Aerial image analysis with generative adversarial networks

Joar Gruneau
joar@gruneau.se

April 10, 2018

Contents

1 Abstract

2 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 [6]. Conditional markov random fields (CRFs) have been a popular post processing step to ensure spatial contiguity in segmentation maps [55].

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 general 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, 49, 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, 25] vegetation monitoring [46, 8], urban area monitoring [26], 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 market-

place 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, 6, 30, 57, 5, 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 [6].

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.

3 Relevant Theory

3.1 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.

3.1.1 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.

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

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

$$\mathcal{L}(G, D) = -(\ell_{bce}(D(y), 1) + \ell_{bce}(D(G(x)), 0)) \tag{2}$$

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

$$G^* = argmin_G[max_D[\mathcal{L}(G, D)]] \tag{3}$$

3.1.2 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 (3).

$$\mathcal{L}(G, D) = -(\ell_{bce}(D(x, y), 1) + \ell_{bce}(D(x, G(x)), 0)) \tag{4}$$

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 multi class image segmentation a multi class cross entropy loss is a better option to enforce the generator to assign a high probability to the correct class. The multi class cross entropy loss is given below.

$$\ell_{mce}(\hat{y}, y) = -\sum_{n=1}^{H*W}\sum_{n=1}^{C} y * log(\hat{y}) \tag{5}$$

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.

$$G^* = argmin_G[max_D[\mathcal{L}(G, D)] + \lambda \ell_{mce}(G)] \tag{6}$$

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

3.2 Training generative adversarial networks

Generative adversarial networks are notoriously difficult to train. The min-max game formulated in 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.

3.2.1 Modify the loss function

3.2.2 Do not mix samples in a mini batch

Only use samples of one type i the mini batch. Use batch normalization as well

3.2.3 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.

3.2.4 Perform label smoothing

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.

3.3 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 conected 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, 48, 50]. Ronneberger *et al* [34] attributes this to the networks structure which creates internal data augmentation.

Note that the size of the input image and the output prediction in 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].

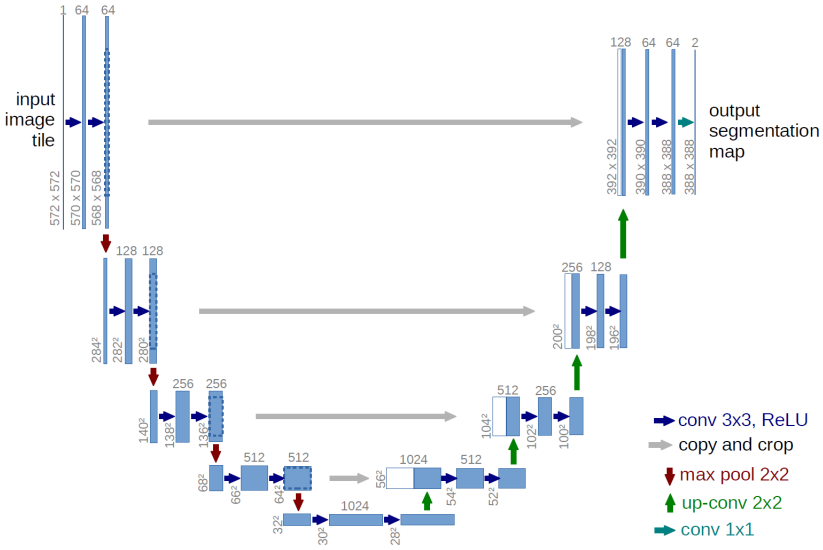

Figure 1: The initial u-net architecture proposed by Ronneberger *et al* [34]

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.

3.4 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 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.

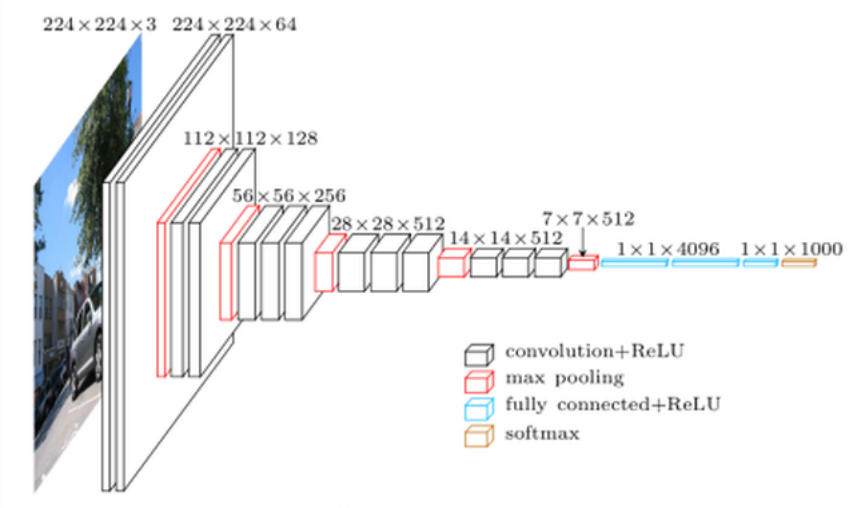

Figure 2: The VGG16 architecture proposed by [41]

ResNet is another widely popular classification network. It has performed very well on classification tasks [14, 45] It has also been shown that wider residual networks are more memory efficient while still obtaining comparable performance [47, 53]. The success behind the ResNet lies in the residual

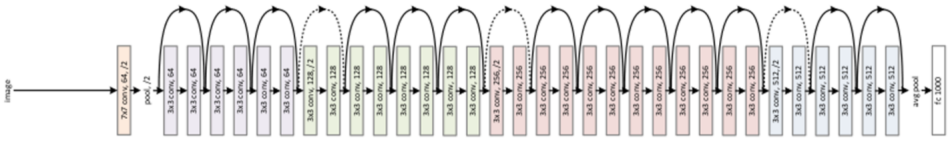

Figure 3: The 34 layer deep ResNet architecture [41]

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.

$$y = F(x, W) + x \tag{7}$$

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.

3.5 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 [44]. Below is the weighted cross entropy loss for a two class segmentation problem.

$$\mathcal{L}_{wce} = -\omega_c * y * log(\hat{y}) + (1 - y)log(1 - \hat{y}) \tag{8}$$

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.

$$w_c = \frac{N - \sum_n \hat{y}_{c=1}}{\sum_n \hat{y}_{c=1}} \tag{9}$$

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

$$w_c = \frac{\sum_n \sum_{W*H} y_{c=0}}{\sum_n \sum_{W*H} y_{c=1}} \tag{10}$$

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]

$$\mathcal{L}_{IoU} = 1 - \frac{\Sigma y \otimes \hat{y}}{\Sigma y + \Sigma \hat{y} - \Sigma y \otimes \hat{y}} \tag{11}$$

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.

$$\omega(x) = \omega_c(x) + \omega_0 * epx(\frac{(-d_1(x) - d_2(x))^2}{2\sigma^2}) \tag{12}$$

Here $\omega_c(x)$ depends on the class frequency and could be computes as 9 or 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.

3.6 The datasets

3.7 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 60006000 resolution. 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.

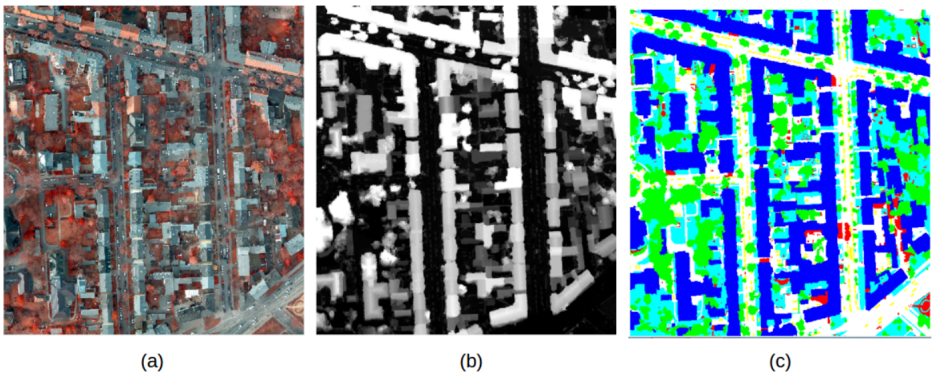

Figure 4: An example image from the Potsdam dataset with a: the rgb channel, b: the height data channel and c: the ground truth segmentation map.

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

$$Precision = \frac{True\ positive}{True\ Positive + False\ poitive} \tag{13}$$

$$Recall = \frac{True\ positive}{True\ Positive + False\ negative} \tag{14}$$

$$F1\ score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{15}$$

This work focuses on detecting vehicles in low cost images so only the rgb images will be used and they will be down sampled to a resolution of 1000 1000 pixels. Only the car class will also be predicted. One important thing 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.

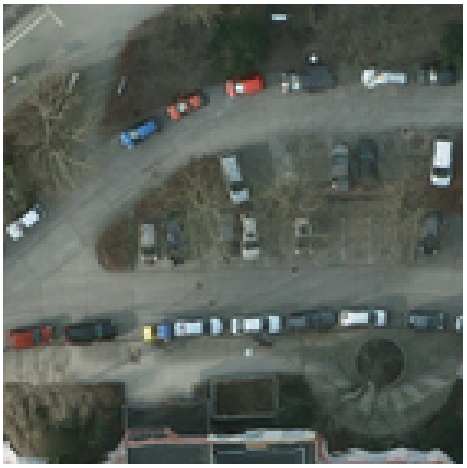

Image **(a)**, the input validation image

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

Figure 5: Figure shows the ground truth labels and the pixel weight map for a training image

The models will be evaluated on the [1]

4 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, 56]. Luc *et al* [23] proposed a adveserial 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 scal-

ing 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 segmentor trained only with the cross entropy loss by a significant margin.

5 Network Architecture and initial results

For the generator network a U-NET will be used and for the discriminator a ResNet with fourteen layers will be used. The U-NET will use only one image in the mini batch to minimize the computational cost. The input to the discriminative network will be the satellite image concatenated with the ground truth or generated segmentation. We will use a patchGan 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 patchGan will a smaller window was more stable during training than a patchGan with a larger window or an imageGan. 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 (1) and the multi class cross entropy loss (**??**) we can now define our loss function as.

$$\mathcal{L}(G, D) = \ell_{mce}(G(x), y) - \lambda(\ell_{bce}(D(x, y), 1) + \ell_{bce}(D(x, G(x)), 0)) \quad (16)$$

The generator will try to minimize this loss while the descriminator will try to maximize it. Following the example of [12, 23] and replace the term $-\lambda\ell_{bce}(D(G(x), y), 0)$ with $+\lambda\ell_{bce}(D(G(x), y), 1)$. Hence instead of minimizing the probability of the descriminative 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,

$$\mathcal{L}_{bce}(G, D) = \lambda(\ell_{bce}(D(x, G(x)), 1) - \ell_{bce}(D(x, y), 1)) \tag{17}$$

The objective for the network hence becomes.

$$G^* = argmin_G[max_D[\mathcal{L}_{bce}(G, D)] + \ell_{mce}(G(x), y)] \tag{18}$$

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 the discriminative network quickly learnt to spot the difference between the discrete ground truth labels and the coontinous 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 functions 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.

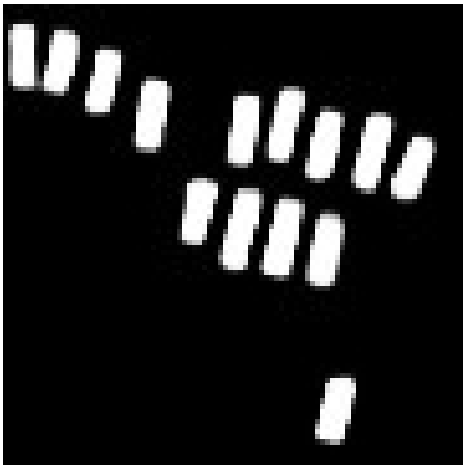

Image **(a)**, the input validation image

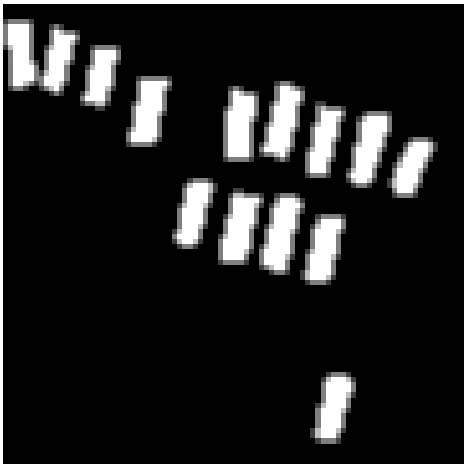

Image **(b)**, the ground truth segmentation

Figure 6: 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 combat this a an alternating training regime described below was instead 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 $cutoff = 1.0$. 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.

The GAN minimax 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 the original network. This changes greatly stabilized the training and made it easier to train the GAN. 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. Below is the f1 score on the validation data for both types of GANs.

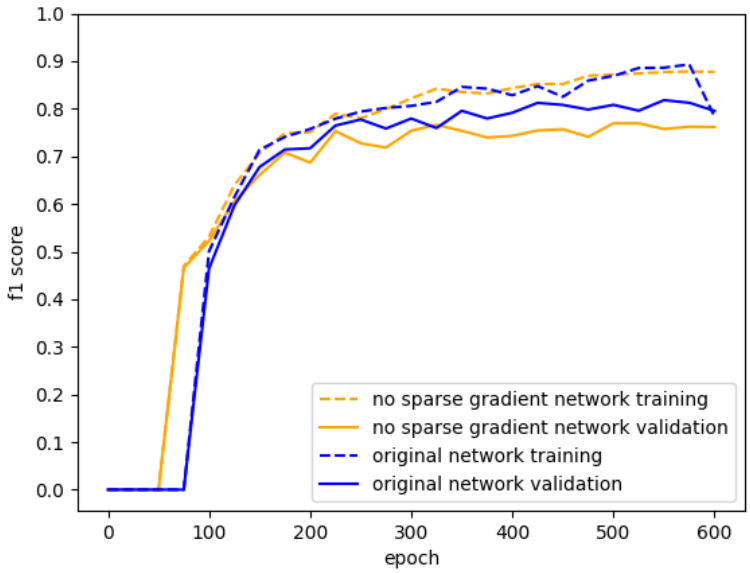

Figure 7: The VGG16 architecture proposed by [41]

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. In the below plots we displays the segmentation as a continuous values where the brightness of each pixel is the probability that that pixel belongs to a car 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. The segmented images looks very similar on a large scale.

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

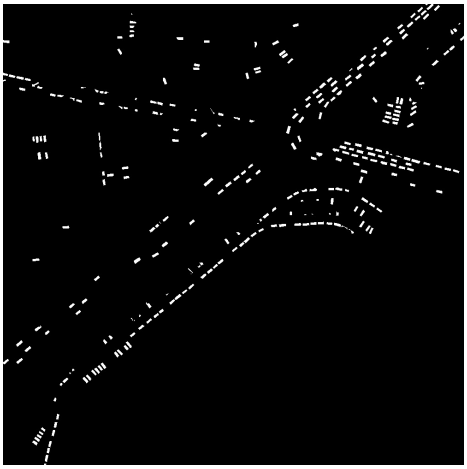

Image **(b)**, the ground truth segmentation

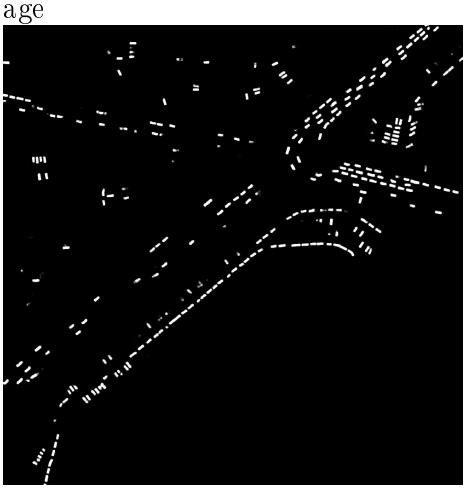

Image **(c)**, segmentation without an adversarial loss

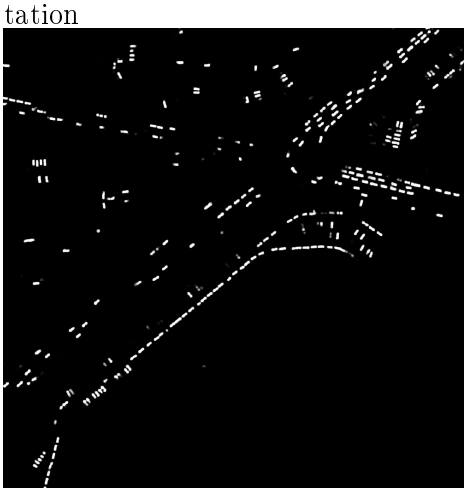

Image **(d)**, segmentation with an adversarial loss

Figure 8: Figure **??** shows the ground truth segmentation,

However if we look closely there are differences between the two networks. The adversarial network managed to enforce 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. Below are cherry picked images which displays the differences of the two networks. adversarial loss.

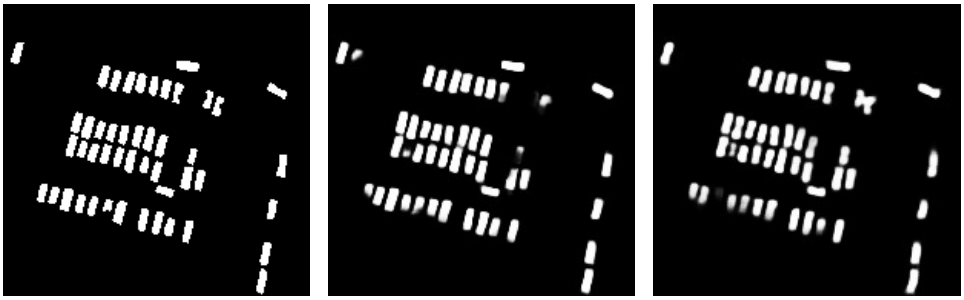

Figure 9: Figure **??** shows the ground truth segmentation,

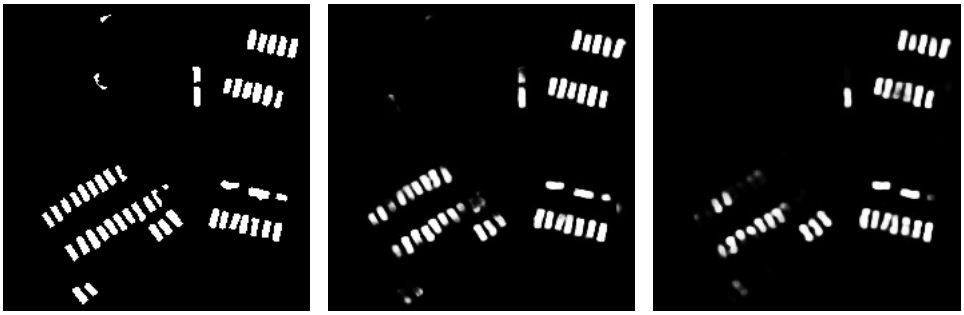

Figure 10: Figure **??** shows the ground truth segmentation,

When comparing the f1 scores the adversarial loss proved not to be helpful.

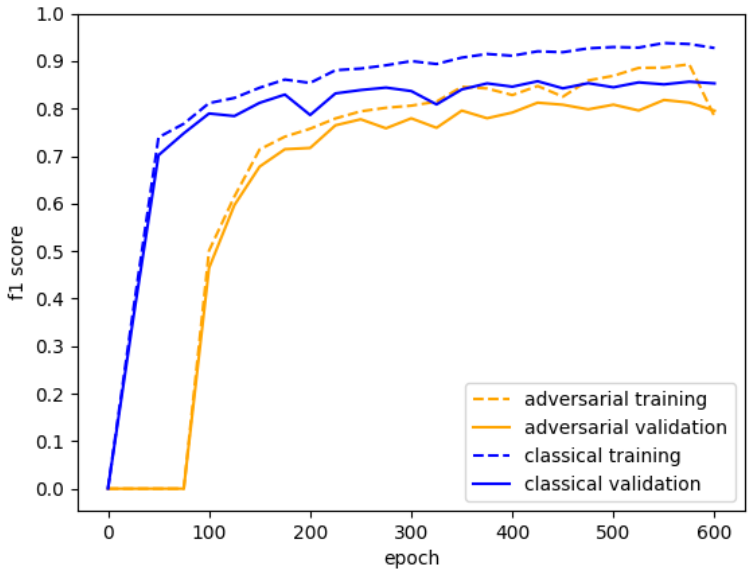

Figure 11: The VGG16 architecture proposed by [41]

Another approach to enforce object separation was to weight the loss for each pixel according to the proximity to the two nearest cars according to [??]. Initially $\omega_v ehicle$ was set to reflect the class imbalance. This led the model to achieve a good recall but a low precision and recall. The values $\omega_v ehicle = 2$ and $\omega_0 = 10$ was found to give a good f1 score while also enforcing better object separation. Below is the binary labels a well as the corresponding weight map for each pixel for the given values.

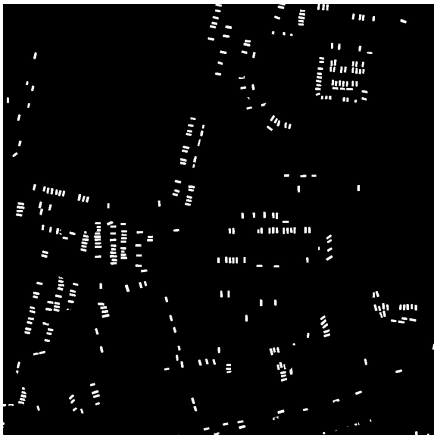

Image **(a)**, the input validation image

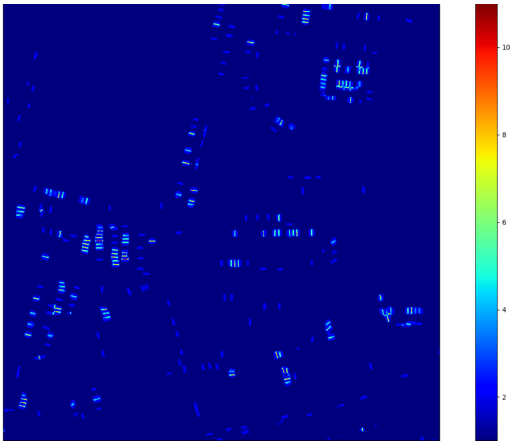

Image **(b)**, the ground truth segmentation

Figure 12: Figure shows the ground truth labels and the pixel weight map for a training image

The network with the weighted cross entropy loss achieved a comparable f1 score with the original network but but learned to separate objects much better. In the below segmentations we use discrete predictions to better display object separations.

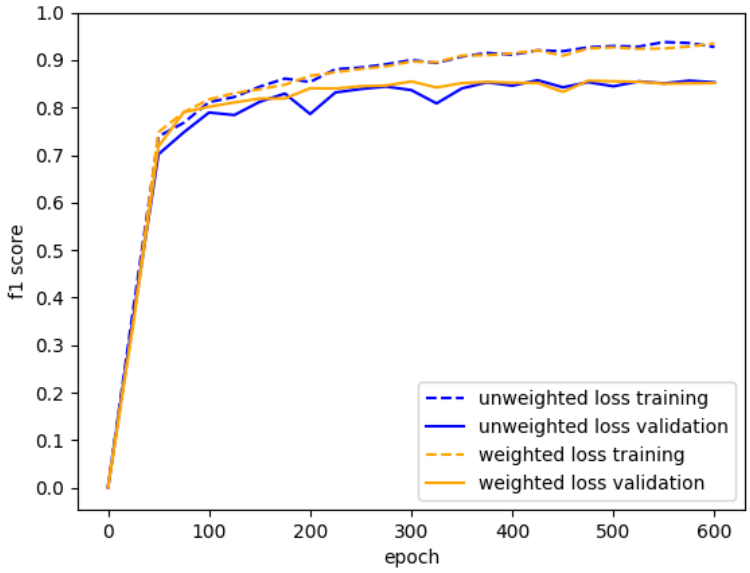

Figure 13

Figure 14

Figure 15

Since the U-NET with the pixel weighting scheme showed so promising initial result this was the choosen method for the evaluation of all of the final results.

6 Results

7 Discussion

8 Conclusion

References

[1] 2d Semantic Labeling - ISPRS. URL `http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html`.

[2] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan, and M. Zuair. Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sensing*, 9(4):312, Mar. 2017. doi: 10.3390/rs9040312. URL `http://www.mdpi.com/2072-4292/9/4/312`.

[3] A. Arbelle and T. R. Raviv. Microscopy Cell Segmentation via Adversarial Neural Networks. *arXiv:1709.05860 [cs]*, Sept. 2017. URL http://arxiv.org/abs/1709.05860. arXiv: 1709.05860.

[4] A. Arnab, S. Jayasumana, S. Zheng, and P. Torr. Higher Order Conditional Random Fields in Deep Neural Networks. *arXiv:1511.08119 [cs]*, Nov. 2015. URL http://arxiv.org/abs/1511.08119. arXiv: 1511.08119.

[5] N. Audebert, B. L. Saux, and S. Lefèvre. On the usability of deep networks for object-based image analysis. *arXiv:1609.06845 [cs]*, Sept. 2016. URL http://arxiv.org/abs/1609.06845. arXiv: 1609.06845.

[6] N. Audebert, B. Le Saux, and S. Lefèvre. Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images. *Remote Sensing*, 9(4):368, Apr. 2017. doi: 10.3390/rs9040368. URL http://www.mdpi.com/2072-4292/9/4/368.

[7] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *arXiv:1511.00561 [cs]*, Nov. 2015. URL http://arxiv.org/abs/1511.00561. arXiv: 1511.00561.

[8] J. A. J. Berni, P. J. Zarco-Tejada, L. Suarez, and E. Fereres. Thermal and Narrowband Multispectral Remote Sensing for Vegetation Monitoring From an Unmanned Aerial Vehicle. *IEEE Transactions on Geoscience and Remote Sensing*, 47(3):722–738, Mar. 2009. ISSN 0196-2892. doi: 10.1109/TGRS.2008.2010457.

[9] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv:1704.06857 [cs]*, Apr. 2017. URL http://arxiv.org/abs/1704.06857. arXiv: 1704.06857.

[10] R. Girshick. Fast R-CNN. *arXiv:1504.08083 [cs]*, Apr. 2015. URL http://arxiv.org/abs/1504.08083. arXiv: 1504.08083.

[11] I. Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv:1701.00160 [cs]*, Dec. 2016. URL http://arxiv.org/abs/1701.00160. arXiv: 1701.00160.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. URL http://arxiv.org/abs/1406.2661. arXiv: 1406.2661.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, Dec. 2015. URL http://arxiv.org/abs/1512.03385. arXiv: 1512.03385.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity Mappings in Deep Residual Networks. *arXiv:1603.05027 [cs]*, Mar. 2016. URL http://arxiv.org/abs/1603.05027. arXiv: 1603.05027.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *arXiv:1703.06870 [cs]*, Mar. 2017. URL http://arxiv.org/abs/1703.06870. arXiv: 1703.06870.

[16] A. C. Holt, E. Y. W. Seto, T. Rivard, and P. Gong. Object-based detection and classification of Vehicles from high-resolution aerial photography. *Photogrammetric Engineering and Remote Sensing*, 75(7):871–880, July 2009. ISSN 0099-1112. URL https://iths.pure.elsevier.com/en/publications/object-based-detection-and-classification-of-vehicles-from-high-r.

[17] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *arXiv:1611.07004 [cs]*, Nov. 2016. URL http://arxiv.org/abs/1611.07004. arXiv: 1611.07004.

[18] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[20] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv:1609.04802 [cs, stat]*, Sept. 2016. URL http://arxiv.org/abs/1609.04802. arXiv: 1609.04802.

[21] R. Li, W. Liu, L. Yang, S. Sun, W. Hu, F. Zhang, and W. Li. DeepUNet: A Deep Fully Convolutional Network for Pixel-level Sea-Land Segmentation. *arXiv:1709.00201 [cs]*, Sept. 2017. URL http://arxiv.org/abs/1709.00201. arXiv: 1709.00201.

[22] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312 [cs]*, May 2014. URL `http://arxiv.org/abs/1405.0312`. arXiv: 1405.0312.

[23] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic Segmentation using Adversarial Networks. *arXiv:1611.08408 [cs]*, Nov. 2016. URL `http://arxiv.org/abs/1611.08408`. arXiv: 1611.08408.

[24] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]*, Nov. 2014. URL `http://arxiv.org/abs/1411.1784`. arXiv: 1411.1784.

[25] T. Moranduzzo and F. Melgani. Automatic Car Counting Method for Unmanned Aerial Vehicle Images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(3):1635–1647, Mar. 2014. ISSN 0196-2892. doi: 10.1109/TGRS.2013.2253108.

[26] 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)*, pages 2362–2365, July 2015. doi: 10.1109/IGARSS.2015.7326283.

[27] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting. *arXiv:1604.07379 [cs]*, Apr. 2016. URL `http://arxiv.org/abs/1604.07379`. arXiv: 1604.07379.

[28] A. Polzounov, I. Terpugova, D. Skiparis, and A. Mihai. Right whale recognition using convolutional neural networks. *arXiv:1604.05605 [cs]*, Apr. 2016. URL `http://arxiv.org/abs/1604.05605`. arXiv: 1604.05605.

[29] M. A. Rahman and Y. Wang. Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, and T. Isenberg, editors, *Advances in Visual Computing*, pages 234–244, Cham, 2016. Springer International Publishing. ISBN 978-3-319-50835-1.

[30] S. Razakarivony and F. Jurie. Vehicle Detection in Aerial Imagery : A small target detection benchmark. *Journal of Visual Communication and Image Representation, Elsevier*, Mar. 2015. URL `https://hal.archives-ouvertes.fr/hal-01122605`.

[31] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative Adversarial Text to Image Synthesis. May 2016. URL `https://arxiv.org/abs/1605.05396`.

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv:1506.01497 [cs]*, June 2015. URL `http://arxiv.org/abs/1506.01497`. arXiv: 1506.01497.

[33] M. Rezaei, K. Harmuth, W. Gierke, T. Kellermeier, M. Fischer, H. Yang, and C. Meinel. Conditional Adversarial Network for Semantic Segmentation of Brain Tumor. *arXiv:1708.05227 [cs]*, Aug. 2017. URL `http://arxiv.org/abs/1708.05227`. arXiv: 1708.05227.

[34] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*, May 2015. URL `http://arxiv.org/abs/1505.04597`. arXiv: 1505.04597.

[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*, volume 1, pages 760–764 vol.1, Oct. 2003. 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)*, pages 916–923, Mar. 2017. doi: 10.1109/WACV.2017.107.

[37] A. G. Schwing and R. Urtasun. Fully Connected Deep Structured Networks. *arXiv:1503.02351 [cs]*, Mar. 2015. URL `http://arxiv.org/abs/1503.02351`. arXiv: 1503.02351.

[38] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv:1312.6229 [cs]*, Dec. 2013. URL `http://arxiv.org/abs/1312.6229`. arXiv: 1312.6229.

[39] E. Shelhamer, J. Long, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *arXiv:1605.06211 [cs]*, May 2016. URL `http://arxiv.org/abs/1605.06211`. arXiv: 1605.06211.

[40] K. Simonyan and A. Zisserman. Two-Stream Convolutional Networks for Action Recognition in Videos. *arXiv:1406.2199 [cs]*, June 2014. URL `http://arxiv.org/abs/1406.2199`. arXiv: 1406.2199.

[41] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*, Sept. 2014. URL `http://arxiv.org/abs/1409.1556`. arXiv: 1409.1556.

[42] J. Son, S. J. Park, and K.-H. Jung. Retinal Vessel Segmentation in Fundoscopic Images with Generative Adversarial Networks. *arXiv:1706.09318 [cs]*, June 2017. URL http://arxiv.org/abs/1706.09318. arXiv: 1706.09318.

[43] N. Souly, C. Spampinato, and M. Shah. Semi and Weakly Supervised Semantic Segmentation Using Generative Adversarial Network. *arXiv:1703.09695 [cs]*, Mar. 2017. URL http://arxiv.org/abs/1703.09695. arXiv: 1703.09695.

[44] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. *arXiv:1707.03237 [cs]*, 10553:240–248, 2017. doi: 10.1007/978-3-319-67558-9_28. URL http://arxiv.org/abs/1707.03237. arXiv: 1707.03237.

[45] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *arXiv:1602.07261 [cs]*, Feb. 2016. URL http://arxiv.org/abs/1602.07261. arXiv: 1602.07261.

[46] K. Uto, H. Seki, G. Saito, and Y. Kosugi. Characterization of Rice Paddies by a UAV-Mounted Miniature Hyperspectral Sensor System. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(2):851–860, Apr. 2013. ISSN 1939-1404. doi: 10.1109/JSTARS.2013.2250921.

[47] Z. Wu, C. Shen, and A. v. d. Hengel. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *arXiv:1611.10080 [cs]*, Nov. 2016. URL http://arxiv.org/abs/1611.10080. arXiv: 1611.10080.

[48] Y. Xue, T. Xu, H. Zhang, R. Long, and X. Huang. SegAN: Adversarial Network with Multi-scale L_1 Loss for Medical Image Segmentation. *arXiv:1706.01805 [cs]*, June 2017. URL http://arxiv.org/abs/1706.01805. arXiv: 1706.01805.

[49] Y. Xue, T. Xu, H. Zhang, R. Long, and X. Huang. SegAN: Adversarial Network with Multi-scale L_1 Loss for Medical Image Segmentation. *arXiv:1706.01805 [cs]*, June 2017. URL http://arxiv.org/abs/1706.01805. arXiv: 1706.01805.

[50] D. Yang, D. Xu, S. K. Zhou, B. Georgescu, M. Chen, S. Grbic, D. Metaxas, and D. Comaniciu. Automatic Liver Segmentation Using an Adversarial Image-to-Image Network. *arXiv:1707.08037 [cs]*, July 2017. URL http://arxiv.org/abs/1707.08037. arXiv: 1707.08037.

[51] D. Yang, D. Xu, S. K. Zhou, B. Georgescu, M. Chen, S. Grbic, D. Metaxas, and D. Comaniciu. Automatic Liver Segmentation Using an Adversarial Image-to-Image Network. *arXiv:1707.08037 [cs]*, July 2017. URL http://arxiv.org/abs/1707.08037. arXiv: 1707.08037.

[52] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. UnitBox: An Advanced Object Detection Network. *arXiv:1608.01471 [cs]*, pages 516–520, 2016. doi: 10.1145/2964284.2967274. URL http://arxiv.org/abs/1608.01471. arXiv: 1608.01471.

[53] S. Zagoruyko and N. Komodakis. Wide Residual Networks. *arXiv:1605.07146 [cs]*, May 2016. URL http://arxiv.org/abs/1605.07146. arXiv: 1605.07146.

[54] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. *arXiv:1311.2901 [cs]*, Nov. 2013. URL http://arxiv.org/abs/1311.2901. arXiv: 1311.2901.

[55] Y. Zhao, L. Zhang, P. Li, and B. Huang. Classification of High Spatial Resolution Imagery Using Improved Gaussian Markov Random-Field-Based Texture Features. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5):1458–1468, May 2007. ISSN 0196-2892. doi: 10.1109/TGRS.2007.892602.

[56] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional Random Fields as Recurrent Neural Networks. *arXiv:1502.03240 [cs]*, pages 1529–1537, Dec. 2015. doi: 10.1109/ICCV.2015.179. URL http://arxiv.org/abs/1502.03240. arXiv: 1502.03240.

[57] J. Zhong, T. Lei, and G. Yao. Robust Vehicle Detection in Aerial Images Based on Cascaded Convolutional Neural Networks. *Sensors (Basel, Switzerland)*, 17(12), Nov. 2017. ISSN 1424-8220. doi: 10.3390/s17122720. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5751529/.