNN) have had great success for computer vision tasks [38, 54, 40, 19]. The success is possible 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 [18] to multiple object detection and bounding box prediction [22]. CNN networks have also had great success in different segmentation task [9]. 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. Much work has gone into constructing well suited loss functions for segmentation [34, 52, 29]

Often the loss function fails to enforce important properties such as spatial contiguity in the segmentation maps [23] or proper spatial seperation [5]. Conditional markov random fields (CRFs) have been a popular post processing step to ensure spatial contiguity in segmentation maps [56]. 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 [11]. 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 [20, 17]. The success comes from the gen-

eral approach where the network can learn it's own loss function which has proven beneficial for many tasks where a effective loss function is hard to express simply.

GANs have also been applied to image segmentation and has shown to give an increased performance [23, 43]. 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 [43, 50, 51, 33, 3].

Analysis of aerial 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 [35, 26] vegetation monitoring [47, 8], urban area monitoring [25], water reserve capacity monitoring, generate new maps [17] and even to detect endangered whales [28]. It can also be used to predict market trends since 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.

Much research has been performed investigating object detection an more specifically vehicle detection in aerial imagery [2, 16, 5, 30, 58, 6, 36]. However object detection in aerial 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 image. 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[16]. This achieved good performance but at a great computational cost since redundant computation of low-level filters for overlapping patches had to be performed [23]. 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]. There have been work of two stage pipelines where a CNN segmentor first segments the image and the segmentation patchet are then fed into a object classification network which classifies the patches and predict bounding boxes [5]. A more advanced approach of such networks is

the two stage fast region-based convolutional network (fast R-CNN) [32, 10] or mask region-based convolutional network (mask R-NN) [15] which computes the bounding boxes predictions using a second stage region proposal network (RPN) on internal convolutional feature maps. These networks decreases the computational cost compared to previous networks. 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.

In this work we propose a generative adversarial segmentation network. The aim is to learn a better loss function for the segmentor so that vehicles can be detected by only performing connected component extraction on the segmentation maps. Since the adversarial part of the network only is used while training and connected component extraction is very computational efficient this guarantees computational efficient pipeline to detect and segment vehicles in aerial images compared to fast R-NN and mask R-NN which uses a two stage approach.

1.1 Research Question

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Chapter 2

Related work

2.1 Generative adversarial networks

Goodfellow *et al* [11] first proposed the generative adversarial network (GAN). The network consists of two parts, a generator and a discriminator. The generators 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 [31]. Since we in this work are only interested in segmentation where a deterministic mapping from the image to the segmentation map is desired we will not input any noise vector 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 [17]. We first define the binary cross entropy loss.

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

Here $\hat{z}$ is the prediction and z 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 stands for the discriminating network and G for the generating network. 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 [24]. 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 semantic maps but semantic maps corresponding to the input image. The objective function will in this case be given by and the networks is trained with the minimax equation (2.3).

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

It has been shown that a multi term loss function can improve the quality of the generator [27, 17]. 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.

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

Here y is the ground truth segmentation maps while $\hat{y}$ is the predicted maps. The discriminators objective is unchanged but the generator now has to fool the discriminator as well as minimizing the distance to the ground truth.

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

Here λ is just 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 minmax 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 to strong it will be able to differentiate between the generated samples and the true samples very effectively no matter what the the generator does. The derivative of the loss will then be very small and we will have a problem with vanishing gradients for the generative network. In practice it is very hard to achieve this balance but some comon tricks is making the generative or discriminative networks more or less complex as well as training them at different amount on steps each iteration. [??] also mentions some other tricks to stabilize gan training. Modify the loss function Do not mix samples in a mini batch Only use samples of one type i the mini batch. Use batch normalization as well Avoid sparse gradients The minmax game becomes mere unstable with sparse gradients. This means that we should use leaky Relu 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 [??] can be used. Smooth target labels so the discriminator will not be able to differentiate between the generated and ground truth labels by checking if they are continuous or not.

2.2 Segmentor networks

For the generative part of a gan a segmentor network is needed. This network takes a image as an input and produces segmentation maps. A fully convolotional network (FCN) was first proposed Shelhamer and Long *et al* [39]. 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 is that by the authors were that fully connected layers can be viewed as convolutions with kernels that cover their entire input region. Hencea 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* [34] 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 upsampled layers. Thess skip connections are crucial to segmentation tasks as the initial feature maps maintain low-level features such that can be properly exploited for accurate segmentation.The network has been shown to produce high accuracy results even on small sized datasets [42, 34, 17, 50, 51]. Ronneberger *et al* [34] attributes this to the networks structure which creates internal data augmentation.

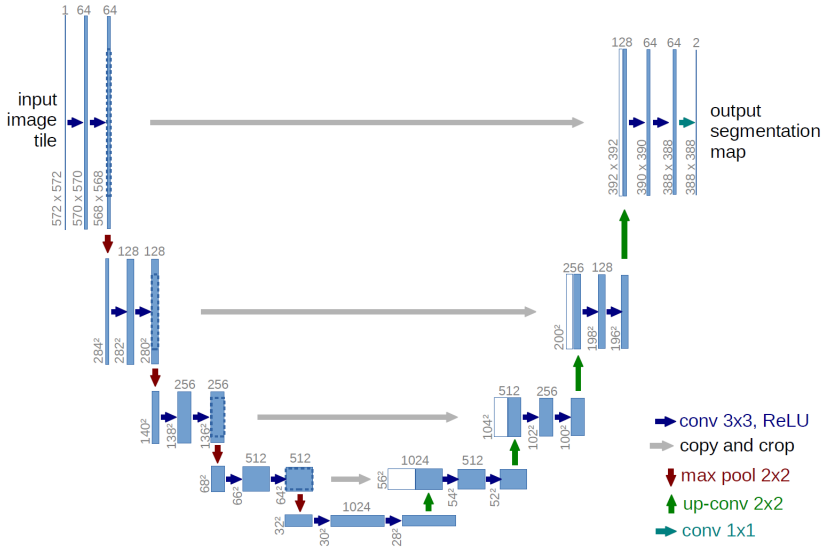

Figure 2.1: The initial u-net architecture proposed by Ronneberger *et al* [34]

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 the context is extrapolated by mirroring the image [21]. It is also important that we choose an initial image size so the activation maps length is even through all layers of the network. [34]. In theory the u-net can handle images of any size

but in practice we have memory limitations for the GPU. Segmentation it therefore done on patches and the output prediction can be directly stitched together without any overlapping.

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 [41] and the ResNet networks [13].

The VGG networks in form of the sixteen layer VGG16 or the 19 layer VGG19 have have performed very well on a wide variety of tasks such as classification [41]. They have been used inside Fast Region-based Convolutional Networks (fast R-CNN) [32, 10] to generate object activation maps for the reginal 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]. The VGG16 network takes inputs of images

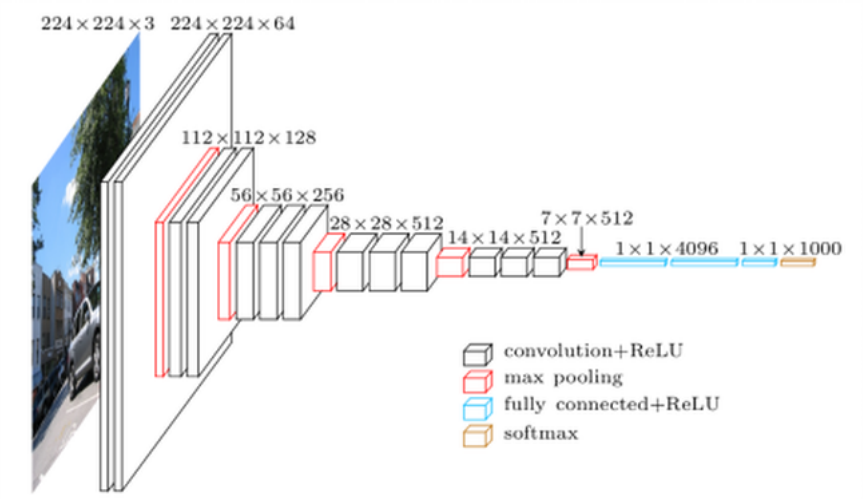

Figure 2.2: The VGG16 architecture proposed by [41]

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.

ResNet is another widely popular classification network. It has performed very well on classification tasks [14, 46] It has also been shown that wider residual networks are more memory efficient while still obtaining comparable performance [49, 53]. The success behind the ResNet

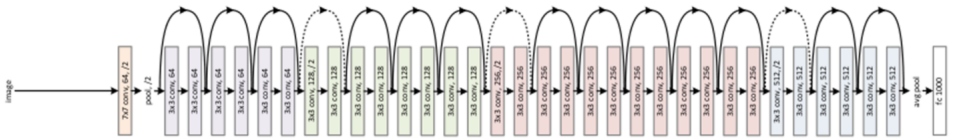

Figure 2.3: The 34 layer deep ResNet architecture [41]

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 block. 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 [13]. Hence we get e 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 aerial 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 [45]. Below is the weighted cross entropy loss for a two class

segmentation problem.

$$\mathcal{L}_{wce} = -\omega_c y log(\hat{y}) + (1-y) log(1-\hat{y}) \tag{2.8}$$

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.

$$w_c = \frac{N - \sum_n \hat{y}_{c=1}}{\sum_n \hat{y}_{c=1}} \tag{2.9}$$

Here N is the total number of pixels and $\sum_n \hat{y}$ the number of pixels defined to the foreground class.

$$w_c = \frac{\sum_n \sum_{W*H} y_{c=0}}{\sum_n \sum_{W*H} y_{c=1}} \tag{2.10}$$

Which is the sum over all of the background pixels in the training dataset divided by the foreground pixels. Another popular approach is to minimize the intersect over union loss [52, 29]

$$\mathcal{L}_{IoU} = 1 - \frac{\sum_{W*H} y \otimes \hat{y}}{\sum_{W*H} y + \sum_{W*H} \hat{y} - \sum_{W*H} y \otimes \hat{y}} \tag{2.11}$$

Here $\otimes$ operator is the pixel wise multiplication. To ensure good object seperation Ronneberger *et al* [34] proposes to scale the pixel wise loss based on the proximity to the closet two foreground objects according to.

$$\omega(x) = \omega_c + \omega_0 * epx(\frac{(-d_1(x) - d_2(x))^2}{2\sigma^2}) \tag{2.12}$$

Here $\omega_c(x)$ depends on the class frequency and could be computes 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 Connected component extraction

The most simple and intuitive method of labelling connected components in an image is the two pass algorithm [44]. The algorithm iterates

two times over the binary two dimensional image. In the first pass the algorithm assigns temporary labels and records equivalences for each foreground pixel. In the second pass the algorithm replaces each temporary label by the smallest label of its equivalence class. 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 we will only use this mode. Since the algorithm makes a constant number of passes over the image array and a constant number of neighbours are compared at each step this is a O(n) time algorithm where n is the number of pixels i the input image.

2.6 Related 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, 23]. There have been work that used mean field inference expressed as a recurrent convolutional networks to do CRF like post processing [37, 57]. Luc *et al* [23] proposed a adversarial segmentation network to enforce higher order potentials without being limited to a single class. Instead on 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 pre processing.

Son and Jung *et al* [42] 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 [17] 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 togheter with a cross entropy term preformed the best and outperformed the non adversarial segmantic network trained only with the cross entropy loss by a significant margin.

Chapter 3

The datasets

3.1 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 resolution with a resoloution 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 is 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 semantic dataset.

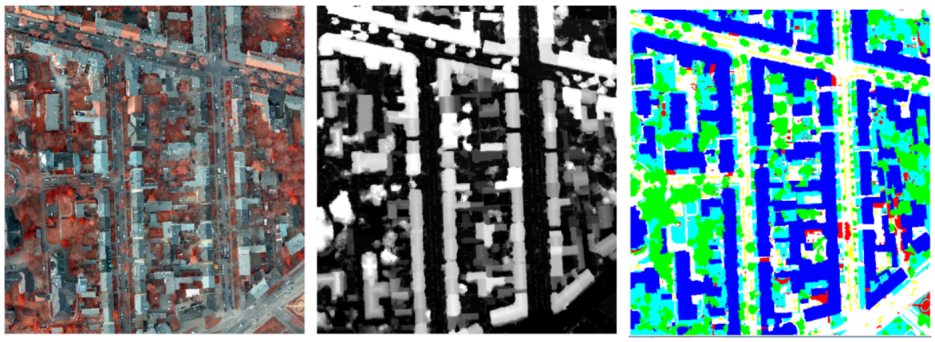

Figure 3.1: An example image from the Potsdam dataset with a: the rgb channel, b: the height data channel and c: the ground truth segmentation map.

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

$$Precision = \frac{True\ positive}{True\ Positive + False\ poitive} \tag{3.1}$$

$$Recall = \frac{True\ positive}{True\ Positive + False\ negative} \tag{3.2}$$

$$F1\ score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{3.3}$$

This work focuses on detecting vehicles in low cost images so the images will be down sampled to a resolution of 1000×1000 pixels which corresponds a resolution of 30 cm per pixel. This is done to match the resolution to the the resolution of commercial satellites such as DigitalGlobes WorldView 3 and 4 [48]. Only the car class will also be predicted. One important factor to keep in mind is that this is a multi class semantic 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 semantic network since it must learn to differentiate between very similar objects.

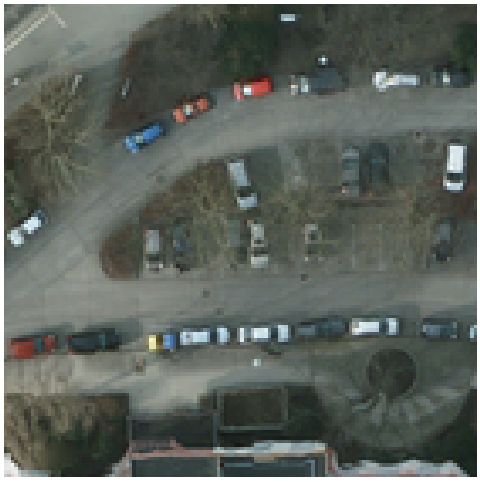

Image **(a)**, the input validation image

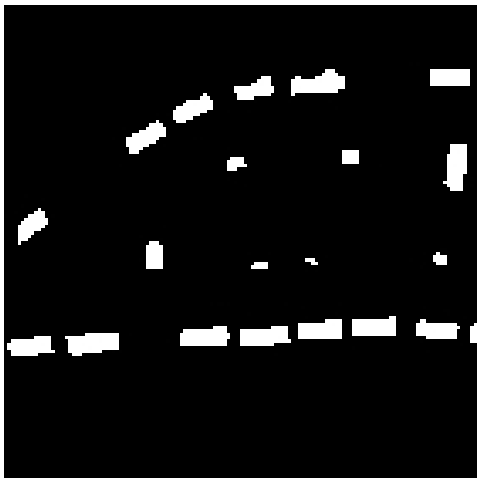

Image **(b)**, the ground truth segmentation

Figure 3.2: Figure shows the ground truth labels and the pixel weight map for a training image

3.1.1 The VEDAI dataset

The VEDAI dataset [30] 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 [58] 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. The evaluation metrics on the VEDAI dataset are precision recall and F1-score:

Chapter 4

Method

For the generator network a U-NET will be used and for the discriminator a ResNet will be used. The ResNet is choosen over the VGG due to its slightly better performance and computational cost 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. We will use a patch-Gan structure where the discriminator will only have 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 this is that an patch-Gan will a smaller window was more stable during training than 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. Using the definition of the binary cross entropy loss (2.1) and the pixel wise cross entropy loss (2.5) we can now define our loss function as.

$$\mathcal{L}(G, D) = -\ell_{bce}(D(x, y), 1) - \ell_{bce}(D(x, G(x)), 0) + \lambda \ell_{pxl}(G(x), y) \quad (4.1)$$

The generator will try to minimize this loss while the discriminator will try to maximize it. Following the example of [12, 23] and replace the term $-\ell_{bce}(D(G(x), y), 0)$ with $+\ell_{bce}(D(G(x), y), 1)$. 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 binary cross entropy

loss then becomes,

$$\mathcal{L}_{bce}(G, D) = \lambda(\ell_{bce}(D(x, G(x)), 1) - \ell_{bce}(D(x, y), 1)) \tag{4.2}$$

The objective for the network hence becomes.

$$G^* = argmin_G[max_D[\mathcal{L}_{bce}(G, D)] + \lambda\ell_{pxl}(G(x), y)] \tag{4.3}$$

Here we set $\lambda = 0.01$, 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. 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 it's decision on this. The generator would then be able to decrease it's loss significantly by only chopping up objects in vehicle sized object and disregarding the nature of the object. This led to a slow convergence of the network.

Since [23] 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 learnt 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. Below is an image of the ground truth label, the predicted segmentation using classical training and the predicted segmentation with an adversarial loss and concatenation in the basic manner. [**********'] 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.

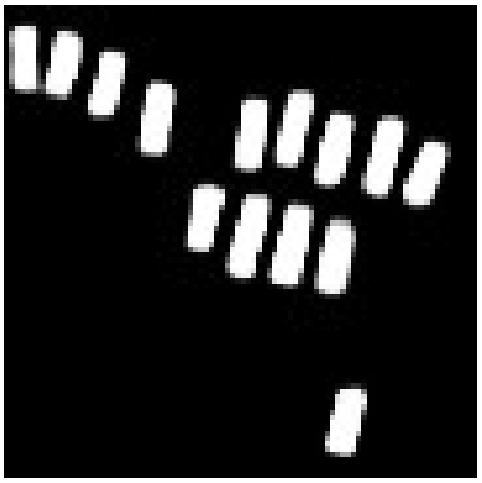

Image **(a)**, the input validation image

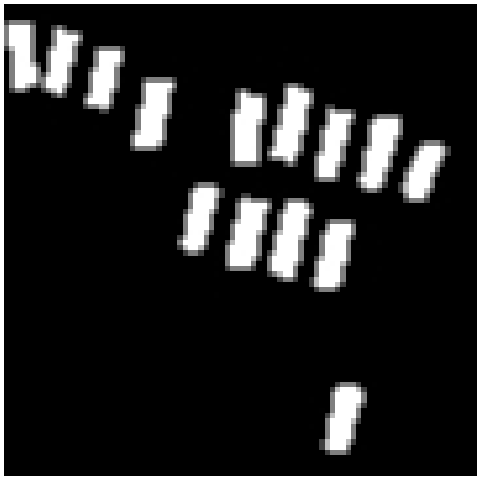

Image **(b)**, the ground truth segmentation

Figure 4.1: Figure shows the ground truth labels and the pixel weight map for a training image

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. Initially a U-NET with 64 kernels in the first layer was used. For the adversarial networks a ResNet with depth 14 18,34, 50 and 101 was tried but for all of 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.

$$\mathcal{L}_{bce}(G) = \ell_{bce}(D(x, G(x)), 1) + \lambda \ell_{pxl}(G(x), y) \tag{4.4}$$

$$\mathcal{L}_{bce}(D) = \ell_{bce}(D(x, G(x)), 0) + \ell_{bce}(D(x, y), 1) \tag{4.5}$$

Here both networks tries to minimize it's 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_network_loss > cut_off:
                        update_discriminative_network
```

Here % is the modulus operator and $check\_discriminator = 20$ and $cut\_off = 1.0$. Since we stop training the discriminative network when

it's 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 5

Result

5.1 With weight scaling

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

5.2 Comparison with earlier work

Comparison on the ISPRS Potsdam dataset

Model	Proposed Model	SBD
Resolution used	**30 cm/pixel**	12.5 cm/pixel
Pixelwise F1 score	0.851	**0.884**
Vehiclewise F1 score	**0.800**	0.773
Mean prediction error counting cars	**1.67 %**	3.57 %
Evaluation time per tile*	**0.19 seconds**	28.19 seconds

Table 5.1: 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 fair comparisons. The evaluation time should therefore not be regarded as exact but as an indication of the speed difference between the two models.

Comparison on the Vedai dataset

Model	**Detection time per image***	**Vehiclewise 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	$\approx$ **0.19/4**	**0.554**

Table 5.2: Shows the comparison between the proposed model and the Faster R-CNN (Z&F), Faster R-CNN (VGG-16), Fast R-CNN (VGG-16) [55] and the Cascaded Convolutional Neural Networks (CCNN) [59] on the Vedai dataset. * The other models were evaluated 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 fair comparisons. The evaluation time should therefore not be regarded as exact but as an indication of the speed difference between the two models.

Cascaded Convolutional Neural Networks

Chapter 6

Discussion

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). URL: http://arxiv.org/abs/1609.06845 (visited on 03/09/2018).

[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] 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).

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

[11] 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).

[12] 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).

[13] 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).

[14] 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).

[15] 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).

[16] 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).

[17] 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).

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

[19] 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.

[20] 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).

[21] 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).

[22] 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).

[23] 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).

[24] 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).

[25] 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.

[26] 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.

[27] 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).

[28] 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).

[29] 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.

[30] 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).

[31] 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).

[32] 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).

[33] 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).

[34] 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).

[35] 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.

[36] 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.

[37] 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).

[38] 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).

[39] 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).

[40] 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).

[41] 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).

[42] 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).

[43] 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).

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

[45] 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).

[46] 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).

[47] 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.

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

[49] 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).

[50] 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).

[51] 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).

[52] 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).

[53] 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).

[54] 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).

[55] 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).

[56] 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.

[57] 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).

[58] 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).

[59] 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).

Appendix A

Unnecessary Appended Material