Proposal for a Master Thesis
Text Extraction from Infographics Using Fully Convolutional Neural Networks

Morten Jessen

Arbeitsgruppe Knowledge Discovery
Institut für Informatik
Christian-Albrechts-Universität zu Kiel

1 Motivation

Infographics – like graphs – contain a lot of information, especially in scientific literature and the information is not necessarily repeated in the text. This makes the automatic extraction of text from infographics an important part of fully automating the analysis of scientific literature. The special difficulties associated with this task are the difficult distinction between graphic elements and text, different kinds of fonts, letter sizes, text/image resolution, orientation of text and the frequent use of abbreviations. Additionally, there are usually very few annotated training images available to train Machine Learning Algorithms.
The proposed master thesis will try to solve this problem using Fully Convolutional Neural Networks (FCNs). FCNs have surpassed other machine learning techniques in a lot – if not most – image processing tasks like semantic segmentation, character/facial recognition, etc. [7, 15–17, 19]. This suggests that FCNs could be very useful for the extraction of text from infographics as well.
The main problem for approaches using Neural Networks is usually the lack of sufficient training data. The same is true for this problem. The main dataset contains only ~ 240 images. Possible solutions will be evaluated and tested if they seem promising and necessary.

2 Proposed Methods

There are generally two different approaches to this problem. One is a multi-step approach, that first localizes text in general and then uses a second method to read text from the determined locations. The other tries to accomplish the same, but in a single step. This approach is similar to semantic segmentation where every word or letter is represented by a unique class.
FCNs seem to be powerful enough to accomplish this task in a single step which is the main approach of the proposed master thesis [7, 12–14]. At the same time it seems unfavourable to use word-classes, because of the frequent use of abbreviations in infographics. The more practical approach is to use symbol-classes which then have to be assembled to full words in a second step.

The following chapters give a short introduction to the benefits of using FCNs, the chosen ResNet architecture and explore the possible solutions to the lack of training data.

2.1 (Fully) Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are specialized neural networks for grid-like data like pictures – which are essentially 2- or 3-dimensional grids that are able to learn features, instead of using manually designed features. They use a variant of the mathematical specialized linear operation called convolution. This operation calculates the output for a certain activation function by summing up the weighted values in the surrounding area of the previous layer. This area is called the receptive field of that activation function, because the calculated value if affected by these inputs. This process allows the lower layers to grasp local information while higher layers can use the condensed output of these lower layers to capture global information. This also reduces the amount of parameters needed by a network – which in turn decreases the computational complexity. [8]
Like all other layers convolutional layers are used to produce a new output map from an input map. In fully connected layers this is done by having a learned weight for every input feature for every activation unit, which means that the weight matrix for each output feature has to be the same size as the input for each activation unit in every layer. For deep networks this is computationally very expensive especially for higher resolution output maps. In convolution operations this weight-matrix – called kernel – is much smaller and is reused for the calculation of every element of the output layer like illustrated in figure 1. This reduces the amount of parameters greatly while keeping the ability to use surrounding features to calculate an output feature. The kernel's values are usually learned during the training process.

The reduced size of the kernel allows using a lot of different kernels in the same layer to create multiple feature maps while keeping the computational complexity manageable. In fully connected solutions with big kernels the use of many maps is infeasible, due to the amount of parameters introduced by those layers. [2] The conceptual idea of using multiple feature maps is that different kernels find different features. While lower layer kernels may find horizontal, vertical, or diagonal edges higher layer ones might learn to combine those edges to general object shapes or find class-specific features. [12]
While fully connected layers need kernels with the same size as their input, convolutional layers are not limited in that respect. This does not imply that the same convolutional layer works well for inputs of any size.
The last layer of CNNs is usually a fully connected layer that uses all the information contained in the previous output map to ascertain the class affiliation of each pixel. In fully convolutional networks this last layer is replaced by a convolutional layer as well. This allows the network to use input images of any size[1] and avoid the implied loss of spatial information due to the connection to

[1] The input size for a specific model is still going to be fixed, but the same network could be trained with different sized input images without changing the architecture.

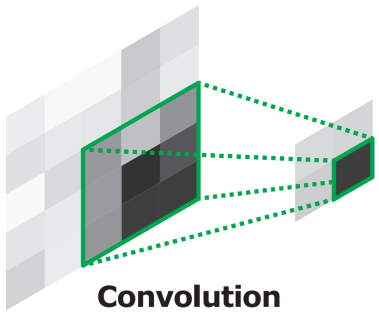
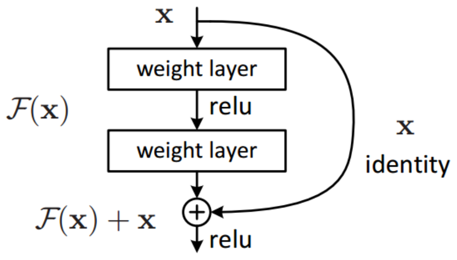


Fig. 1. Illustration of a 3×3 convolutional kernel processing an input with a step-witdh – called stride – of 2. The kernel will first be applied to the top left 3×3 window of input features to create the respective output feature. Then it will move to the top right – moving two spaces – and create the top right output and continue moving in steps of two until the whole input has been covered. This results in the shown 2×2 output, where all 4 features are created using the same 3×3 kernel, but different input values. Reprinted from [16].

all input values at once. Convolution, with its fixed receptive field, is better at using the relevant spatial data in a restricted area to make better segmentation decisions. Lastly, like previously explained, the convolutional layer is much less computationally complex than the fully connected layer.
The disadvantage of this procedure is that every feature on the output layer has a limited receptive field – meaning that the features of previous layers which affect its activation are limited to a certain area. The difficulty is to find a balanced approach that increases the size of the receptive field on the one hand – providing global context – while not loosing local information on the other hand.

2.2 Residual Network [10]

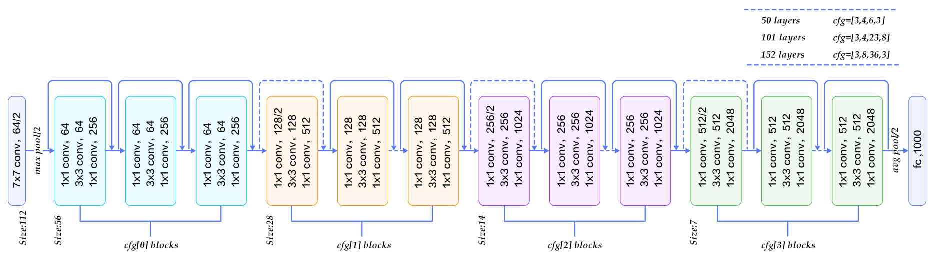
The problem adressed by the ResNet [10] is the degredation problem. Based on the observation that a layer can assume the identity function – which would make it mathematically negligible – it would be reasonable to assume that adding additional layers to a network could either reduce the training error or not change it at all. But actual tests show that this is not the case. Instead with the increase of the network's depth, the accuracy of the network first gets saturated and then – unexpectedly – drops rapidly. To avoid this behaviour additional paths through the model are added which allow easy approximation of the identify function if necessary. The basic building block is visualized in Fig. 2.
Instead of approximating a single function $H(x)$, the normal path through the layers can appoximate $H(x) - x$. An additional circumventing path is added which is implemented as an identity function, which results in a final output of the unit of $F(x) = (H(x) - x) + x$. With this changed architecture it is very easy for the algorithm to approximate the identity function by setting the weights inside the traditional path to zero. Tests by He, et al. have shown that this



x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 2. Building block for the ResNet. Reprinted from [10].

method works and that the ResNet can gain accuracy by increasing the depth of the network up to ~ 1000 layers.
The ResNet architecture is shown in Fig 3. The initial input data is rescaled by a 7x7 convolutional layer and a stride 2 max-pooling to a size of 112px. The following path can generally be divided into four major parts, labeled as cfg[0-3] blocks in the image. From one block type to the next the amount of kernels is doubled (256, 512, 1024, 2048) and the input size is halved (56, 28, 14, 7). The additional circumventig paths can be seen in the graphic as well.



50 layers cfg=[3,4,6,3]
101 layers cfg=[3,4,23,8]
152 layers cfg=[3,8,36,3]
7x7 conv, 64/2
max pool/2
1x1 conv, 64 3x3 conv, 64 1x1 conv, 256
1x1 conv, 128/2 3x3 conv, 128 1x1 conv, 512
1x1 conv, 128 3x3 conv, 128 1x1 conv, 512
1x1 conv, 256/2 3x3 conv, 256 1x1 conv, 1024
1x1 conv, 256 3x3 conv, 256 1x1 conv, 1024
1x1 conv, 512/2 3x3 conv, 512 1x1 conv, 2048
1x1 conv, 512 3x3 conv, 512 1x1 conv, 2048
avg pool/2
fc, 1000
Size:112
Size:56
Size:28
Size:14
Size:7
cfg[0] blocks
cfg[1] blocks
cfg[2] blocks
cfg[3] blocks

Fig. 3. Architecture of ResNet. In this project ResNet-101 has been used. Reprinted from [3].

2.3 Additional Training Data

Lack of training data is one of the main problem of deep learning approaches. In this case the data set only contains ~ 240 images with annotations for the bounding boxes for text in general. It is doubtful that this is enough training data for a FCN and it is likely that solutions have to be found for this problem, especially if the approach using symbol detection is used other training data has to be utilized.

There are multiple possible solutions to this problem. One is to combine data from other datasets (and/or problems) to provide more (or any) training data. There are big datasets – for example COCO-Text [1] or Total Text Dataset [5] – for text extraction from natural images. Although natural images are very different from the infographics as described above, it is very possible that the neural network could (at least) be pre-trained on those datasets. It could also be viable – but less probable – to train the network only on natural images and achieve acceptable results on infographics.

Beside natural images containing text there are big datasets of symbols to train neural networks for character recognition. In a different approach the network could be trained to recognize symbols using those datasets. To achieve acceptable results the relative size of the symbols has to be variable which could be achieved by easy data augmentation techniques like mirroring, rotation and scaling.

Those augmentation techniques could be applied to the initial dataset provided for this master thesis to improve the results. Though it is questionable if they are enough to increase the training set size to sufficient levels and this does not solve the problem of missing symbol annotations.

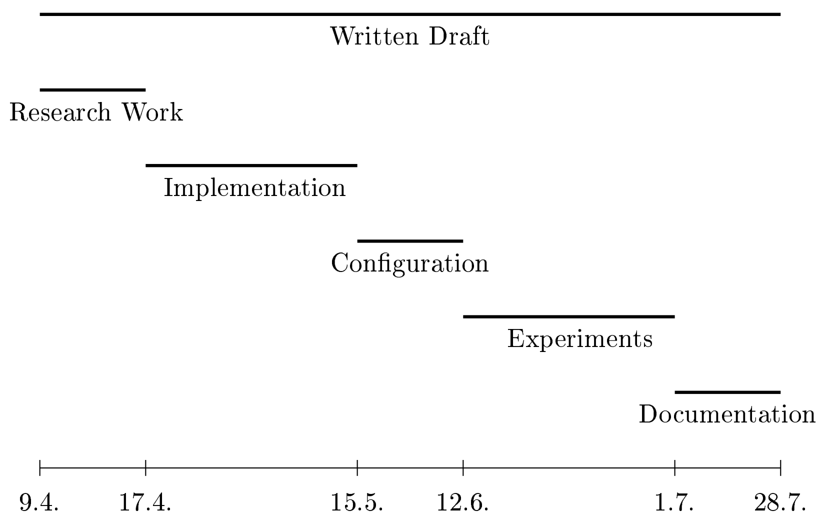
The final and most complex solution to the lack of training data is the generation of synthetic training data. There are synthetic datasets like The Synthethic Word Dataset [4] which contains 9 Million images covering 90000 English words. This dataset could be used in a similar fashion as the natural datasets mentioned previously. The most promising – but also most difficult – approach is to generate individual training data using a Generative Adversarial Network (GAN) [9]. There are several recently published papers [6, 18] using GANs to augment datasets. GANs consist of two neural networks that contest each other in a zero-sum game. While one network tries to generate images that are indistinguishable from real data the second network distinguishes fake from real images. While this approach might work very well for this problem, because the training data has a relatively simple structure, it may still be difficult to implement and train and may go beyond the scope of this master thesis.

3 Objective

The objective of the proposed master thesis consists of the following items:

Mandatory:	Implement different variants of the chosen Neural Network(s) and evaluate their performance for the two different approaches (Semantic Segmentation, Localization + Optical Character Recognition).
Optional 1:	Use natural image datasets to (pre-)train the networks.
Optional 2:	Use (simple) data augmentation methods to increase the size of the training set size.
Optional 3:	Use a Generative Adversarial Network to create synthetic training data. [6, 18]

4 Time Schedule



5 Prerequisites

5.1 Server

- Python 2/3.x
- Tensorflow 1.7
- Keras
- (possibly) Caffe2
- CUDA 8.x/9.1
- Sufficient Hardware (GPUs) and "timeslots" to run the experiments.

6 Preliminary Structure

1. Introduction
2. Problem description
3. Theoretical background
   - 3.1. Fully Convolutional Networks
   - 3.2. ResNet
   - 3.3. Mask R-CNN [11]
   - 3.4. Generative Adversarial Networks
4. Data Sets
5. Evaluation Methods
6. Experiments
   - 6.1. Configuration Results for Network Parameters
   - 6.2. Comparison of Different Network Sizes
   - 6.3. Comparison of Localization vs. Classification Approach
7. Final Result Evaluation and Comparison
8. Outlook

References

1. COCO-Text: Dataset for Text Detection and Recognition. `https://vision.cornell.edu/se3/coco-text-2/`. Accessed: 2018-03-29.
2. CS231n Convolutional Neural Networks for Visual Recognition. `http://cs231n.github.io/convolutional-networks/`. Accessed: 2018-04-27.
3. Image Classification. `http://book.paddlepaddle.org/03.image_classification/`. Accessed: 2017-12-01.
4. Synthetic Word Dataset. `http://www.robots.ox.ac.uk/~vgg/data/text/`. Accessed: 2018-03-29.
5. Total Text Dataset. `https://github.com/cs-chan/Total-Text-Dataset`. Accessed: 2018-03-29.
6. Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.
7. Liang-Chieh Chen et al. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915v2*, 2017.
8. Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. MIT press Cambridge, 2016.
9. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
10. Kaiming He et al. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
11. Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.
12. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
13. Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.
14. Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–104. IEEE, 2004.
15. Jonathan Long et al. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
16. Hyeonwoo Noh et al. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015.
17. Olaf Ronneberger et al. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.
18. Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017.

19. Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

April 27, 2018

Morten Jessen

Falk Böschen

Prof. Dr. Ansgar Scherp