

# STN-OCR: A single Neural Network for Text Detection and 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 STN-OCR, a step towards semi-supervised neural networks for scene text recognition that can be optimized end-to-end. In contrast to most existing works that consist of multiple deep neural networks and several pre-processing steps we propose to use a single deep neural network that learns to detect and recognize text from natural images in a semi-supervised way. STN-OCR is a network that integrates and jointly learns a spatial transformer network [16], that 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 investigate how our model behaves on a range of different tasks (detection and recognition of characters, and lines of text). Experimental results on public benchmark datasets show the ability of our model to handle a variety of different tasks, without substantial changes in its overall network structure.*

## 1. 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 can 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 [10, 16, 25, 26] in other areas of computer vision enabled the creation of even better scene text detection and recognition systems than before [5, 9, 28]. Although the problem of Optical

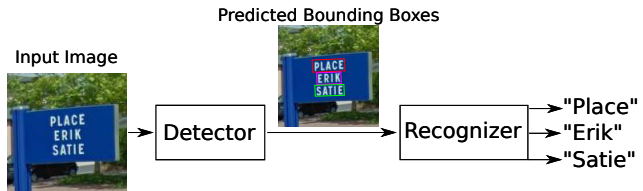


Figure 1. Schematic overview of our proposed system. The input image is fed to a single neural network that consists of a text detection part and a text recognition part. The text detection part learns to detect text in a semi-supervised way, by being jointly trained with the recognition part.

Character Recognition (OCR) can be seen as solved for printed document texts 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.*

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. This is contrary to existing works, where text detection and text recognition systems are trained separately in a fully-supervised way. Recent work [3] 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 scene text recognition system that behaves more like a human by dividing the task at hand into smaller subtasks and solving these subtask independently from each other. In order to achieve this behavior we learn a single DNN that is able to divide the input im-

age into subtasks (single characters, words or even lines of text) and solve these subtasks independently of each other. This is achieved by jointly learning a localization network that uses a recurrent spatial transformer [16, 31] as attention mechanism and a text recognition network (see Figure 1 for a schematic overview of the system). In this setting the network only receives the image and the labels for the text contained in that image as input. The localization of the text is learned by the network itself, making this approach semi-supervised.

Our contributions are as follows: (1) We present a system that is a step towards solving end-to-end scene text recognition by integrating spatial transformer networks. (2) We train our proposed system end-to-end in a semi-supervised way. (3) We demonstrate that our approach is able to reach state-of-the-art/competitive performance on a range of standard scene text detection and recognition benchmarks. (4) We provide our code<sup>1</sup> and trained models<sup>2</sup> to the research community.

This paper is structured in the following way: In section 2 we outline work of other researchers related to ours. Section 3 describes our proposed system in detail and provides best practices on how to train such a system. We show and discuss our results on standard benchmark datasets in section 4 and conclude our findings in section 5.

## 2. 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 these 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. We will discuss some of these systems for each category below.

**Hand Crafted Features** In the beginning methods based on hand crafted features and human knowledge have been used to perform text detection. These systems used features like MSERs [24], Stroke Width Transforms [4] or HOG-Features [32] to identify regions of text and provide them to

the text recognition stage of the system. In the text recognition stage sliding window classifiers [21] and ensembles of SVMs [34] or k-Nearest Neighbor classifiers using HOG features [33] 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 end-to-end recognition systems by approaches using DNNs. Gómez and Karatzas [5] 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.* [9] propose a text detection model based on the YOLO-Architecture [25] that uses a fully convolutional deep neural network to identify text regions. The text regions identified by these approaches can then be used as input for further systems based on DNNs that perform text recognition.

Bissacco *et al.* [1] propose a complete end-to-end architecture that performs text detection using hand crafted features. The identified text regions are binarized and then used as input to a deep fully connected neural network that classifies each found character independently. Jaderberg *et al.* [15, 17] propose several systems that use deep neural networks for text detection and text recognition. In [17] Jaderberg *et al.* propose a sliding window text detection approach that slides a convolutional text detection model across the image in multiple resolutions. The text recognition stage uses a single character CNN, which is slid across the identified text region. This CNN shares its weights with the CNN used for text detection. In [15] Jaderberg *et al.* propose to use a region proposal network with an extra bounding box regression CNN for text detection and a CNN that takes the whole text region as input and performs classification across a pre-defined dictionary of words, making this approach only applicable to one given language.

Goodfellow *et al.* [6] propose a text recognition system for house numbers, that has been refined by Jaderberg *et al.* [18] for unconstrained text recognition. This system uses a single CNN, which takes the complete extracted text region as input, and provides the text contained in that text region. This is achieved by having one independent classifier for each possible character in the given word. Based on this idea He *et al.* [11] and Shi *et al.* [27, 28] propose text recognition systems that treat the recognition of characters from the extracted text region as a sequence recognition problem. He *et al.* [11] use a naive sliding window approach that creates slices of the text region, which are used as input to their text recognition CNN. The features produced by the text recognition CNN are used as input to a Recurrent Neural Network (RNN) that predicts the sequence of

<sup>1</sup><https://github.com/Bartzi/stn-ocr>

<sup>2</sup><https://bartzi.de/research/stn-ocr>

characters. In our experiments on pure scene text recognition (see section 4.3 for more information) we use a similar approach, but our system uses a more sophisticated sliding window approach, where the choice of the sliding windows is automatically learned by the network and not engineered by hand. Shi *et al.* [27] utilize a CNN that uses the complete text region as input and produces a sequence of feature vectors, which are fed to a RNN that predicts the sequence of characters in the extracted text region. This approach generates a fixed number of feature vectors based on the width of the text region. That means for a text region that only contains a few characters, but has the same width as a text region with sufficiently more characters, this approach will produce the same amount of feature vectors used as input to the RNN. In our pure text recognition experiments we utilized the strength of our approach to learn to attend to the most important information in the extracted text region, hence producing only as many feature vectors as necessary. Shi *et al.* [28] improve their approach by firstly adding an extra step that utilizes the rectification capabilities of Spatial Transformer Networks [16] for rectifying the extracted text line. 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.* [30] propose an end-to-end trainable system that is able to detect and 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. Furthermore the attention mechanism used in our approach shows a more human-like behaviour because it is sequentially localizes and recognizes text from the given image.

### 3. 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 line of text, read each character sequentially and then attend to the next line of text. Most current end-to-end systems for scene text recognition do not behave in that way. These

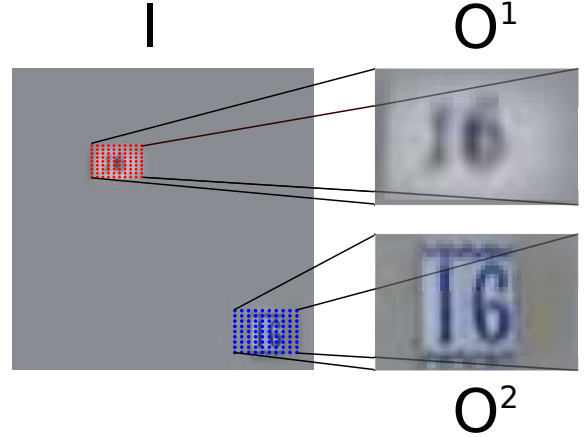


Figure 2. Operation method of grid generator and image sampler. First the grid generator uses the  $N$  affine transformation matrices  $A_{\theta}^n$  to create  $N$  equally spaced sampling grids (red and blue grids on the left side). These sampling grids are used by the image sampler to extract the 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.

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 the textual content of each text region. In order to this we created a simple DNN consisting of two stages: (1) text detection (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. We also report best practices for successfully training such a system.

#### 3.1. Detecting Text with Spatial Transformers

A spatial transformer proposed by Jaderberg *et al.* [16] 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 localisation network computing a function  $f_{loc}$ , that predicts the parameters  $\theta$  of the spatial transformation to be applied. These predicted parameters are used in the second part to create a sampling grid that defines which features of the input feature map should be mapped to the output feature map. 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  $\theta$  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  $A_\theta^n$ , where  $n \in \{0, \dots, N-1\}$ :

$$f_{loc}(I) = A_\theta^n = \begin{bmatrix} \theta_1^n & \theta_2^n & \theta_3^n \\ \theta_4^n & \theta_5^n & \theta_6^n \end{bmatrix} \quad (1)$$

$N$  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, hence the network learns to produce transformation parameters that can zoom on characters, words or text lines that are to be extracted from the image.

In our system the  $N$  transformation matrices  $A_\theta^n$  are produced by using a feed-forward CNN together with a RNN. Each of the  $N$  transformation matrices is computed using the hidden state  $h_n$  for each time-step of the RNN:

$$c = f_{loc}^{conv}(I) \quad (2)$$

$$h_n = f_{loc}^{rnn}(c, h_{n-1}) \quad (3)$$

$$A_\theta^n = g_{loc}(h_n) \quad (4)$$

where  $g_{loc}$  is another feed-forward network, and each transformation matrix  $A_\theta^n$  is conditioned on the globally extracted convolutional features ( $f_{loc}^{conv}$ ) together with the hidden state of the previously performed time-step.

The CNN in the localization network used by us is a variant of the well known ResNet by He *et al.* [10]. We use a variant of ResNet because we found that with this network structure our system learns faster and more successful, as compared to experiments with other network structures like the VGGNet [29]. 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. In addition to the structure we also used Batch Normalization [13] for all our experiments. The RNN used in the localization network is a Bidirectional Long-Short Term Memory (BLSTM) [8, 12] unit. This BLSTM 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 section 4. Figure 3 provides a structural overview of this network.

**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  $W_o$ , together with the affine transformation matrices  $A_\theta^n$  to produce  $N$  regular grids  $G_i^n$  of coordinates  $u_i^n, v_j^n$  of

the input feature map  $I$ , where  $i \in H_o$  and  $j \in W_o$ :

$$\begin{pmatrix} u_i^n \\ v_j^n \end{pmatrix} = A_\theta^n \begin{pmatrix} x_{w_o} \\ y_{h_o} \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_1^n & \theta_2^n & \theta_3^n \\ \theta_4^n & \theta_5^n & \theta_6^n \end{bmatrix} \begin{pmatrix} x_{w_o} \\ y_{h_o} \\ 1 \end{pmatrix} \quad (5)$$

During inference we can extract the  $N$  resulting grids  $G_i^n$  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 and if they are lower than height  $H$  or width  $W$  of the input feature map  $I$  the grid generator is producing a grid that performs a downsampling operation in the next step.

**Image Sampling** The  $N$  sampling grids  $G_i^n$  produced by the grid generator are now used to sample values of the feature map  $I$  at their corresponding 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 that extracts the value at a given coordinate by bilinear interpolating the values of the nearest neighbors. With that we define the values of the  $N$  output feature maps  $O^n$  at a given location  $i, j$  where  $i \in H_o$  and  $j \in W_o$ :

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

This bilinear sampling is (sub-)differentiable, hence it is possible to propagate error gradients to the localization network by 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. The localization network receives the input image as input feature map and produces a set of affine transformation matrices that are used by the grid generator to calculate the position of the pixels that shall be sampled by the bilinear sampling operation.

### 3.2. 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 3) 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 if we use 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



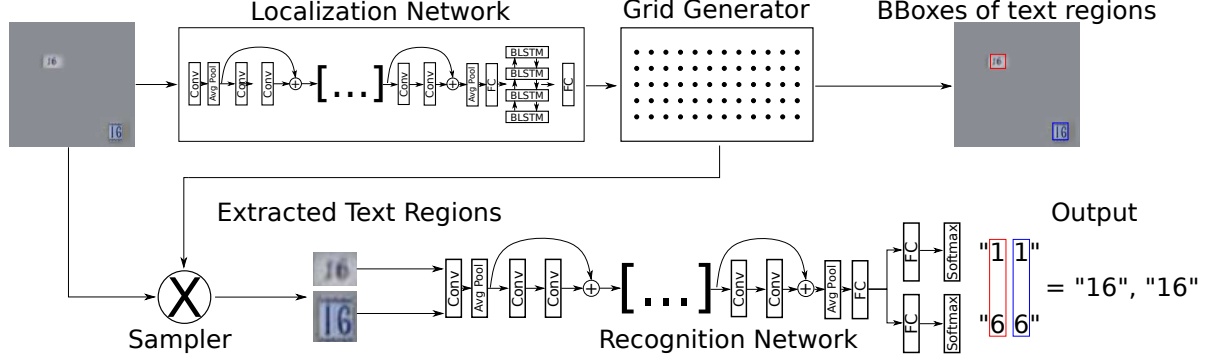


Figure 3. The network used in our work consists of two major parts. The first is the localization network that takes the input image and predicts  $N$  transformation matrices, that are applied to  $N$  identical grids, forming  $N$  different sampling grids. The generated sampling grids are used in two ways: (1) for calculating the bounding boxes of the identified text regions (2) for sampling the input image with  $N$  sampling grids to extract  $N$  text regions. The  $N$  extracted text images are then used in the recognition network to perform text recognition. The whole system is trained end-to-end by only supplying information about the text labels for each text region.

stage, because the detection stage needs to receive strong gradients 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  $L_\epsilon$ , where  $L_\epsilon = L \cup \{\epsilon\}$ , with  $L = \{0 - 9a - z\}$  and  $\epsilon$  representing the blank label. Depending on the task this probability distribution is either generated by a fixed number of  $T$  softmax classifiers, where each softmax classifier is used to predict one character of the given word:

$$x^n = O^n \quad (7)$$

$$\hat{y}_t^n = \text{softmax}(f_{\text{rec}}(x^n)) \quad (8)$$

$$\hat{y}^n = \sum_{t=1}^T \hat{y}_t^n \quad (9)$$

where  $f_{\text{rec}}(x)$  is the result of applying the convolutional feature extractor on the sampled input  $x$ .

Another possibility is to train the network using Connectionist Temporal Classification (CTC) [7] and retrieve the most probable labeling by setting  $\hat{y}$  to be the most probable labeling path  $\pi$ , that is given by:

$$p(\pi|x^n) = \prod_{t=1}^T \hat{y}_{\pi_t}^n, \forall \pi \in L_\epsilon^T \quad (10)$$

$$\hat{y}_t^n = \text{argmax}_p(\pi|x^n) \quad (11)$$

$$\hat{y}^n = B(\sum_{t=1}^T \hat{y}_t^n) \quad (12)$$

with  $L_\epsilon^T$  being the set of all labels that have the length  $T$  and  $p(\pi|x^n)$  being the probability that path  $\pi \in L_\epsilon^T$  is predicted by the DNN.  $B$  is a function that removes all predicted blank labels and all repeated labels (e.g.  $B(\text{IC-CC-V}) = B(\text{II-CCC-C-V}) = \text{ICCV}$ ).

### 3.3. 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 input image. We do not use any labels for training the text detection stage. This stage is learning to detect regions of text only by using the error gradients obtained by either calculating the cross-entropy loss or the CTC loss of the predictions and the textual labels. 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. The solution to this problem is to perform a series of pre-training steps where the difficulty is gradually increasing. Furthermore we find that the optimization algorithm chosen to train the network has a great influence on the convergence of the network. We found that it is beneficial to use Stochastic Gradient Descent (SGD) for pre-training the network on a simpler task and Adam [20] for finetuning the already pre-trained network on images with more text lines. We argue that SGD performs better during pre-training because the learning rate  $\eta$  is kept constant during a longer period of time, which enables the text detection stage to explore the input images and better find text regions. With decreasing learning rate the updates in the detection stage become smaller and the text detection stage (ideally) settles on already found text regions. At the same time the text recognition network can start to use the extracted text regions and learn to recognize the text in that regions. While training the network with SGD it is important to note that choosing a too high learning rate will result in divergence of the model early on. We found that using initial learning rates between  $1^{-5}$  and  $5^{-7}$  tend to work in nearly all cases, except in cases where the network should only be fine-tuned. Here we found that using Adam is the more reliable choice, as Adam chooses the learning rate for

each parameter in an adaptive way and hence does not allow the detection network to explore as radically as it does when using SGD.

## 4. Experiments

In this section we evaluate our presented network architecture on several standard scene text detection/recognition datasets. We present the results of experiments for three different datasets, where the difficulty of the task at hand increases for each dataset. We first begin with experiments on the SVHN dataset [23], that we used to prove that our concept as such is feasible. The second type of dataset we performed experiments on were datasets for focused scene text recognition, where we explored the performance of our model, when it comes to find and recognize single characters. The third dataset we experimented with was the French Street Name Signs (FSNS) dataset [30], which is the most challenging dataset we used, as this dataset contains a vast amount of irregular, low resolution text lines that are more difficult to locate and recognize than text lines from the SVHN dataset. 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.

### 4.1. Experimental Setup

**Localization Network** The localization network used in every experiment is based on the ResNet architecture [10]. 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  $3 \times 3$  convolution followed by a  $2 \times 2$  average pooling layer with stride 2. After these layers three residual blocks with two  $3 \times 3$  convolutions, each followed by batch normalization [13], are used. The number of convolutional filters is 32, 48 and 48 respectively and ReLU [22] is used as activation function for each convolutional layer. A  $2 \times 2$  max-pooling with stride 2 follows after the second residual block. The last residual block is followed by a  $5 \times 5$  average pooling layer and this layer is followed by a BLSTM with 256 hidden units. For each time step of the BLSTM a fully connected layer with 6 hidden units follows. This layer predicts the affine transformation matrix, that is used to generate the sampling grid for the bilinear interpolation. As rectification of scene text is beyond the scope of this work we disabled skew and rotation in the affine transformation matrices by setting the according parameters to 0. We will discuss the rectification capabilities of Spatial Transformers for scene text detection in our future work.

**Recognition Network** The inputs to the recognition network are  $N$  crops from the original input image that represent the text regions found by the localization network.

Method	64px
Maxout CNN [6]	96
ST-CNN [16]	96.3
Ours	95.2

Table 1. Sequence recognition accuracies on the SVHN dataset. When recognizing house number on crops of  $64 \times 64$  pixels, following the experimental setup of [6]

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. Depending on the experiment we either used an ensemble of  $T$  independent softmax classifiers as used in [6] and [17], where  $T$  is the maximum length that a word may have, or we used CTC with best path decoding as used in [11] and [27].

**Implementation** We implemented all our experiments using MXNet [2]. 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.

### 4.2. Experiments on the SVHN dataset

With our first experiments on the SVHN dataset [23] we wanted to prove that our concept works and can be used with real world data. We therefore first conducted experiments similar to the experiments in [16] 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 4 we show samples from two purpose build datasets<sup>3</sup> 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. An interesting observation we made during training on this data was that we were able to achieve our best results when we did two training steps. The first step was to train the complete model from scratch (all weights initialized randomly) and then train the model with the same data again, but this time with the localization network pre-initialized with the weights obtained from the last training and the recognition net initialized with random weights. This strategy leads to better localization results of the localization network and hence improved recognition results.

During our experiments on the second dataset, created

<sup>3</sup>datasets are available here: <https://bartzi.de/research/stn-ocr>

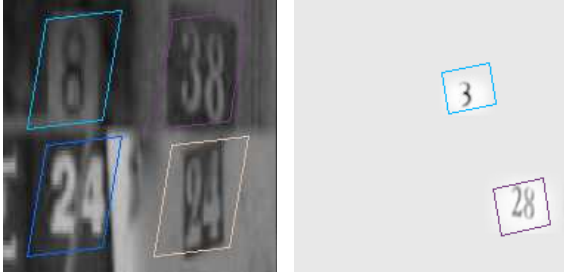


Figure 4. Samples from our generated datasets, including BBoxes predicted by our model. *Left*: Sample from regular grid dataset, *Right*: Sample from dataset with randomly positioned house numbers.

by us, we found that it is not possible to train a model from scratch, that can find and recognize more than two textlines that are scattered across the whole image. It is possible to train such a network by first training the model on easier tasks first (few textlines, textlines closer to the center of the image) and then increase the difficulty of the task gradually. In the supplementary material we provide short video clips that show how the network is exploring the image while learning to detect text for a range of different experiments.

### 4.3. Experiments on Robust Reading Datasets

In our next experiments we used datasets where text regions are already cropped from the input images. We wanted to see whether our text localization network can be used as an intelligent sliding window generator that adapts to irregularities of the text in the cropped text region. Therefore we trained our recognition model using CTC on a dataset of synthetic cropped word images, that we generated using our own data generator, that works similar to the data generator introduced by Jaderberg *et al.* [14]. In Table 2 we report the recognition results of our model on the ICDAR 2013 robust reading [19], the Street View Text (SVT) [32] and the IIIT5K [21] benchmark datasets. For evaluation on the ICDAR 2013 and SVT datasets, we filtered all images that contain non-alphanumeric characters and discarded all images that have less than 3 characters as done in [28, 32]. We obtained our final results by post-processing the predictions using the standard hunspell english (en-US) dictionary. Overall we find that our model achieves state-of-the-art performance for unconstrained recognition models on the ICDAR 2013 and IIIT5K dataset and competitive performance on the SVT dataset. In Figure 5 we show that our model learns to follow the slope of the individual text regions, proving that our model produces sliding windows in an intelligent way.

Method	ICDAR 2013	SVT	IIIT5K
Photo-OCR [1]	87.6	78.0	-
CharNet [18]	81.8	71.7	-
DictNet* [15]	<b>90.8</b>	80.7	-
CRNN [27]	86.7	80.8	78.2
RARE [28]	87.5	<b>81.9</b>	81.9
Ours	<b>90.3</b>	79.8	<b>86</b>

Table 2. Recognition accuracies on the ICDAR 2013, SVT and IIIT5K robust reading benchmarks. Here we only report results that do not use per image lexicons. (\*[15] is not lexicon-free in the strict sense as the outputs of the network itself are constrained to a 90k dictionary.)

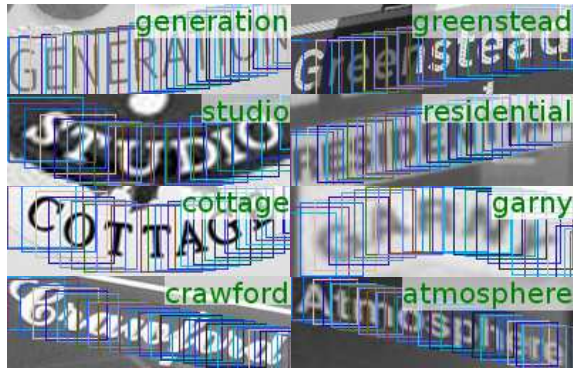


Figure 5. Samples from ICDAR, SVT and IIIT5K datasets that show how well our model finds text regions and is able to follow the slope of the words.

### 4.4. Preliminary 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.* [30] to be our third dataset to perform experiments on. The results we report here are preliminary and are only meant to show that our network architecture is also applicable to this kind of data, although it does not yet reach state-of-the-art results. The FSNS dataset contains images of french street name signs that 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 embedded in natural scenes with distracting backgrounds and (2) contains a lot of images that do not include the full name of the streets.

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 the labels of the original dataset do not include any hint on which words can be found in which text line. We therefore changed our approach and started with experiments where we tried to find individual words instead of textlines with



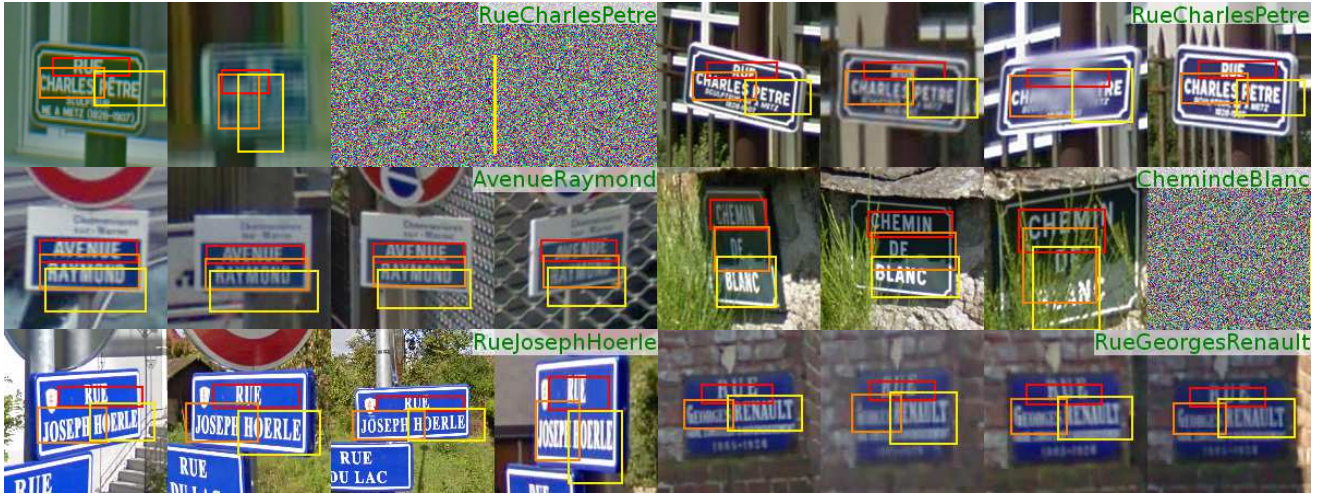


Figure 6. Samples from the FSNS dataset, these examples show that our system is able to detect a range of differently arranged text lines and also recognize the content of these words

more than one word. We adapted the groundtruth accordingly and used all images that contain a maximum of three words for our experiments, which leaves us with approximately 80 % of the original data from the dataset. Figure 6 shows some examples from the FSNS dataset where our model correctly localized the individual words and also correctly recognized the words. Using this approach we were able to achieve a reasonably good character recognition accuracy of 97 % on the test set, but only a word accuracy of 71.8%. The discrepancy in character recognition rate and word recognition rate is caused by the fact that the model we trained for this task uses independent softmax classifiers for each character in a word. Having a character recognition accuracy of 97 % means that there is a high probability that at least one classifier makes a mistake and thus increases the sequence error.

## 5. Conclusion

In this paper we presented a system that can be seen as a step towards solving end-to-end scene text recognition, using only 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 extensive pre-training on easier sub-tasks before the model can converge on the real task. We also showed that the same network architecture can be used to reach competitive or state-of-the-art results on a range of 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 textlines/characters that can be detected at once, 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

- [1] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Phototocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 785–792, 2013.
- [2] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv:1512.01274 [cs]*, 2015.
- [3] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.
- [4] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2963–2970, 2010.
- [5] L. Gomez-Bigorda and D. Karatzas. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *arXiv:1604.02619 [cs]*, 2016.
- [6] I. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *ICLR2014*, 2014.
- [7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Ma-*



- chine Learning, ICML '06, pages 369–376, New York, NY, USA, 2006. ACM.
- [8] A. Graves, N. Jaitly, and A. r. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 273–278, 2013.
  - [9] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.
  - [10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
  - [11] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang. Reading scene text in deep convolutional sequences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3501–3508. AAAI Press, 2016.
  - [12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
  - [13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 448–456, 2015.
  - [14] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, NIPS*, 2014.
  - [15] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2015.
  - [16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems 28*, pages 2017–2025. Curran Associates, Inc., 2015.
  - [17] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *Computer Vision - ECCV 2014*, number 8692 in Lecture Notes in Computer Science, pages 512–528. Springer International Publishing, 2014.
  - [18] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision - ECCV 2014*, number 8692 in Lecture Notes in Computer Science, pages 512–528. Springer International Publishing, 2014.
  - [19] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013.
  - [20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, 2015.
  - [21] A. Mishra, K. Alahari, and C. Jawahar. Scene text recognition using higher order language priors. In *BMVC 2012-23rd British Machine Vision Conference*, pages 127.1–127.11. British Machine Vision Association, 2012.
  - [22] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
  - [23] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.
  - [24] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In *Computer Vision - ACCV 2010*, number 6494 in Lecture Notes in Computer Science, pages 770–783. Springer Berlin Heidelberg, 2010.
  - [25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016.
  - [26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.
  - [27] B. Shi, X. Bai, and C. Yao. 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*, 2016.
  - [28] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4168–4176, 2016.
  - [29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
  - [30] R. Smith, C. Gu, D.-S. Lee, H. Hu, R. Unnikrishnan, J. Ibarz, S. Arnoud, and S. Lin. End-to-end interpretation of the french street name signs dataset. In *Computer Vision - ECCV 2016 Workshops*, pages 411–426. Springer, Cham, 2016.
  - [31] S. K. Sønderby, C. K. Sønderby, L. Maaløe, and O. Winther. Recurrent spatial transformer networks. *arXiv:1509.05329 [cs]*, 2015.
  - [32] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464, 2011.
  - [33] K. Wang and S. Belongie. Word spotting in the wild. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision - ECCV 2010*, number 6311 in Lecture Notes in Computer Science, pages 591–604. Springer Berlin Heidelberg, 2010.
  - [34] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4042–4049, 2014.