The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

SEE: Towards Semi-Supervised End-to-End Scene Text Recognition

Christian Bartz, Haojin Yang, Christoph Meinel
Hasso Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert Straße 2-3
14482 Potsdam, Germany
{christian.bartz, haojin.yang, meinel}@hpi.de

Abstract

Detecting and recognizing text in natural scene images is a challenging, yet not completely solved task. In recent years several new systems that try to solve at least one of the two sub-tasks (text detection and text recognition) have been proposed. In this paper we present SEE, a step towards semi-supervised neural networks for scene text detection and recognition, that can be optimized end-to-end. Most existing works consist of multiple deep neural networks and several pre-processing steps. In contrast to this, we propose to use a single deep neural network, that learns to detect and recognize text from natural images, in a semi-supervised way. SEE is a network that integrates and jointly learns a spatial transformer network, which can learn to detect text regions in an image, and a text recognition network that takes the identified text regions and recognizes their textual content. We introduce the idea behind our novel approach and show its feasibility, by performing a range of experiments on standard benchmark datasets, where we achieve competitive results.

Introduction

Text is ubiquitous in our daily lives. Text can be found on documents, road signs, billboards, and other objects like cars or telephones. Automatically detecting and reading text from natural scene images is an important part of systems, that are to be used for several challenging tasks, such as image-based machine translation, autonomous cars or image/video indexing. In recent years the task of detecting text and recognizing text in natural scenes has seen much interest from the computer vision and document analysis community. Furthermore, recent breakthroughs (He et al. 2016a; Jaderberg et al. 2015b; Redmon et al. 2016; Ren et al. 2015) in other areas of computer vision enabled the creation of even better scene text detection and recognition systems than before (Gómez and Karatzas 2017; Gupta, Vedaldi, and Zisserman 2016; Shi et al. 2016). Although the problem of Optical Character Recognition (OCR) can be seen as solved for text in printed documents, it is still challenging to detect and recognize text in natural scene images. Images containing natural scenes exhibit large variations of illumination, perspective distortions, image qualities, text fonts, diverse backgrounds, etc.

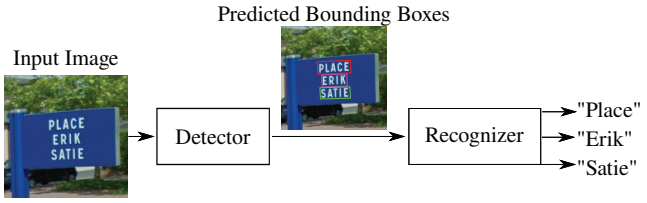

Figure 1: Schematic overview of our proposed system. The input image is fed to a single neural network that consists of a text detection part and a text recognition part. The text detection part learns to detect text in a semi-supervised way, by being jointly trained with the recognition part.

The majority of existing research works developed end-to-end scene text recognition systems that consist of complex two-step pipelines, where the first step is to detect regions of text in an image and the second step is to recognize the textual content of that identified region. Most of the existing works only concentrate on one of these two steps.

In this paper, we present a solution that consists of a single Deep Neural Network (DNN) that can learn to detect and recognize text in a semi-supervised way. In this setting the network only receives the image and the textual labels as input. We do not supply any groundtruth bounding boxes. The text detection is learned by the network itself. This is contrary to existing works, where text detection and text recognition systems are trained separately in a fully-supervised way. Recent work (Dai, He, and Sun 2016) showed that Convolutional Neural Networks (CNNs) are capable of learning how to solve complex multi-task problems, while being trained in an end-to-end manner. Our motivation is to use these capabilities of CNNs and create an end-to-end trainable scene text recognition system, that can be trained on weakly labelled data. In order to create such a system, we learn a single DNN that is able to find single characters, words or even lines of text in the input image and recognize their content. This is achieved by jointly learning a localization network that uses a recurrent spatial transformer (Jaderberg et al. 2015b; Sønderby et al. 2015) as attention mechanism and a text recognition network. Figure 1 provides a schematic overview of our proposed system.

Our contributions are as follows: (1) We present a novel

end-to-end trainable system for scene text detection and recognition by integrating spatial transformer networks. (2) We propose methods that can improve and ease the work with spatial transformer networks. (3) We train our proposed system end-to-end, in a semi-supervised way. (4) We demonstrate that our approach is able to reach competitive performance on standard benchmark datasets. (5) We provide our code[1] and trained models[2] to the research community.

This paper is structured in the following way: We first outline work of other researchers that is related to ours. Second, we describe our proposed system in detail. We then show and discuss our results on standard benchmark datasets and finally conclude our findings.

Related Work

Over the course of years a rich environment of different approaches to scene text detection and recognition have been developed and published. Nearly all systems use a two-step process for performing end-to-end recognition of scene text. The first step, is to detect regions of text and extract these regions from the input image. The second step, is to recognize the textual content and return the text strings of the extracted text regions.

It is further possible to divide these approaches into three broad categories: (1) Systems relying on hand crafted features and human knowledge for text detection and text recognition. (2) Systems using deep learning approaches, together with hand crafted features, or two different deep networks for each of the two steps. (3) Systems that do not consist of a two step approach but rather perform text detection and recognition using a single deep neural network. For each category, we will discuss some of these systems.

Hand Crafted Features In the beginning, methods based on hand crafted features and human knowledge have been used to perform text detection and recognition. These systems used features like MSERs (Neumann and Matas 2010), Stroke Width Transforms (Epshtein, Ofek, and Wexler 2010) or HOG-Features (Wang, Babenko, and Belongie 2011) to identify regions of text and provide them to the text recognition stage of the system. In the text recognition stage sliding window classifiers (Mishra, Alahari, and Jawahar 2012) and ensembles of SVMs (Yao et al. 2014) or k-Nearest Neighbor classifiers using HOG features (Wang and Belongie 2010) were used. All of these approaches use hand crafted features that have a large variety of hyper parameters that need expert knowledge to correctly tune them for achieving the best results.

Deep Learning Approaches More recent systems exchange approaches based on hand crafted features in one or both steps of recognition systems by approaches using DNNs. Gómez and Karatzas (Gómez and Karatzas 2017) propose a text-specific selective search algorithm that, together with a DNN, can be used to detect (distorted) text regions in natural scene images. Gupta et al. (Gupta, Vedaldi, and Zisserman 2016) propose a text detection model based on the YOLO-Architecture (Redmon et al. 2016) that uses a fully convolutional deep neural network to identify text regions.

Bissacco et al. (Bissacco et al. 2013) propose a complete end-to-end architecture that performs text detection using hand crafted features. Jaderberg et al. (Jaderberg et al. 2015a; Jaderberg, Vedaldi, and Zisserman 2014) propose several systems that use deep neural networks for text detection and text recognition. In (Jaderberg et al. 2015a) Jaderberg et al. propose to use a region proposal network with an extra bounding box regression CNN for text detection. A CNN that takes the whole text region as input is used for text recognition. The output of this CNN is constrained to a pre-defined dictionary of words, making this approach only applicable to one given language.

Goodfellow et al. (Goodfellow et al. 2014) propose a text recognition system for house numbers, that has been refined by Jaderberg et al. (Jaderberg, Vedaldi, and Zisserman 2014) for unconstrained text recognition. This system uses a single CNN, taking the whole extracted text region as input, and recognizing the text using one independent classifier for each possible character in the given word. Based on this idea He et al. (He et al. 2016b) and Shi et al. (Shi, Bai, and Yao 2016) propose text recognition systems that treat the recognition of characters from the extracted text region as a sequence recognition problem. Shi et al. (Shi et al. 2016) later improved their approach by firstly adding an extra step that utilizes the rectification capabilities of Spatial Transformer Networks (Jaderberg et al. 2015b) for rectifying extracted text lines. Secondly they added a soft-attention mechanism to their network that helps to produce the sequence of characters in the input image. In their work Shi et al. make use of Spatial Transformers as an extra pre-processing step to make it easier for the recognition network to recognize the text in the image. In our system we use the Spatial Transformer as a core building block for detecting text in a semi-supervised way.

End-to-End trainable Approaches The presented systems always use a two-step approach for detecting and recognizing text from scene text images. Although recent approaches make use of deep neural networks they are still using a huge amount of hand crafted knowledge in either of the steps or at the point where the results of both steps are fused together. Smith et al. (Smith et al. 2016) and Wojna et al. (Wojna et al. 2017) propose an end-to-end trainable system that is able to recognize text on French street name signs, using a single DNN. In contrast to our system it is not possible for the system to provide the location of the text in the image, only the textual content can be extracted. Recently Li et al. (Li, Wang, and Shen 2017) proposed an end-to-end system consisting of a single, complex DNN that is trained end-to-end and can perform text detection and text recognition in a single forward pass. This system is trained using

[1] https://github.com/Bartzi/see

[2] https://bartzi.de/research/see

groundtruth bounding boxes and groundtruth labels for each word in the input images, which stands in contrast to our method, where we only use groundtruth labels for each word in the input image, as the detection of text is learned by the network itself.

Proposed System

A human trying to find and read text will do so in a sequential manner. The first action is to put attention on a word, read each character sequentially and then attend to the next word. Most current end-to-end systems for scene text recognition do not behave in that way. These systems rather try to solve the problem by extracting all information from the image at once. Our system first tries to attend sequentially to different text regions in the image and then recognize their textual content. In order to do this, we created a single DNN consisting of two stages: (1) text detection, and (2) text recognition. In this section we will introduce the attention concept used by the text detection stage and the overall structure of the proposed system.

Detecting Text with Spatial Transformers

A spatial transformer proposed by Jaderberg et al. (Jaderberg et al. 2015b) is a differentiable module for DNNs that takes an input feature map I and applies a spatial transformation to this feature map, producing an output feature map O. Such a spatial transformer module is a combination of three parts. The first part is a localization network computing a function f_{loc}, that predicts the parameters θ of the spatial transformation to be applied. These predicted parameters are used in the second part to create a sampling grid, which defines a set of points where the input map should be sampled. The third part is a differentiable interpolation method, that takes the generated sampling grid and produces the spatially transformed output feature map O. We will shortly describe each component in the following paragraphs.

Localization Network The localization network takes the input feature map $I \in \mathbb{R}^{C\times H\times W}$, with C channels, height H and width W and outputs the parameters θ of the transformation that shall be applied. In our system we use the localization network (f_{loc}) to predict N two-dimensional affine transformation matrices A^n_θ, where $n \in \{0, \dots, N-1\}$:

$$f_{loc}(I) = A^n_\theta = \begin{bmatrix} \theta^n_1 & \theta^n_2 & \theta^n_3 \\ \theta^n_4 & \theta^n_5 & \theta^n_6 \end{bmatrix} \tag{1}$$

N is thereby the number of characters, words or textlines the localization network shall localize. The affine transformation matrices predicted in that way allow the network to apply translation, rotation, zoom and skew to the input image.

In our system the N transformation matrices A^n_θ are produced by using a feed-forward CNN together with a Recurrent Neural Network (RNN). Each of the N transformation matrices is computed using the globally extracted convolutional features c and the hidden state h_n of each time-step of the RNN:

$$c = f^{conv}_{loc}(I) \tag{2}$$
$$h_n = f^{rnn}_{loc}(c, h_{n-1}) \tag{3}$$
$$A^n_\theta = g_{loc}(h_n) \tag{4}$$

where g_{loc} is another feed-forward/recurrent network. We use a variant of the well known ResNet architecture (He et al. 2016a) as CNN for our localization network. We use this network architecture, because we found that with this network structure our system learns faster and more successfully, as compared to experiments with other network structures, such as the VGGNet (Simonyan and Zisserman 2015). We argue that this is due to the fact that the residual connections of the ResNet help with retaining a strong gradient down to the very first convolutional layers. The RNN used in the localization network is a Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) unit. This LSTM is used to generate the hidden states h_n, which in turn are used to predict the affine transformation matrices. We used the same structure of the network for all our experiments we report in the next section. Figure 2 provides a structural overview of this network.

Rotation Dropout During our experiments, we found that the network tends to predict transformation parameters, which include excessive rotation. In order to mitigate such a behavior, we propose a mechanism that works similarly to dropout (Srivastava et al. 2014), which we call rotation dropout. Rotation dropout works by randomly dropping the parameters of the affine transformation, which are responsible for rotation. This prevents the localization network to output transformation matrices that perform excessive rotation. Figure 3 shows a comparison of the localization result of a localization network trained without rotation dropout (top) and one trained with rotation dropout (middle).

Grid Generator The grid generator uses a regularly spaced grid G_o with coordinates y_{h_o}, x_{w_o}, of height H_o and width W_o. The grid G_o is used together with the affine transformation matrices A^n_θ to produce N regular grids G^n with coordinates u^n_i, v^n_j of the input feature map I, where $i \in H_o$ and $j \in W_o$:

$$\begin{pmatrix} u^n_i \\ v^n_j \end{pmatrix} = A^n_\theta \begin{pmatrix} x_{w_o} \\ y_{h_o} \\ 1 \end{pmatrix} = \begin{bmatrix} \theta^n_1 & \theta^n_2 & \theta^n_3 \\ \theta^n_4 & \theta^n_5 & \theta^n_6 \end{bmatrix} \begin{pmatrix} x_{w_o} \\ y_{h_o} \\ 1 \end{pmatrix} \tag{5}$$

During inference we can extract the N resulting grids G^n, which contain the bounding boxes of the text regions found by the localization network. Height H_o and width W_o can be chosen freely.

Localization specific regularizers The datasets used by us, do not contain any samples, where text is mirrored either along the x- or y-axis. Therefore, we found it beneficial to add additional regularization terms that penalizes grid, which are mirrored along any axis. We furthermore found that the network tends to predict grids that get larger over

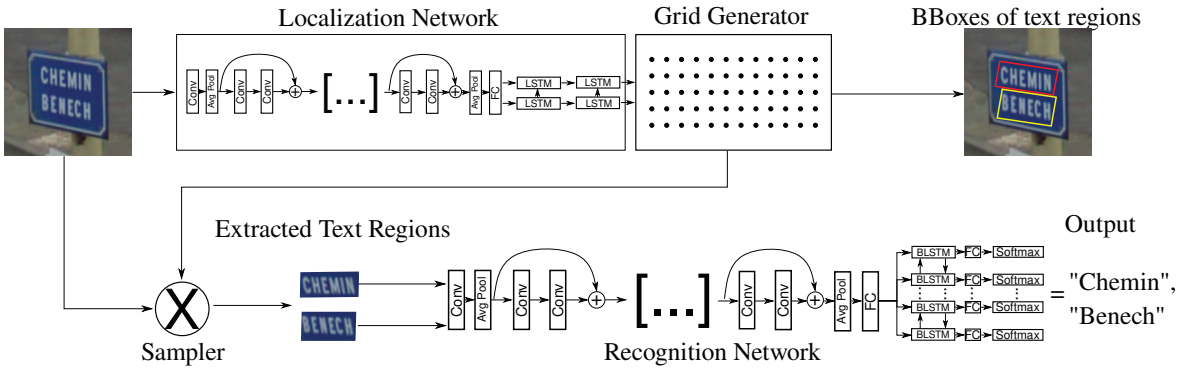

Figure 2: The network used in our work consists of two major parts. The first is the localization network that takes the input image and predicts N transformation matrices, which are used to create N different sampling grids. The generated sampling grids are used in two ways: (1) for calculating the bounding boxes of the identified text regions (2) for extracting N text regions. The recognition network then performs text recognition on these extracted regions. The whole system is trained end-to-end by only supplying information about the text labels for each text region.

the time of training, hence we included a further regularizer that penalizes large grids, based on their area. Lastly, we also included a regularizer that encourages the network to predict grids that have a greater width than height, as text is normally written in horizontal direction and typically wider than high. The main purpose of these localization specific regularizers is to enable faster convergence. Without these regularizers, the network will eventually converge, but it will take a very long time and might need several restarts of the training. Equation 7 shows how these regularizers are used for calculating the overall loss of the network.

Image Sampling The N sampling grids G^n produced by the grid generator are now used to sample values of the feature map I at the coordinates u_i^n, v_j^n for each $n \in N$. Naturally these points will not always perfectly align with the discrete grid of values in the input feature map. Because of that we use bilinear sampling and define the values of the N output feature maps O^n at a given location i, j where $i \in H_o$ and $j \in W_o$ to be:

$$O_{ij}^n = \sum_h^H \sum_w^W I_{hw} max(0, 1-|u_i^n - h|) max(0, 1-|v_j^n - w|) \tag{6}$$

This bilinear sampling is (sub-)differentiable, hence it is possible to propagate error gradients to the localization network, using standard backpropagation.

The combination of localization network, grid generator and image sampler forms a spatial transformer and can in general be used in every part of a DNN. In our system we use the spatial transformer as the first step of our network. Figure 4 provides a visual explanation of the operation method of grid generator and image sampler.

Text Recognition Stage

The image sampler of the text detection stage produces a set of N regions, that are extracted from the original input image. The text recognition stage (a structural overview of this stage can be found in Figure 2) uses each of these N different regions and processes them independently of each other. The processing of the N different regions is handled by a CNN. This CNN is also based on the ResNet architecture as we found that we could only achieve good results, while using a variant of the ResNet architecture for our recognition network. We argue that using a ResNet in the recognition stage is even more important than in the detection stage, because the detection stage needs to receive strong gradient information from the recognition stage in order to successfully update the weights of the localization network. The CNN of the recognition stage predicts a probability distribution $\hat{y}$ over the label space L_ϵ, where $L_\epsilon = L \cup \{\epsilon\}$, with L being the alphabet used for recognition, and ϵ representing the blank label. The network is trained by running a LSTM for a fixed number of T timesteps and calculating the cross-entropy loss for the output of each timestep. The choice of number of timesteps T is based on the number of characters, of the longest word, in the dataset. The loss $\mathcal{L}$ is computed as follows:

$$\mathcal{L}_{grid}^n = \lambda_1 \times \mathcal{L}_{ar}(G^n) + \lambda_2 \times \mathcal{L}_{as}(G^n) + \mathcal{L}_{di}(G^n) \tag{7}$$

$$\mathcal{L} = \sum_{n=1}^{N} (\sum_{t=1}^{T} (P(l_t^n | O^n)) + \mathcal{L}_{grid}^n) \tag{8}$$

Where $\mathcal{L}_{ar}(G^n)$ is the regularization term based on the area of the predicted grid n, $\mathcal{L}_{as}(G^n)$ is the regularization term based on the aspect ratio of the predicted grid n, and $\mathcal{L}_{di}(G^n)$ is the regularization term based on the direction of the grid n, that penalizes mirrored grids. λ_1 and λ_2 are scaling parameters that can be chosen freely. The typical range of these parameters is $0 < \lambda_1, \lambda_2 < 0.5$. l_t^n is the label l at time step t for the n-th word in the image.

Model Training

The training set X used for training the model consists of a set of input images I and a set of text labels L_I for each

Figure 3: *Top:* predicted bounding boxes of network trained without rotation dropout. *Middle:* predicted bounding boxes of network trained with rotation dropout. *Bottom:* visualization of image parts that have the highest influence on the outcome of the prediction. This visualization has been created using Visualbackprop (Bojarski et al. 2016).

input image. We do not use any labels for training the text detection stage. The text detection stage is learning to detect regions of text by using only the error gradients, obtained by calculating the cross-entropy loss, of the predictions and the textual labels, for each character of each word. During our experiments we found that, when trained from scratch, a network that shall detect and recognize more than two text lines does not converge. In order to overcome this problem we designed a curriculum learning strategy (Bengio et al. 2009) for training the system. The complexity of the supplied training images under this curriculum is gradually increasing, once the accuracy on the validation set has settled.

During our experiments we observed that the performance of the localization network stagnates, as the accuracy of the recognition network increases. We found that restarting the training with the localization network initialized using the weights obtained by the last training and the recognition network initialized with random weights, enables the localization network to improve its predictions and thus improve the overall performance of the trained network. We argue that this happens because the values of the gradients propagated to the localization network decrease, as the loss decreases, leading to vanishing gradients in the localization network and hence nearly no improvement of the localization.

Experiments

In this section we evaluate our presented network architecture on standard scene text detection/recognition benchmark datasets. While performing our experiments we tried to answer the following questions: (1) Is the concept of letting the network automatically learn to detect text feasible? (2) Can we apply the method on a real world dataset? (3) Can we get any insights on what kind of features the network is trying to extract?

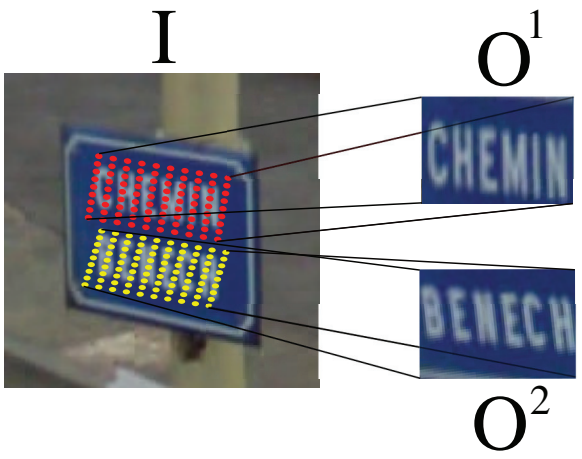

Figure 4: Operation method of grid generator and image sampler. First the grid generator uses the N affine transformation matrices A_θ^n to create N equally spaced sampling grids (red and yellow grids on the left side). These sampling grids are used by the image sampler to extract the image pixels at that location, in this case producing the two output images O^1 and O^2. The corners of the generated sampling grids provide the vertices of the bounding box for each text region, that has been found by the network.

In order to answer these questions, we used different datasets. On the one hand we used standard benchmark datasets for scene text recognition. On the other hand we generated some datasets on our own. First, we performed experiments on the SVHN dataset (Netzer et al. 2011), that we used to prove that our concept as such is feasible. Second, we generated more complex datasets based on SVHN images, to see how our system performs on images that contain several words in different locations. The third dataset we experimented with, was the French Street Name Signs (FSNS) dataset (Smith et al. 2016). This dataset is the most challenging we used, as it contains a vast amount of irregular, low resolution text lines, that are more difficult to locate and recognize than text lines from the SVHN datasets. We begin this section by introducing our experimental setup. We will then present the results and characteristics of the experiments for each of the aforementioned datasets. We will conclude this section with a brief explanation of what kinds of features the network seems to learn.

Experimental Setup

Localization Network The localization network used in every experiment is based on the ResNet architecture (He et al. 2016a). The input to the network is the image where text shall be localized and later recognized. Before the first residual block the network performs a 3×3 convolution,

followed by batch normalization (Ioffe and Szegedy 2015), ReLU (Nair and Hinton 2010), and a 2×2 average pooling layer with stride 2. After these layers three residual blocks with two 3×3 convolutions, each followed by batch normalization and ReLU, are used. The number of convolutional filters is 32, 48 and 48 respectively. A 2×2 max-pooling with stride 2 follows after the second residual block. The last residual block is followed by a 5×5 average pooling layer and this layer is followed by a LSTM with 256 hidden units. Each time step of the LSTM is fed into another LSTM with 6 hidden units. This layer predicts the affine transformation matrix, which is used to generate the sampling grid for the bilinear interpolation. We apply rotation dropout to each predicted affine transformation matrix, in order to overcome problems with excessive rotation predicted by the network.

Recognition Network The inputs to the recognition network are N crops from the original input image, representing the text regions found by the localization network. In our SVHN experiments, the recognition network has the same structure as the localization network, but the number of convolutional filters is higher. The number of convolutional filters is 32, 64 and 128 respectively. We use an ensemble of T independent softmax classifiers as used in (Goodfellow et al. 2014) and (Jaderberg, Vedaldi, and Zisserman 2014) for generating our predictions. In our experiments on the FSNS dataset we found that using ResNet-18 (He et al. 2016a) significantly improves the obtained recognition accuracies.

Alignment of Groundtruth During training we assume that all groundtruth labels are sorted in western reading direction, that means they appear in the following order: 1. from top to bottom, and 2. from left to right. We stress that currently it is very important to have a consistent ordering of the groundtruth labels, because if the labels are in a random order, the network rather predicts large bounding boxes that span over all areas of text in the image. We hope to overcome this limitation, in the future, by developing a method that allows random ordering of groundtruth labels.

Implementation We implemented all our experiments using Chainer (Tokui et al. 2015). We conducted all our experiments on a work station which has an Intel(R) Core(TM) i7-6900K CPU, 64 GB RAM and 4 TITAN X (Pascal) GPUs.

Experiments on the SVHN dataset

With our first experiments on the SVHN dataset (Netzer et al. 2011) we wanted to prove that our concept works. We therefore first conducted experiments, similar to the experiments in (Jaderberg et al. 2015b), on SVHN image crops with a single house number in each image crop, that is centered around the number and also contains background noise. Table 1 shows that we are able to reach competitive recognition accuracies.

Based on this experiment we wanted to determine whether our model is able to detect different lines of text that are arranged in a regular grid, or placed at random locations in the image. In Figure 5 we show samples from our two generated datasets, that we used for our other experiments based on SVHN data. We found that our network performs well on the task of finding and recognizing house numbers that are arranged in a regular grid.

Method	64px
(Goodfellow et al. 2014)	96.0 %
(Jaderberg et al. 2015b)	96.3 %
Ours	95.2 %

Table 1: Sequence recognition accuracies on the SVHN dataset. When recognizing house number on crops of 64×64 pixels, following the experimental setup of (Goodfellow et al. 2014)

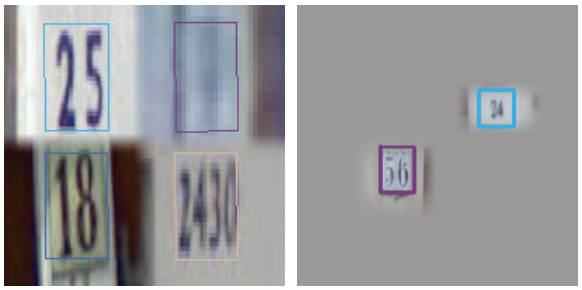

Figure 5: Samples from our generated datasets, including bounding boxes predicted by our model. *Left:* Sample from regular grid dataset, *Right:* Sample from dataset with randomly positioned house numbers.

During our experiments on the second dataset, created by us, we found that it is not possible to train a model from scratch, which can find and recognize more than two textlines that are scattered across the whole image. We therefore resorted to designing a curriculum learning strategy that starts with easier samples first and then gradually increases the complexity of the train images.

Experiments on the FSNS dataset

Following our scheme of increasing the difficulty of the task that should be solved by the network, we chose the French Street Name Signs (FSNS) dataset by Smith et al. (Smith et al. 2016) to be our next dataset to perform experiments on. The FSNS dataset contains more than 1 million images of French street name signs, which have been extracted from Google Streetview. This dataset is the most challenging dataset for our approach as it (1) contains multiple lines of text with varying length, which are embedded in natural scenes with distracting backgrounds, and (2) contains a lot of images where the text is occluded, not correct, or nearly unreadable for humans.

During our first experiments with that dataset, we found that our model is not able to converge, when trained on the supplied groundtruth. We argue that this is because our network was not able to learn the alignment of the supplied labels with the text in the images of the dataset. We therefore chose a different approach, and started with experiments

Figure 6: Samples from the FSNS dataset, these examples show the variety of different samples in the dataset and also how well our system copes with these samples. The bottom row shows two samples, where our system fails to recognize the correct text. The right image is especially interesting, as the system here tries to mix information, extracted from two different street signs, that should not be together in one sample.

Method	Sequence Accuracy
(Smith et al. 2016)	72.5 %
(Wojna et al. 2017)	84.2 %
Ours	78.0 %

Table 2: Recognition accuracies on the FSNS benchmark dataset.

where we tried to find individual words instead of textlines with more than one word. Table 2 shows the performance of our proposed system on the FSNS benchmark dataset. We are currently able to achieve competitive performance on this dataset. We are still behind the results reported by Wojna et al. (Wojna et al. 2017). This likely due to the fact that we used a feature extractor that is weaker (ResNet-18) compared to the one used by Wojna et al. (Inception-ResNet v2). Also recall that our method is not only able to determine the text in the images, but also able to extract the location of the text, although we never explicitly told the network where to find the text! The network learned this completely on its own in a semi-supervised manner.

Insights

During the training of our networks, we used Visualbackprop (Bojarski et al. 2016) to visualize the regions that the network deems to be the most interesting. Using this visualization technique, we could observe that our system seems to learn different types of features for each subtask. Figure 3 (bottom) shows that the localization network learns to extract features that resemble edges of text and the recognition network learns to find strokes of the individual characters in each cropped word region. This is an interesting observation, as it shows that our DNN tries to learn features that are closely related to the features used by systems based on hand-crafted features.

Conclusion

In this paper we presented a system that can be seen as a step towards solving end-to-end scene text recognition, only using a single multi-task deep neural network. We trained the text detection component of our model in a semi-supervised way and are able to extract the localization results of the text detection component. The network architecture of our system is simple, but it is not easy to train this system, as a successful training requires a clever curriculum learning strategy. We also showed that our network architecture can be used to reach competitive results on different public benchmark datasets for scene text detection/recognition.

At the current state we note that our models are not fully capable of detecting text in arbitrary locations in the image, as we saw during our experiments with the FSNS dataset. Right now our model is also constrained to a fixed number of maximum words that can be detected with one forward pass. In our future work, we want to redesign the network in a way that makes it possible for the network to determine the number of textlines in an image by itself.

References

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 41–48. New York, NY, USA: ACM.

Bissacco, A.; Cummins, M.; Netzer, Y.; and Neven, H. 2013. Photoocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE International Conference on Computer Vision*, 785–792.

Bojarski, M.; Choromanska, A.; Choromanski, K.; Firner, B.;

Jackel, L.; Muller, U.; and Zieba, K. 2016. Visualbackprop: efficient visualization of cnns. *arXiv:1611.05418 [cs]*.

Dai, J.; He, K.; and Sun, J. 2016. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3150–3158.

Epshtein, B.; Ofek, E.; and Wexler, Y. 2010. Detecting text in natural scenes with stroke width transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2963–2970.

Goodfellow, I.; Bulatov, Y.; Ibarz, J.; Arnoud, S.; and Shet, V. 2014. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *ICLR2014*.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2315–2324.

Gómez, L., and Karatzas, D. 2017. Textproposals: A text-specific selective search algorithm for word spotting in the wild. *Pattern Recognition* 70:60–74.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

He, P.; Huang, W.; Qiao, Y.; Loy, C. C.; and Tang, X. 2016b. Reading scene text in deep convolutional sequences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 3501–3508. AAAI Press.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, 448–456.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2015a. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision* 116(1):1–20.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; and Kavukcuoglu, K. 2015b. Spatial transformer networks. In *Advances in Neural Information Processing Systems 28*, 2017–2025. Curran Associates, Inc.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Deep features for text spotting. In *Computer Vision - ECCV 2014*, number 8692 in Lecture Notes in Computer Science, 512–528. Springer International Publishing.

Li, H.; Wang, P.; and Shen, C. 2017. Towards end-to-end text spotting with convolutional recurrent neural networks. *arXiv:1707.03985 [cs]*.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012. Scene text recognition using higher order language priors. In *BMVC 2012-23rd British Machine Vision Conference*, 127.1–127.11. British Machine Vision Association.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 5.

Neumann, L., and Matas, J. 2010. A method for text localization and recognition in real-world images. In *Computer Vision - ACCV 2010*, number 6494 in Lecture Notes in Computer Science, 770–783. Springer Berlin Heidelberg.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28*, 91–99. Curran Associates, Inc.

Shi, B.; Bai, X.; and Yao, C. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4168–4176.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Smith, R.; Gu, C.; Lee, D.-S.; Hu, H.; Unnikrishnan, R.; Ibarz, J.; Arnoud, S.; and Lin, S. 2016. End-to-end interpretation of the french street name signs dataset. In *Computer Vision - ECCV 2016 Workshops*, 411–426. Springer, Cham.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15(1):1929–1958.

Sønderby, S. K.; Sønderby, C. K.; Maaløe, L.; and Winther, O. 2015. Recurrent spatial transformer networks. *arXiv:1509.05329 [cs]*.

Tokui, S.; Oono, K.; Hido, S.; and Clayton, J. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

Wang, K., and Belongie, S. 2010. Word spotting in the wild. In *Computer Vision - ECCV 2010*, number 6311 in Lecture Notes in Computer Science, 591–604. Springer Berlin Heidelberg.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, 1457–1464.

Wojna, Z.; Gorban, A.; Lee, D.-S.; Murphy, K.; Yu, Q.; Li, Y.; and Ibarz, J. 2017. Attention-based extraction of structured information from street view imagery. *arXiv:1704.03549 [cs]*.

Yao, C.; Bai, X.; Shi, B.; and Liu, W. 2014. Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4042–4049.