Chapter 2

Related Works

2.1 Which Related Works Exists in Known Literature?

In the recent years Convolutional Neural Networks grew in popularity due to some great results they achieved in visual object classification and other pattern recognition tasks. Since CNNs use many hidden layers the generic term *"deep learning"* has arisen for CNNs in particular and other Neural Network with a lot of layers. One of the most cited scientist in correlation with deep learning is Yann LeCun. He has numerous publications in this area, which I cannot compress all in this chapter, but to mention some I came across a lot while browsing other papers to this topic, here are some of the most related Yann LeCun paper to my thesis: The [LKF10] shows how to develop a very basic CNN and describes how to do some optimizations that lower the number of required labeled training samples significantly by using an unsupervised learning algorithm. This paper was published in 2010, which is 2 years before the well-known AlexNet was published by Alex Krizhevsky and his colleagues, which made as a Neural Network the first big break through in image recognition. The paper [LBD+90] indicates, that already in 1990 successful attempts where made to perform some basic kind of object recognition tasks with multi layer Neural Networks, which were in many ways similar to todays deep learning networks. The network achieved an 1% error rate and a 9% rejection rate on handwritten digits, which was for that time a great accomplishment. Several years later in April 2012 Yann LeCun with Pierre Sermanet and Soumith Chintala returned to digits recognition. This time to photos of house numbers. They used a ConvNet and demonstrated different kind of pooling tactics to optimize the process.
In 2012 the AlexNet [KSH12] was published by Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton. It is considered to be the first CNN that outperformed conventional non Neural Network approaches in image recognition and revived the whole research in CNNs and Neural Networks in general. More on the AlexNet later in the background chapter [3.1.3].
"Going deeper with convolutions"[SLJ+14] is a interesting paper by researchers working at Google Inc. in which they describe the architecture of their GoogLeNet. The GoogLeNet is a deep Convolutional Neural network which decreased highly the computing resources of a Neural Network during training and testing. The architectural decisions are based on a theory called *"Hebbian principle"*, that states, that *"neurons that fire together, wire together"*, which effect is reinforced by the layout of the network by clustering neurons with highly correlated outputs and other comparable optimizations. This paper also inspired my introduction of this thesis and the example for the great improvement of CNNs with the Siberian husky and the Eskimo dog in the introduction are from this paper.

Speaking of *"going deeper with convolutions"*, the ResNet is probably the deepest CNN today with up to 152 layers, which are reasonable trainable in less than a week. [HZRS15] is a paper by a Microsoft Research group that features the architecture of the ResNet, which was the jumping-off point for this thesis approach. It has a deep but elegant design and can be programmed code length efficient with for example Torch7 in under 800 lines of code. Since I go into detail about this Neural Network in the Approach chapter [4.1.1], I will leave it for now at this point.
Up next I can recommend two papers by Matthew Zeiler and Rob Fergus which explored in reference to the publication of the AlexNet the intuition behind Convolutional Networks and show how to visualize the filters and weights correctly. Before that paper someone could get the feeling, that nobody really understands whats going on inside of these very successful networks, or better said it was difficult to explain the inner mechanics of a CNN, without talking about the inner state of different layers and how to illustrate them. The first of those two papers is the 2010 published paper [ZKTF10], which focuses on the different kind of features that different layers can learn to internalize and how they look like. The second paper is the 2013 published paper [ZF13], which developed some key ideas about improving the performance of CNNs, which were also later adopted by the GoogLeNet and the ResNet. In this paper the ZF Net is featured, which is a CNN that they later used to combine it with a so called DeConvNet, which is a deconvolutional or up-convolutional network to retrace the recognized features of different objects back to the input pixels of the whole network that stimulates those specific features. This paper laid the foundation for the second part of this thesis approach, where I try to determine the space an certain object occupies on an image.

2.1.1 Segmentation with the Help of Ground Truth Information

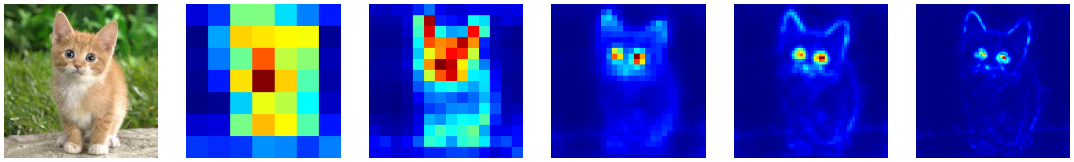
To improve the determination of the space an object occupies on an image drastically someone can train the deconvolutional part of the network with ground truth[1] data of the training images. The ZF Net and the VResNet don't train the deconvolution, they just extract information and can recover this way some heat map data, where certain objects lay in an image, but those heat maps are noisy and imprecisely. Since ground truth information aren't easy to acquire, those heat maps are realistically assessed the only option to choose, when developing a tracking network operating under real world circumstances. But whenever it is possible to access ground truth data I would recommend to consider some of the following networks in this subchapter, because those networks are able of a real and astonishingly precisely image segmentation.
In the paper [NHH15] they trained their deconvolutional network on top of a adopted VGG 16-layer Net. For that they use the PASCAL VOC 2012 dataset, which is a classification/detection competition dataset which offers training images with ground truth data. The paper attacks two big problems optimizing the image segmentation process. Firstly the problem of *"Inconsistent labels due to large object size"* and secondly the *"Missing labels due to small object size"* problem. At the end the paper compares the result of different segmentation techniques and shows, that their segmentation method reaches a 72.5% pixel-wise match with ground truth data, which is at this dataset a very good score.
[LSD14] is a paper about a comparable deconvolutional network with another network

[1] Ground truth data add a additional layer to an image, which sets an object class label to every pixel of the image. This data are in most cases implemented by hand and capture a perfect silhouette of an object.

architecture. It has a more nonlinear network layout and features skip layer connections like VResNet. It also features special convolutional layers as replacement for fully connected layers, which offer some advantages and is a way to make fully connected layers revertable for deconvolution. It also demonstrates untrained heat maps, that don't need ground truth information to train. But in comparison to my heat maps, they weren't processed by a deconvolutional network, which reassemble a better resolution of the heat map.
The next paper [BKC15] also uses a VGG 16-layer Net as the convolutional part of the network. A key part of the paper is road scene understanding and the attempt to realize end-to-end learning of deep segmentation architectures. The network is implemented with the Caffe framework [JSD+14] like the other two papers above in this subchapter. The key novel character of the featured SegNet in this paper is that it interchanged the conventional fully convolutional (deconvolutional) layers by an own new version, which combine the unpooling process with the upsampling process by the deconvolution in one step, which reduces the total amount of parameters of the network and has some other advantages as well.
The last paper in this category is the [ZC12] paper. The paper doesn't use a Neural Network, instead it uses the Conditional Random Field (CRF)[LMP01] which is a popular tool for object-based image segmentation and makes use of the Hidden Markov Model, which I don't further engaged in during my work on this thesis. Like the other 3 segmentation methods in this subchapter, the CRF uses ground truth information for training, too. The Hidden Markov Model, which uses some kind of Dynamic Bayesian Network, which is a probabilistic directed acyclic graphical model, change it state with a certain kind of probability which can be trained by supervised training. In the paper they say, that they *"develop an efficient inference algorithm that converges in a few seconds on a standard resolution image"*, which would be comparable if not even faster to most of the CNNs that perform image segmentation. But the image classification results are from my layman point of view in the context of CRF not quite as good as the results of the other segmentation Neural Networks.

2.1.2 Other Related Works with Different Methods or Objectives

The objective of this paper [DV16] is not a break through in some image classification competitions, it is to help understanding some core modules of CNNs, like strides, padding and pooling. It also exercises different combinations of values in strides, padding and pooling in detail and show which combination works best. Another source on this topic i can recommend is: [ODO16]. It also shows how to prevent checkerboard artefact's through some deconvolutional layers, which cost me a lot of time in developing a deconvolutional network and its just a problem of finding the right combination of values for strides, padding and pooling.
The paper [Zha16] comes close to this thesis approach by upsampling the heat map through unpooling back to the resolution of the input image. The paper also uses a VGG 16 Net model and perform some kind of deconvolution comparable to this thesis approach, but they use a winner takes it all principle in the upsampling process. This way it is guaranteed, that wrong activations of neural receptors don't appear in the final output heat map.



(a) tabby cat (b) pool5 (c) pool4 (d) pool3 (e) pool2 (f) pool1

Figure 2.1: [Zha16, Page 6, Fig. 3.]

For example you can see in figure 2.1f that most of the recognition values in the first heat map (figure 2.1b) were triggered by the eyes of the cat. But because of the winner-take-all principle a lot of the rest of the cat is lost. Because of that, I tried at my own approach another concept, which in the best case gives an idea of the scale of the object for distance measurements for object tracking purposes. However, this way a lot of false activations of neural receptors make it into the output images (see results 5...).
The "Recurrent Models of Visual Attention"[MHGK14] is one of the most ambitious research to make computer vision more human. It combines reinforcement learning with Neural Networks, which are two of the most promising areas of research in artificially intelligence, to emulate the human visual attention. The idea is simple. The Neural Network should learn by which images it should focus the field of attention at what areas to recognize the most features in big pictures by processing at least details as possible. The reason why the paper suggest to use of reinforcement learning, is because the learnable method for choosing the processing order for the field of attention is non-differentiable (figure 2.2), which means, no conventional backpropagation possible. But with reinforcement learning we can emulate this non-differentiable backpropagation using policy gradients.

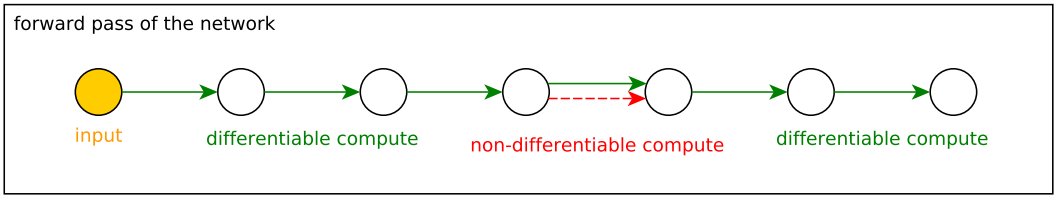


Figure 2.2: Network flow

We will train the parameters inside the green arrows with backpropagation as usual, but the parameters inside the red arrow will be updated independently of the backward pass using policy gradients, so that samples that hit low loss scores are reinforced (figure 2.3).

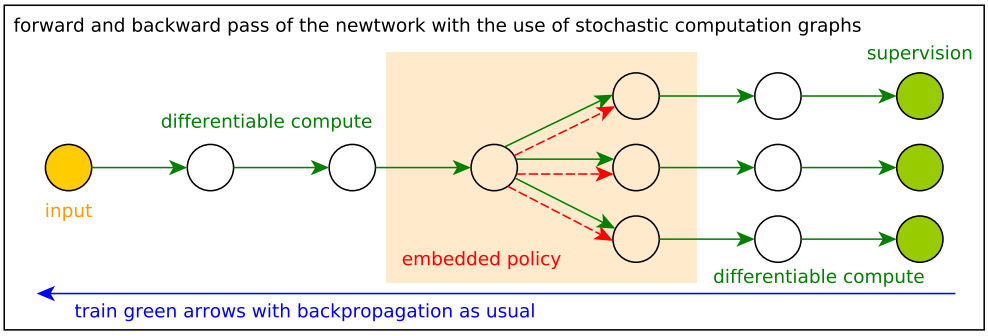
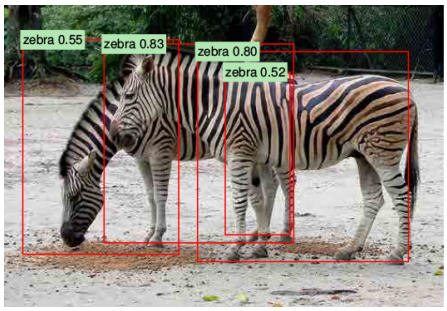
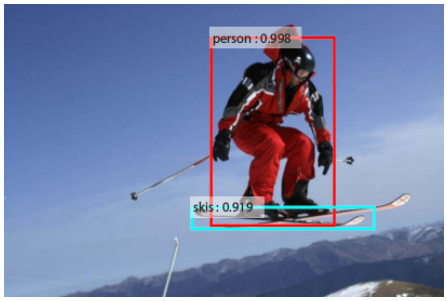


Figure 2.3: Network flow with PG

This method does not involve a heat map feature through some deconvolution, but it is possible to extract the glimpse path of the network and this could be an indication for the location of the object.
A more straight forward approach to find objects on big images are Region Based CNNs (R-CNNs). [GDDM13] is a paper about such an R-CNN. An R-CNN creates caption boxes for different kind of objects in images. They also can handle multiple objects which may or may not overlap, which may looks like this:



(a) [GDDM13, R-CNN]



(b) [RHGS15, Faster-RCNN]

(c) [chr17]

Figure 2.4: R-CNN caption boxes

However the older (2013) R-CNN version from the paper has some problems with creating multiple caption boxes for the same object. Which the newer ones can handle better.

This is an effect which arises from the R-CNN algorithm itself, which at first disassembles the images in many tiny parts and than scans the different parts for the objects. After that it determines which part is a segment of a bigger object and reassembles all the parts from each recognized object together. The process is however problematic and recognizes one object multiple times (figure 2.5): Therefore the R-CNN algorithm is performing after this step an object merging operation. With the effect, that newer R-CNN don't count one object multiple times most of the time. However R-CNNs need something like ground truth information during the training process, too.

Figure 2.5: [chr17]

The next paper that works on caption boxes is the [FLR+17] paper, which uses like this thesis approach a residual Neural Network and something called Single Shot Detector (SSD). It combines by the current state of scientific knowledge CNN with a caption boxes creating algorithm avoiding the problem of multiple object detections by using the SSD. The SSD is also a deep neural network, that *"completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network"* [LAE+15]. This approach needs ground truth data or predetermined optimal caption boxes to train the network. Another big difference to this thesis approach is, that their way of deconvolution is a deconvoluted version of the SSD they called Deconvolutional Single Shot Detector (DSSD). This thesis approach uses a deconvoluted version of an residual net, so that the entire CNN part is mirrored.

2.1.3 Further Related Works:

A very important optimization layer in this thesis approach is the *"bath normalization"* layer, which is preimplemented in Torch7 and is based on a paper [IS15] by Ioffe et al. paper .

The Batch Normalization basically tries to bring its input layer's values back to a zero-mean and unit variance condition, with an additional feature, that this optimization is optional weighted how hard it should kick in so that the weights of the Batch Normalization are trainable like the rest of the network. A more detailed explanation is provided in the background chapter [4.1.2].

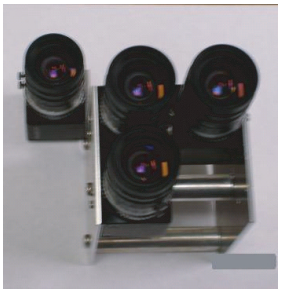


Figure 2.6: [Page 3, Fig.1. YPW+07]

Another more conventional technique without the usage of Neural Network is featured in this paper: [YPW+07]. Differential Camera Tracking however needs a series of photos taken from particularly positions and angles, a series of images of a moving object or a special camera like the prototype mentioned in this thesis in figure 2.6. But besides from that special requirement, it was before the upcoming utilization of CNNs one of the few sound methods for object tracking. A

big advantage in comparison to Neural Networks is that Differential Camera Tracking needs no labeled training images or any kind of ground truth information. Therefore one can think of a combination of Differential Camera Tracking and CNNs where the Differential Camera automatically cull moving objects in video streams and feed them labeled into a CNN that then learns the newly captured objects to recognize and locate them in future images without a movement.
At last but not least is a paper to be referred, that gives a glimpse of possible future use cases for deep learning: [LFDA16] The big question that this paper attacks is: *"does training the perception and control systems jointly end-to-end provide better performance than training each component separately?"*[abstract LFDA16]. End-to-end training from sensor system to actuator elements is a very promising concept. It would skip a lot of unnecessary intermediate calculations and would provide a more natural information processing of the robot. The robot could be trained via reinforcement learning in a real world setting, where the sensors learn something by the logic of the environment and not by some extra provided training samples. This is comparable to the idea of Deep Q Learning just in a real world setting and not in the well-known Atari games, where Deep Q Learning destroys human performances entirely. But due to all the hazard in real world applications, I think, it will take some time until robots will beat humans on a regular basis outside of sterile simulations.

2.2 Wherein Differs the Approach of the Thesis from State of the Art?

In the first part of this chapter I already compared some of the related works to this thesis approach, but to some up and complete the list of differences follows this subchapter:
On the first glance this thesis approach might not appear much different from the related works, but this thesis motivation was a new one and lead in the end to some interesting results and optimizations to consider about Convolutional Neural Networks. The idea was to create an autonomous learning computer program, which is able of real world object detection tasks, which only uses a Neural Network to do so, so no Markov random fields, Partial differential equation-based methods or other methods allowed. It also doesn't get any kind of groundtruth information about the object location and dimension during the training process, which means it has to elevate object classification data to an understanding about an object position on its own without further contextual information. The motivation behind this is, that a human brain does just relay on neurons to process information and has neither another numeracy skills nor groundtruth information to learn from. There are a few other approaches which meet this conditions, but they don't make use of an ResNet, which is one of the best CNNs today. This is because of a problem determining the input stimuli for a detected object due to some nonlinearities of the ResNet, which makes it impractical to retrace the input stimuli to the pixel input of an image.
Another alteration which most of the state of the art CNNs doesn't have is that this thesis network doesn't use a Fully Connected layer, which has the advantage that the input image size can be variable without further changes in the network itself. More on this later in chapter [4.1]. However this idea was previously implemented in [LSD14] as well.
A truly new idea in this thesis approach was to train the ResNet without the deconvolutional object tracking part and then convert the ResNet in a VGG Net like network with

the same trained weights in the neurons but without the nonlinear connections of the ResNet. Surprisingly, this conversion doesn't result in a big loss in the object classification abilities of the network. The loss is negligible and offers new ways of working with neural networks. In this case reverting the network for the deconvolutional part of the approach. Pretrained networks are not new, but to transform one network into another in a specific way with the possibility to use the transformed network for different tasks without loosing the learned information is an interesting concept to explore.
The last new idea to test was in the deconvolutional part of the network. I performed a noise function on the layers of the classification Volume after the last layer of the CNN part, which didn't own the highest object detection score. The layer with the highest score, which usually (chapter [5.3]) spots the object correctly, isn't touched by this noise function. But all other values in the volume will now be changed slightly every time for multiple deconvolution (object localization) processes, so that in the end I can see the pixels of the input image oscillating which don't lay in the object area of the image. I think this is a good way to detect the right pixels of an object class in a better acerbity, because just zeroing the values which doesn't supports the object detection is futile for volume sparse dimension locations which are low in all layers, but lay inside the object area. This may happen to locations of the image which are very dark.