Transferring a Segmentation Task Between Real and Synthetic Data

Eljo Dorrestijn[1] , **Robert-Jan Bruintjes**[1] , **Attila Lengyel**[1] , **Jan van Gemert1**[1]

[1]TU Delft

Abstract

In the field of ecology, camera traps are important tools to collect information on the wildlife of certain areas. The problem that arises with many camera traps is that they can collect more images than a human can realistically go trough all by themselves. To help classify these images computer vision is proposed as an alternative to manual classification. Many modern computer vision applications use neural networks. A hard part for the neural networks is that to train them well a large data set is needed, and sometimes it is almost impossible to build this dataset. This is where synthetic samples can be used instead of real samples. These samples are created by using computer graphics software to create realistic looking images to enlarge the dataset, or even be the whole dataset. This work evaluates how well a segmentation network was trained on only synthetic samples could perform on the real data. For this multiple segmentation networks were used like: U-Net [12] and SegNet [1] and the networks were trained on different datasets all derived from the synthetic data. The results show that while the networks can work real images that look similar to the synthetic samples, they fail to segment images that are captured in locations that look different from the synthetic samples.

1 Introduction

To do research on wildlife, ecologists often place camera traps to capture footage of wildlife in a certain region. They often face the problem where they have more footage than they can personally comb through. Within the footage there are a many different classes of animals.

In [3] researchers tried to improve the classification of rare classes of wildlife, deer specifically by creating synthetic data to increase the data set size. They did this by generating computer generated images of deer in a virtual environment. The network trained on the combined data set of both natural and synthetic data did have a significant improvement over the network trained on only the natural data set.

In some cases segmentation can be used to good effect to help the classification system reach a higher level of accuracy [7]. So to improve classification even further without the need of more natural data, it might be possible to use synthetic data to train a segmentation system to improve the classification network. But before this can be tested first the question needs to be asked how well a segmentation network will perform on real data if it is trained only on synthetic samples. This question will be central in this research paper. Important sub questions are how well do different types of networks perform comparatively and is it possible to get better performance on real data by using different types of data augmentation.

However segmentation of natural scenes comes with many challenges, because of the the dynamic nature of the scenes [10]. With complex background patterns segmenting a foreground object from the background can be quite difficult. This will be a challenge especially when using synthetic data, which does look realistic, but lacks some of the complexity that can be seen in natural scenes.

Figure 1: Images of a synthetic image with the corresponding image mask

2 Related Works

2.1 Segmentation for Camera Trap Images

There has been done some research into segmentation of camera trap images, but not much [6, 11, 16]. The main reason for this is that image segmentation is mostly used as a tool to help improve classification of images. In [6] they used a multi layered background subtraction method [15]. This is a good method to use when the background does not change, as it uses local texture features to segment the background from the foreground. This method would however probably fall short when used on different camera trap locations and needs ground truth images of the specific location which take much time to create. In [16] a fast segmentation method is proposed for camera trap images that is able to segment images with less computational complexity. For this method a requirement is that the images are collected in sequences, since they are compared to each other to find where the animals are. A benefit of this method is that only the bounding box is needed as a ground truth, which makes it easier to generate training sets, but this also means that the method returns a bounding box segmentation and not pixel based segmentation. In [11] they use a feature extraction followed by a series of image processing methods to segment a tiger from its background. This method makes use of classical computer vision techniques as opposed to machine learning. The benefit of this is that this method does not need a training set with ground truths, but the performance of the method is not as good as other machine learning methods.

A big hurdle for most of these segmentation methods is that they either need a large set of ground truth images or need images in a sequence. To overcome this hurdle it might be possible to substitute the training set with synthetic images. This can make it easier to collect large amount of training images with ground truth segmentation masks. However no research could be found where they used computer generated images to train a segmentation network for camera trap images.

2.2 Using Synthetic Data for Segmentation

There has been done research into using synthetic data to train segmentation networks. In [4, 13] synthetic data was used to generate urban environments to improve their semantic segmentation, while in [14] synthetic data was used for segmentation of different leaves of a plant. While other research using synthetic data to train a segmentation network is used for different types of scenes, the reasons for using synthetic data are mostly the same. One of the main reasons being that for training a segmentation network pixel based information is needed for each training image, which takes a lot of time. Right now supervised training methods are necessary to train a segmentation network, because of the pixel by pixel classification that happens during segmentation as opposed to classification networks that classify over a whole image. Using Synthetic data can resolve this issue by creating both the image and the pixel based ground truth. Another advantage of synthetic data is that you are not limited by the amount of real data that you can collect, since you can just generate as many images as needed to train your network.

2.3 Segmentation Networks

There are many types of Convolutional Neural Networks (CNN) that can efficiently segment images. One of the more popular networks are U-Net [12] and SegNet [1]. U-Net makes use of contracting and expanding layers, sometimes also called encoding and decoding layers. In the conctracting layers the network uses convolutions and max-pooling to downsample. To upsample again the layer is convoluted again, but also is concatenated with the corresponding layer from the contracting part. Segnet is a neural network that is similar to U-Net, but the main difference is that it does not concatenate with the corresponding layer from the contracting part.

To help the network perform better the encoding part of the segmentation network can be replaced by a pre-trained network. This can help the network with generalization. There exist many types of encoders that will work with a U-Net structure, in this research ResNet-50 [8] is used to see the difference in using a pretrained encoder to a non pre-trained encoder. The encoder is pre-trained on a large dataset of general images called imagenet [5].

3 Methodology

To determine if it is possible to transfer a segmentation task between real and synthetic data multiple segmentation networks and datasets were used. The three different types of segmentation networks used are a standard U-Net, a U-Net with a pre-trained ResNet-50 Encoder and a SegNet. These networks were chosen to determine two things, if the network structure and using a pre-trained encoder have a significant impact on the performance of the network.

Then these networks are trained on different datasets all stemming from the same set of synthetic images. These are the full sized images with masks as used in [3]. But out of this set only the day images were used to simplify the problem that has to be solved. Another dataset is created by cropping out 480 by 480 pixel crops around the deer, which results in more pixels of deer per image and the crops have the same size as the first and final layers of the segmentation networks, so that no resizing of the images has to be done before the images can be used for the network. The final dataset is created using the style transfer used in [9] on the cropped data set to create images that are more similar in style to the real camera trap images. As another form of data augmentation to test with images were changed during training, by changing brightness, contrast, saturation and hue.

To determine if these networks perform well they are evaluated against deer during the day from the Caltech Camera Trap dataset [2]. This subset contains 2003 images over 26 different camera trap locations. But for the ground truth only bounding boxes are available, even when there are multiple deer in the image. So to make a comparison between the predicted mask from the networks with the ground truths a bounding box is drawn around the largest generated segmentation mask. This is chosen over drawing a bounding box around all the segmentations, since it is not uncommon for a prediction to have a small segmentation far from the deer and if this happens the evaluation becomes a lot worse.

Then an evaluation is generated using IoU as a metric and using a qualitative analysis of the images. Once an evaluation is generated for all the trained networks they can be compared to each other to find out what settings of synthetic data perform the best on real camera trap data. Since the Locations of the camera traps can differ a lot in colour and lighting, evaluation is also done on individual camera trap locations. The qualitative analysis is done to specify in what specific cases our network falls short to get a better understanding of how it works.

4 Experimental Setup and Results

4.1 Datasets

All of the training datasets are derived from an original dataset containing synthetic images and corresponding masks. From this dataset 9252 images and masks were taken to be used. From the 9252 images and masks only the images with at deer on it were used to create the full sized image dataset of size 6988 images and masks. The cropped images dataset is created by cropping out 480x480 pixel bounding boxes around the masks only if the bounding box is at least 100 pixels in size. This resulted in a dataset of 13696 images and masks. This dataset was then style transferred to look more like the camera trap images, by using and image to image network, resulting in the i2i dataset of 13696 images and masks.

The test dataset is derived from the Caltech Camera Trap dataset [2]. From this dataset only images with deer are selected with times recorded between 7:00 and 19:59 to remove most of the night images, but some images had to be removed manually, since the recorded time cannot guarantee if the image is recorded using the night camera setting. The images are spread out in a long-tailed manner over 26 locations as can be seen in table 1, from over a thousand for one location to only one for another location. This means that when evaluating the performance of the networks on the real data, some locations can weigh much higher when taking the average over all locations. This is why evaluating per location can give more insightful information, since this will indicate where the networks perform well, and where they fall short.

Table 1: Amount of images per locations.

locations	2	4	6	10	11	21	23	25	26
images	1	42	54	36	8	11	17	3	2
locations	27	29	32	34	37	41	42	44	48
images	2	11	11	1017	29	262	35	28	21
locations	53	57	59	61	63	66	70	120	
images	31	138	53	69	79	26	12	5	

4.2 Segmentation Metrics

To do quantitative analysis on how well the network performs metrics are needed. Important metrics for segmentation are accuracy and intersection over union (IoU). Accuracy is calculated by using the following formula: TP + TN / (TP + TN + FP + FN) where TP is every true positive pixel, TN is every true negative pixel, FP is every false positive pixel and FN is every false negative pixel in the segmentation mask. IoU is calculated by dividing the intersection of the predicted segmentation mask and the ground truth mask by the union of the predicted segmentation mask and the ground truth mask. The problem with accuracy is that the accuracy can change because of the size of the bounding box in respect to the image. This is why in the following experiments only the IoU value is used.

4.3 Experiments

The following experiments were ran using Tensorflow 2 on python, with using keras to create the models for the networks and on a virtual machine running Ubuntu 18.04 with a NC6sv2 gpu and 112 GiB memory.

For the experiments three different network structures were used, a U-Net structure, a U-Net structure with a ResNet-50 Encoder and a SegNet structure. Each network is trained on 5 different settings and datasets, resulting in fifteen trained networks. Each network is trained on the previously mentioned datasets for 20 epoch and with a batch size of 4. Once trained all networks are evaluated against the real dear, by comparing the largest bounding box of the predicted segmentation mask. The reason for using 20 epoch is that every network has about a 0.80 IoU score on the validation set of synthetic deer at around 20 epoch. For the following results first the networks with the same architecture are compared to each other and and finally the best networks from each architecture are compared to another. In the following results the locations with less images than 30 will be left out, due to having a small sample size.

Table 2: Mean-IoU U-net

Location	**Full**	**Crop**	**i2i**	**Crop-Aug**	**i2i-Aug**
4	0	14	**15**	7	6
6	0	**15**	11	7	9
10	5	20	28	19	**33**
34	0	4	19	**38**	35
41	0	12	20	**24**	19
42	0	1	**3**	0	0
53	3	38	**68**	57	62
57	5	28	16	**32**	19
59	1	44	56	43	**65**
61	0	17	**26**	19	19
63	0	06	**23**	**23**	13
average	1	12	22	**33**	30

From the results of the U-Net architecture as can be seen in table 2 it can be concluded that the network trained on the full sized images performs significantly worse compared to the other datasets. For the other networks it becomes less clear which data set is best for all locations. On average the cropped and augmented images perform the best, but this network does get outperformed on different locations. The networks also never perform close to perfect in any location, since the highest it gets is a score of 0.68.

Table 3: Mean-IoU U-Net with ResNet-50 encoder

Location	Full	Crop	i2i	Crop-Aug	i2i-Aug
4	0	5	**8**	4	5
6	0	**12**	2	7	9
10	0	18	16	24	**32**
34	0	6	11	26	**33**
41	0	2	5	16	**18**
42	0	**5**	0	2	0
53	0	30	16	54	**62**
57	1	15	23	**31**	21
59	0	30	32	53	**54**
61	0	10	4	19	**23**
63	0	7	4	13	**27**
average	0	9	12	24	**28**

Unlike with the results of the U-Net the results of a U-Net with a pretrained ResNet-50 encoder does have a best performing model, the model using the style transferred dataset combined with data augmentation. What is interesting is that that the pretrained models seen in table 3 performs very similarly to the non pre trained models, only giving small improvements on some locations, but also performing a little bit worse on other locations.

Table 4: Mean-IoU SegNet

Location	Full	Crop	i2i	Crop-Aug	i2i-Aug
4	0	8	10	3	**9**
6	0	13	7	10	**16**
10	0	25	**27**	25	**27**
34	0	5	17	**34**	**34**
41	0	2	21	21	**22**
42	0	3	2	**5**	2
53	0	44	65	**68**	53
57	7	19	22	**29**	22
59	2	38	52	**69**	45
61	0	7	12	**29**	28
63	0	12	18	**20**	**20**
average	1	10	20	**31**	30

The evalutation of the SegNet as seen in table 4 is quite similar to the pretrained and the non pretrained U-Net. All networks perform well on the same locations while also performing worse on the other locations. The best performing networks are often the networks with data augmentation. But between the cropped and the style transferred dataset no best dataset can be chosen. This seems to point towards the idea that while style transfer can improve perfomance in certain senarios, it does not always help. What kind of network is used does not affect the performance by a lot. Different segmentation networks exist, and possibly a network that does have an improvement over the U-Net and SegNet networks, but were not able to be included in this research.

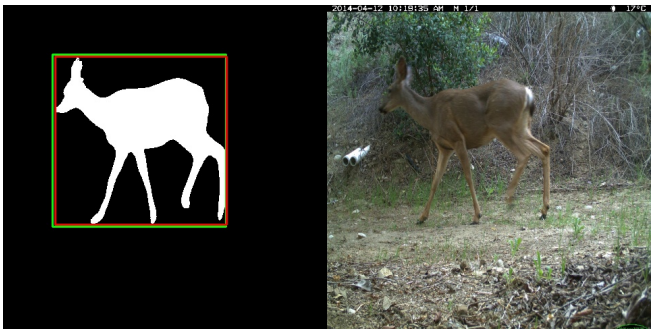

Figure 2: An almost perfect segmentation, in this image the bounding box of the deer overlaps almost perfectly with the ground truth bounding box

Figure 3: A segmentation where the deer is segmented out, but also another larger part of the image is segmented, causing the IoU score to be lower.

Figure 4: A seperated segmentation of a deer, which causes the IoU score to be lower because only the largest segmentation is compared to the ground truth.

Figure 5: An overexposed segmentation where the background is segmented as well and this part is connected to the segmentation of the deer.

4.4 Limitations Segmentation Metric

While using bounding boxes to calculate the IoU to evaluate the performance of the network is possibly one of the better options to get quantitative data on the performance, it still comes with many flaws. Some of the possible success and failure cases can be seen in figure 2 to 5, for example in figure 2 the bounding box around the segmentation mask almost perfectly matches the ground truth bounding box, in this case the metric works as intended. But in the other three images the performance is different when using bounding boxes as opposed to pixel based comparison. In figure 3 another lager part of the image is segmented, so the bounding box is drawn around this part, while the decently segmented deer is only partly covered. In figure 4 multiple segmentation patches can be seen covering the deer, but because they are not connected a smaller bounding box is drawn, which results in a lower score. Finally in figure 5 too much is segmented and quite a large part covers the bounding box of the deer, this case might actually result in a higher score than when a pixel based comparison would be used.

4.5 A Qualitative analysis

Because the quantitative metrics available at the time are far from perfect, a qualtitative analyisis was was chosen to support the quantitative analysis. In this section a deeper look is taken to see what images perform well and what images perform poorly and if possible come up with a hypothesis why there is a difference in performance between certain images.

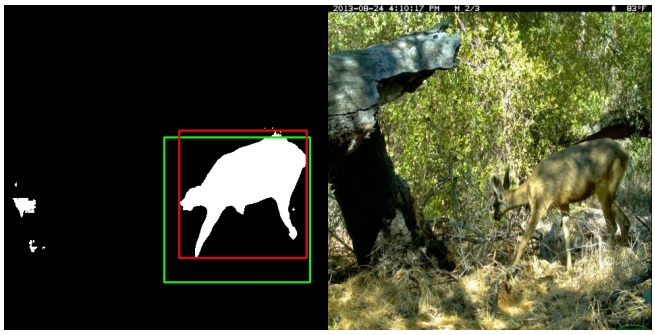

Figure 6: A deer in front of foliage segmented out reasonably well.

Figure 7: A deer behind foliage, making it hard for the network to segment out the deer compared to the deer in front of the foliage.

A hard case for a segmentation network is when the deer is in front of foliage as can be seen in figure 7. In this image the deer blends in very well with its environment making it hard for a network to detect. Whereas when a deer is in front of the foliage as seen in figure 6 the network is able to segment the deer.

Figure 8: An image with a blue sky, which does not appear in the training images.

A problem that our network experiences is that it is not trained very well on images with a blue sky. This is most likely because the training images only have a white sky. On top of this the cropped images have even less sky in them, since when the are cropped from the full images only the part with the deer and its close surrounding is kept of which very rarely sky is part of the surrounding.

5 Responsible Research

An important part of this paper is the reproducibility of it. Whenever another researcher decides to replicate the experiments of this paper, they should be able to get similar or the same results. To make this possible all the steps taken in the experiments are written down and can be followed without needing external information. But for neural networks specific information is needed to exactly replicate the same experiment, which would fill up a paper quite quickly, without adding much to the point that the paper is trying to make. This is why for this paper very basic neural networks are used, so that anyone that wants to replicate the neural networks can easily find how they are built by looking at the papers that are referenced. Of course it is still important to state in what environment and using what codebase the experiments were performed.

Another important topic is the correct use of data. In this paper data has been collected by evaluating the networks on real data. But this evaluation is not without its flaws, since it uses bounding boxes as a ground truth which makes the evaluation more of an indication of the performance. This has to be explained well in the paper so that readers know of the shortcomings of the results but also the reasons why this evaluation technique was chosen. When evaluating the networks are evaluated not only over the whole set of real images, but also by specific camera trap location. For some locations only a handful of images exist and for this reason the reported performance on these locations is not reliable due to the small sample size. This is why every camera trap location with less images than thirty is left out of the reported

evaluation in the paper. The data is saved in another location where it can be found outside of the paper, if an interested reader is curious about the results. But for most readers the point of this paper is better conveyed by showing only the data of locations with enough samples.

6 Discussion

In this paper different models were trained, with varying performance on the real data, but most models were able to correctly segment some real images. For each network there a large difference in score can be seen between certain locations. This can be the case because of several reasons, like different light conditions, a different type of background or the time of day. In our dataset all the synthetic samples were created using the same lighting conditions and type of environment, a lush mountainous forest at noon, as opposed to the more arid environments found in the camera trap images.

Between all of the trained networks no clear best network and dataset combination can be found. This leaves the question if there is another type of data augmentation to use which could improve the performance. Or possibly it could be the case that the synthetic samples are not varied enough in environment for the network to learn how to segment other environments. Using style transferred images did have a positive impact on some locations, but for this paper a style transfer was done using the style of all camera trap images. It might be possible to improve performance on certain locations by using a style transfer that is specifically trained on that location. Another possible improvement could be made by using style transfer for the real images, to make these look more similar to the synthetic images. This way the images look more like the synthetic images that the networks are trained on.

For future research aside from just training a segmentation on deer, a more general background segmentation network could be trained for all animals. This was not within the scope of this paper, as only samples of synthetic deer were available to us at the time, but by adding more animals to the training set, the network might be able to distinguish the differences between background and animal even better.

In this paper we were able to demonstrate that a segmentation task can be transferred between real and synthetic data. It can be quite helpful for ecologists to be able to train a model using synthetic data and still have it work when using real data. Although to make the segmentation flawless a lot more research has to be done into making the synthetic samples look more like the real images.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[2] Sara Beery. Caltech camera traps. https://beerys.github.io/CaltechCameraTraps, 2021-05-31.

[3] Sara Beery, Yang Liu, Dan Morris, Jim Piavis, Ashish Kapoor, Markus Meister, Neel Joshi, and Pietro Perona. Synthetic examples improve generalization for rare classes, 2019.

[4] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes, 2018.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Jhony-Heriberto Giraldo-Zuluaga, Augusto Salazar, Alexander Gomez, and Angélica Diaz-Pulido. Camera-trap images segmentation using multi-layer robust principal component analysis. *The Visual Computer*, 35(3):335–347, Dec 2017.

[7] Alexander Gomez, Augusto Salazar, and Francisco Vargas. Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks, 2016.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[9] Edoardo Lanzini. Image-to-image translation of synthetic samples for rare classes, 2021.

[10] Vijay Mahadevan and Nuno Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):171–177, 2010.

[11] K. Pavan Kumar Reddy and R. Aravind. Segmentation of camera-trap tiger images based on texture and color features. In *2012 National Conference on Communications (NCC)*, pages 1–5, 2012.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[13] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M. Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[14] Daniel Ward, Peyman Moghadam, and Nicolas Hudson. Deep leaf segmentation using synthetic data, 2019.

[15] Jian Yao and Jean-Marc Odobez. Multi-layer background subtraction based on color and texture. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[16] Hayder Yousif, Jianhe Yuan, Roland Kays, and Zhihai He. Fast human-animal detection from highly cluttered camera-trap images using joint background modeling and deep learning classification. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4, 2017.