Development and Evaluation of a Text Recognition Framework using Synthetic Data

Daniel Steininger[1] and **Andreas Zweng**[1] and **Csaba Beleznai**[1] and **Thomas Netousek**[2]

Abstract. Text recognition is an intricate Computer Vision task. The main complexity arises from the fact that text as a character sequence spans a very large space of possible appearances, induced by combinatorially vast character orderings, diverse font styles, weights, colors and backgrounds. In order to encode a rich representation of variations and to generate an informative model by statistical learning, image data balanced along all dimensions of variations are needed. In this paper we present a synthetic text pattern generation framework and its use for localizing text lines and recognizing characters in individual frames of broadcast videos. The paper demonstrates that detection and recognition accuracies can be significantly enhanced by employing synthetic text image patches for training an Aggregated Channel Features (ACF) detector and a Convolutional Neural Network (CNN) character recognizer for the text recognition task. Moreover, an efficient annotation tool is presented for ground truth data generation from videos, enabling evaluation experiments on large-scale (several thousands of frames) video datasets. A quantitative evaluation of the detection functionality and qualitative experiments for character recognition are presented, exhibiting promising results on challenging (low-resolution, compression artifacts) real-world test data.

1 INTRODUCTION

End-to-end text recognition extracting textual information from digital images has been a key pattern recognition research topic for multiple decades. Recognition in constrained (high resolution and contrast, known scale) scenarios such as optical character recognition (OCR) in scanned documents has matured to practically relevant systems, while unconstrained, "in the wild" scenarios still represent a substantial challenge. Video text recognition in broadcast videos - representing the main focus of this paper - lies in-between in complexity terms, since overlay text content is typically free from geometric deformations, however still subject to many variations (style, color, background).

In this paper we present the use of synthetic data to generate rich statistical models for the text detection and character recognition tasks. The text detection and localization step generates a set of text line candidates (delineated by a bounding box) and associated text probabilities, whereas character recognition yields class label estimates - based on a pool of previously trained classes - for each of the characters forming a given text line.

In recent years it has been shown within the context of various visual object recognition tasks that vast amounts of artificial training data can be generated automatically resulting in an improvement of the classification accuracy [8], [14], [12]. By means of synthetic data, not only the ground truth is automatically generated along with the data, but also the training data can be adjusted to a targeted task. In particular, for video text recognition we often know properties such as the expected font type, the font color, the orientation and to some extent the background. This prior knowledge can thus directly be considered when generating the training data.

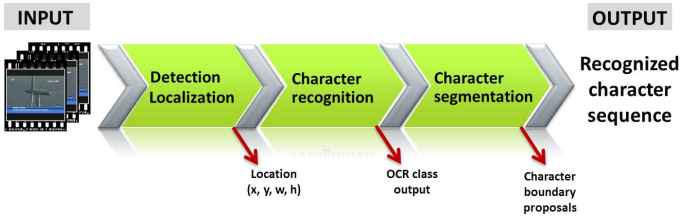

Figure 1. End-to-end processing chain of the text recognition framework.

The recent highly successful deep Convolutional Neural Network learning paradigm exhibits an exceptional generalization capability, but at the same time possesses a high descriptive complexity implying that in order to establish meaningful distributed representations at deeper layers of the network, it requires large amounts of training data. Since manually annotating word regions is highly time consuming, existing datasets are insufficient for reflecting the high variability of real data. Therefore, generating synthetic text data is an essential requirement. In our end-to-end text recognition chain (see Figure 1) we first employ the highly run-time efficient ACF detector [6] (detection and localization step), and afterwards CNN-based character recognition and segmentation stages. All stages are trained by large quantities of synthetic data.

The first promising application of Convolutional Neural Networks (CNNs) for recognizing handwritten numbers was shown in [11]. Although problem size and data variability were limited due to the available hardware at that time, the end-to-end pipeline for training and testing, as well as the compositional power of multi-layer neural networks fundamentally influenced many areas in machine learning in later years. Based on this precursor work, the results of [10] showed the potential of deep architectures in combination with the general-purpose computing capabilities of GPUs, which facilitated training on larger datasets and tackling more challenging tasks.

Recent approaches with classifiers based on CNNs have shown a significant impact on the accuracy of text recognition systems [3][2][12]. The absence of hand-crafted features, implicit learning of prior knowledge and deployment on more powerful hardware make them the most promising approach for Optical Character Recognition. The approach in [4] tunes the classification accuracy with

[1] AIT Austrian Institute of Technology GmbH, Vienna, Austria, email: daniel.steininger.fl@ait.ac.at
[2] eMedia Monitor GmbH, Vienna, Austria

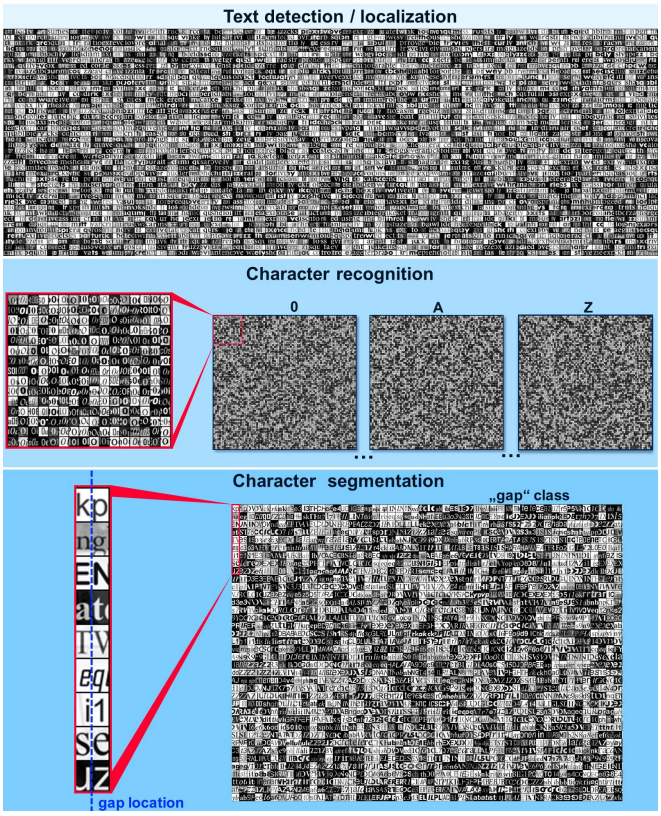

Figure 2. Illustration providing an overview on the characteristics of generated synthetic datasets. Top: for text detection and localization patches of character triplets are created. Center: individual characters are trained on synthetic single-character instances. Bottom: to detect gaps between characters we use an additional ”gap” class, which consists of two-character patch instances centered on their gap location.

synthetic data specialized on specific font categories, whereas [8] presents a sophisticated framework for synthetic text generation, which can easily be adapted to other languages without any human labeling effort.

The paper is organized as follows: Section 2 describes the overall employed methodology, the synthetic data generation scheme and the individual algorithmic steps to train, test and evaluate the text detection and recognition stages. Section 3 presents and discusses the obtained results for text localization and recognition. Finally the paper is concluded in Section 4.

2 METHODOLOGY

In the following section we describe a text synthesis tool and its use in the context of training text detection and character recognition. Furthermore, we present an annotation tool enabling key-frame based manual annotation in broadcast videos, and opening ways for large-scale evaluation of video text detection and recognition.

2.1 General overview

Establishing large datasets for the development of a video text recognition system is essential for two reasons: (i) **Learning representations of text appearance**: as with other complex vision systems, the need for good generalization capability (modeling quality for previously unseen data) is crucial, since high recall is substantial for

Figure 3. Reference lines governing a text line's geometric properties. The right side of the image shows artificial text image patches aligned to these reference lines during the synthesis step.

covering all essential broadcast video content. In order to capture the visual appearance of vast amounts of text, we built a text image synthesis tool capable to generate unlimited amounts of training data well matching the appearance of overlay text in videos. (ii) **Large-scale characterization and evaluation**: desired improvements in the recognition accuracy call for datasets which represent a broad range of variation of the class to be recognized (for text: font styles, weight, size, spacing, background) and of the class which should be ignored (background, also containing difficult patterns such as repetitive textures, clutter and high-frequency noise). In order to accomplish this task (i) we built a video annotation tool targeting large scale annotation and (ii) constructed an evaluation framework following established (ICDAR [9]) standards in the text recognition community.

2.2 Synthetic data generation

We elaborated a text image synthesis tool in Matlab which is capable to create within a normalized geometric reference frame defined by predefined baseline, ascender and descender lines (see Figure 3) patches of local text patterns, also including artificially-induced variations such as font type and weight, text background, spatial offset, scaling variations and deformations. The synthesis tool employs prior on the bigram (adjacent characters) frequency of the English language, thus it tries to recreate not random but plausible font adjacencies. Generated text samples are shown in Figure 2 for the different tasks addressed in this paper.

The synthesis code employs the Matlab native *listfonts* command, which retrieves all available system fonts. These fonts can be used to render a text string by the *text* command, and using a screen capture script from the MathWorks FileExchange [1] we convert the rendered text into an image. This image is subsequently cropped around the desired set of characters while using the predefined text reference lines (Figure 3) for geometric normalization. Moreover, additional geometric transforms are applicable: spatial offsets along the x- and y-directions and a rotation within a predefined angular range. An additional image containing no text can be used as a background overlay to introduce some structure and texture behind the text characters. This structured background targets enhanced invariance of text detection and recognition in presence of clutter, while geometric transformations are performed to increase invariance with respect to deviations from an ideal character position and pose. The introduced perturbations typically result in an increased recall rate, which is necessary to achieve if all relevant text lines need to be found and correctly read.

2.3 Synthetic data aided text detection

2.3.1 Training and testing

We generated 35000 text patch samples (a subset is shown in Figure 2 top) which we used to train the ACF (Aggregate Channel Features) Detector [5],[6] which we have ported to C++. The synthetically trained detector surpasses the one trained on 5000 manually cropped

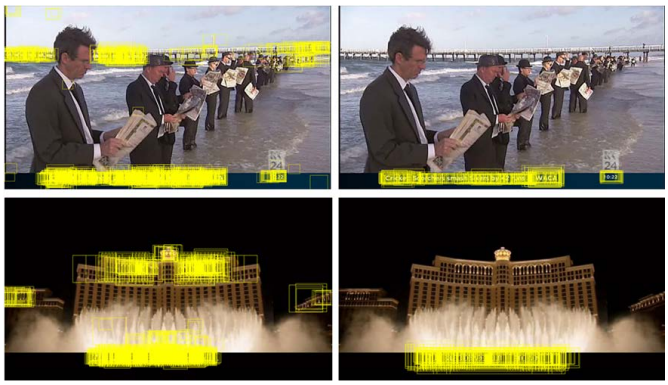

Figure 4. Two difficult (repetitive structures) images demonstrating the classifier accuracy improvement accomplished by using a large synthetic dataset. Yellow rectangles show raw output of the ACF detector trained using real world (a) and synthetic (b) data.

text patches from videos, mostly due to the facts that (i) more data with greater variation is employed, and (ii) nuisance factors such as compression artifacts, color bleeding and low resolution effects are absent. Qualitative and quantitative (precision-recall, DetEval [13]) evaluation of these detectors has been performed and results show a significant improvement over the detector trained with manually cropped real world data.

The ACF detector employs features (intensity channel, gradient magnitude and orientations) - the so-called *channel features* - extracted at multiple image resolutions. A specific advantage of the ACF multi-resolution feature computation is that not all resolution levels have to be computed, but features at certain scale levels can be directly extrapolated from features of nearby scales, at no significant expense of the detection accuracy. This feature approximation trick yields an overall detection framework of excellent run-time performance versus recognition accuracy ratio.

Figure 5 illustrates the obtained accuracy improvement by text detection by a quantitative measure. As it can be seen from the ROC (Receiver Operating Characteristic) curves obtained for manually annotated (*real*) and synthetic samples (*synth*) training data, the use of synthetic data improves the detection rate (true positive rate) by about 5 percent at a false positive rate of 0.1. A detailed specification of this evaluation experiment can be found in the Results section.

2.4 Synthetic data aided text recognition (OCR)

Synthetic data for optical character recognition (OCR) enables the possibility to enhance the recognition rate by increasing the amount of training data when needed. In this paper, training a model using synthetic data is done using Convolutional Neural Networks and recognition is done using a sliding window approach within the previously found bounding box from text recognition. The following sections describe our approaches for training and recognition of text in images.

2.4.1 Training

Training a model using Convolutional Neural Networks issues the question of how much training data will be used and how the training data is distributed among the object classes. Even though some characters have a higher probability of appearance then others, but for training, each class should have the same amount of samples, in

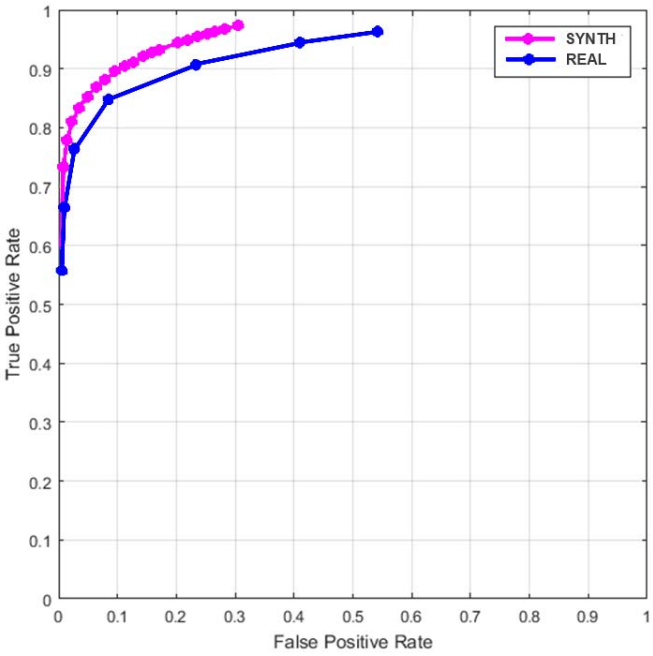

Figure 5. Text detection performance improvement using large synthetic dataset (SYNTH = 35000 samples) vs. the a classifier (REAL), trained on 5000 manually cropped real image sample.

order to have a balanced training set. In this paper, we used a set of 6000 samples per class for the training stage where each character has a separate class for upper-case and lower-case, resulting in 62 classes (52 classes for characters and 10 digits) and therefore 372.000 input images. The network layer architecture is the same as LeNet-5 in [11] which is shown in Figure 6.

For training the adagrad adaptive learning rate method was used, which was originally proposed by Duchi et.al. [7]. The process of training our model consisted of several difficulties such as different font types, background variability, font weight (bold, italic or normal) and inverted colors for half of the training set (black on white and white on black) and took around 5 hours with an evaluation set of 1000 samples for each character and accomplished a recognition rate of 95 percent on an independent test set.

2.4.2 Recognition

The recognition task operates on the output of the text localization. The detected regions are used for further investigation by sliding through the region and evaluating each position for possible characters using the trained model. Figure 11 shows the confidence responses at each sliding window position for each character and digit (from top to bottom: 0 to 9, a to Z). The highest responses are used to compose the text.

The resulting confidence map consists of confidences (values between 0 and 1) at each position of the sliding window. In order to segment the text into characters, we find local maxima in the confidence map and use the corresponding character to compose the text. Often it is the case, that a local maximum is not centered at the character position, which makes it difficult to find the correct positions of the characters. To overcome this problem, we train a second model which is intended to find gaps between characters. Therefore a CNN is trained with 20000 samples of character gaps (gap-class) and 20000 samples of characters (character-class), evenly distributed

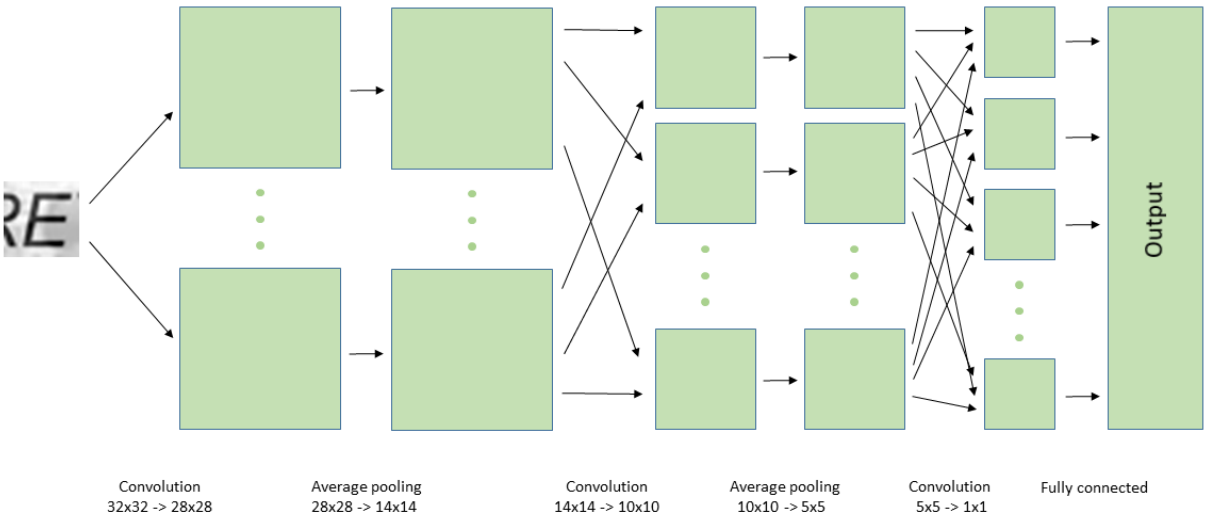

Figure 6. Architecture of the Convolutional Neural Network for Optical Character Recognition.

between all characters and digits from our text recognition model. Since gaps contain significant edges on the left and right side of the image center which are caused by the character borders of any character, and the sliding window outputs show promising results (see Figure 7), the use of this second CNN in our approach is well suited for character segmentation.

Figure 7. Color-coded confidence distribution (bright = high confidence) for recognizing gaps between characters by learning-based character separation

2.5 Annotation and evaluation

We built an easy-to-use software tool (in Matlab, but also compiled to a binary executable), which can perform key-frame based manual annotation by assigning bounding box representations to text lines in discrete video frames (see Figure 10). Due to the key-frame concept these annotations are propagated across time and can be updated and terminated at any time instance of the video. In this way also animated text lines can be annotated. The framework is able to generate specific, from videos derived datasets according to the evaluation criteria: varying spatial and temporal resolutions and predefined annotation data schemes (*txt*, *xml*, *yaml*).

We have annotated multiple hours of broadcast videos downloaded from YouTube. As Figure 8 displays the annotated set contains 5 different English-language TV channels of various lengths, altogether

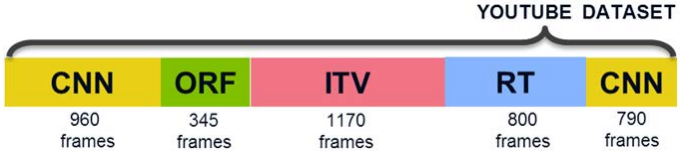

Figure 8. The composition of our generated *YouTube* dataset containing 4065 annotated frames in total, originating from five different broadcast programs.

consisting of 4065 annotated frames. Annotation contains bounding box coordinates (relating to single text lines) and ground truth ASCII text for each bounding box. The dataset contains many complex scenes where clutter, high-frequency and repetitive patterns occur. The annotated dataset forms our evaluation dataset where experiments for assessing the text detection accuracy have been performed.

Conventional bounding box overlap based evaluation measures exhibit a deficiency in the definition of matching between detected and ground truth (GT) bounding boxes (BB). Namely, in most cases there is no one-to-one correspondence, but several detected BBs correspond to a single GT-BB, or a single detected BB matches multiple GT-BBs. The first case is denoted as *splitting*, the latter as *merging*. See these situations depicted in Figure 9. These ambiguous matching situations render overlap-based evaluation results ambiguous: 50 percent overlap to GT might imply that either half of the GT-BBs were detected, or half of the area of a single GT-BB was detected.

We adopted the ICDAR 2015 evaluation scheme [9] relying on the *DetEval* framework [13]. The *DetEval* evaluation measures take the various bounding box correspondence types into account and derive BB-level precision and recall values based on a parameter-tunable GT-to-BB association scheme. Using a fixed association tuning parameter we can generate an ROC curve based characterization (such

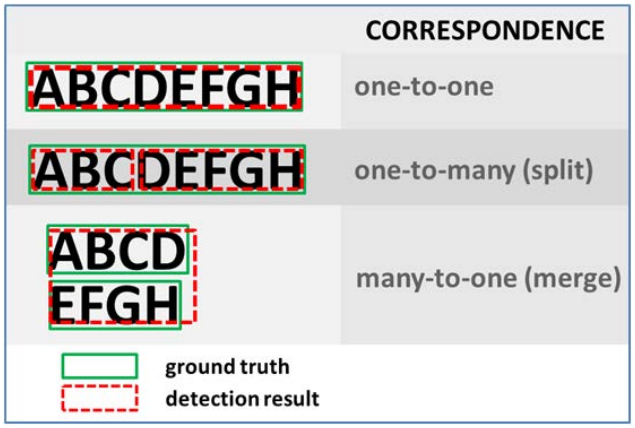

Figure 9. Different correspondence types between ground truth (green) and detection results (red dashed) bounding boxes.

Figure 10. Screenshot of the video annotation tool, capable to quickly annotate large amounts of video text by setting text bounding boxes for selected key-frames and interpolating this information between key-frames.

as in Figure 5), whereas by allowing a variable association parameter, characteristic *DetEval* performance graphs can be computed (not shown).

3 RESULTS

In order to perform the evaluation of the presented text detection stages, following steps were carried out.

Text detection and localization: The 4065 annotated images were used to evaluate the ACF detector [5], in one case trained on 5000 manually annotated text data (denoted as the *REAL* detector), in the second case on 35000 fully synthetic data (denoted as the *SYNTH* detector). The *DetEval* evaluation criterion was used with a fixed BB-to-BB (BB - bounding box) association parameter, while varying the classifier sensitivity. A qualitative comparison is displayed in Figure 4 and the resulting ROC curve is shown in 5. As the plot illustrates, synthetic data brings a significant improvement in terms of the detection rate (True Positive Rate) at a given false alarm rate (False Positive Rate). Training the ACF detector with even more synthetic data (50000 samples) did not improve results, indicating that (i) the data does not introduce additional appearance or structural information about the modeled class and (ii) the representation capability of the employed features and learning strategy is limited. Due to these limitations, repetitive structures and clutter still represent a problem (seen by the non-vanishing amount of False Positives), but synthetic data improves on the rate of false alarms. The ACF detector is fast, detecting and localizing text with about 7 frame-per-seconds in an image frame with a resolution of 1024×768 *pixels* on a modern PC. A subsequent more stringent and - at the same time computationally more demanding - analysis steps, such as an OCR step, is able to reduce the amount of False Positives further.

Text recognition: During optical character recognition, a text line detected in the previous localization step is employed as input. The proposed sliding window approach performs the classification of the local analysis window content and assigns a label to it, matching one of the 62 character classes or the gap class. Due to the tightly coupled spatial analysis and fine-grained classification, a separate spatial character segmentation step is avoided. Typically, the character segmentation step (in terms of foreground-background segmentation) is the most sensitive stage in an end-to-end text recognition framework, where small resolution and degraded character appearance are probable to lead to segmentation failures. Classification results are shown in Figure 11, where individual class-specific confidences are shown as dark peaks (confidence heat map is inverted for better visibility) in the individual rows of character classes. The bottom part of the figure displays the maximum-confidence classifier response at character locations, thus forming the OCR output. As it can be seen, certain recognition errors still occur: the lower case character ”o” is confused with the upper case character ”Q”. The reason for this problem is the presence of upper case and lower case letters as well as letters with a descender within a single recognition region (text row), because the sliding window is selected by the height of the highest character in this region and the bottom-most point is selected by the lowest descender of all characters in the region. Given theses conditions and the fact that each training image is geometrically normalized with respect to its containing character (with border pixels) during training, the appearance can vastly vary between the training images and a given recognition window. At the current state of our recognition step, this geometric scaling discrepancy is not taken into consideration. However, the synthetic data generation tool can be easily ex-

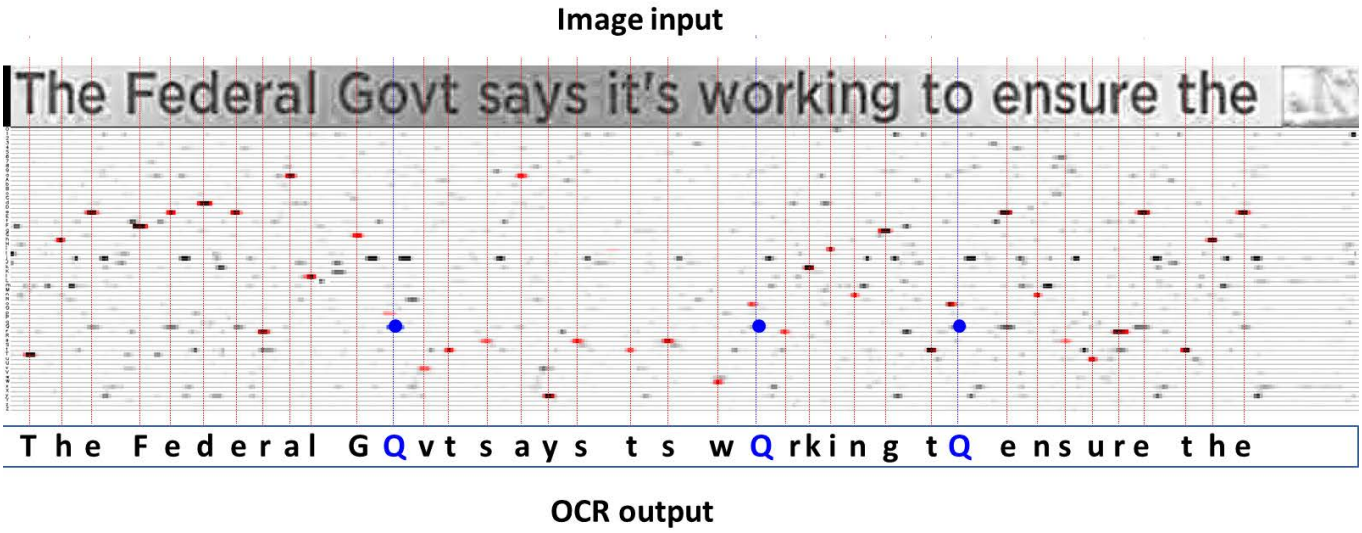

Figure 11. Recognition results shown in detail for a given text line. Individual character classes shown as separate rows (see the small class labels at the left border). Individual dark peaks in respective rows indicate a high confidence in a given class. Peaks highlighted in red display the maximum classifier response at a given character center, resulting in the corresponding OCR output at the bottom of the figure.

tended to also include samples with such scaling variations, while the Convolutional Neural Network is capable to accommodate these additional variations. This extra representational effort will probably lead to a greatly enhanced recognition performance.

4 CONCLUSION

In this paper we present an applied example of using large amounts of synthetic data for generating representative statistical models for the text detection and character recognition tasks. Furthermore, we show that annotating and synthesizing text is not an overly complicated task, and simple software tools can generate vast amounts of data with broad appearance characteristics. The rich variability encompassed by the synthetic data yields improved accuracy for the text detection task, and it enables character recognition without an explicit segmentation step.

Future work will mainly involve the improvement of the sliding window based recognition approach. Since Convolutional Neural Networks provide a large capacity to accommodate appearance, scale and other variations for a large number of classes, we plan to enrich the training set with even more variations. We plan to train each character in different text configurations such as variable padding thus improving the classifier's invariance with respect to position and geometric scaling. This is highly relevant for the typical case when a text line consists of geometrically varying characters, such as uppercase, lower-case and characters with ascender and descenders.

ACKNOWLEDGEMENTS

The work was partially supported by the Vision+ project under the COMET program of the Austrian Research Promotion Agency (FFG) and the research initiative 'Intelligent Vision Austria' with funding from the Austrian Federal Ministry of Science, Research and Economy and the Austrian Institute of Technology.

REFERENCES

[1] Mathworks fileexchange. https://de.mathworks.com/matlabcentral/fileexchange/?s_tid=gn_mlc_fx. Accessed: 2016-09-20.

[2] Alessandro Bissacco, Mark Cummins, Yuval Netzer, and Hartmut Neven, 'Photoocr: Reading text in uncontrolled conditions', in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 785–792, (2013).

[3] Adam Coates, Blake Carpenter, Carl Case, Sanjeev Satheesh, Bipin Suresh, Tao Wang, David J Wu, and Andrew Y Ng, 'Text detection and character recognition in scene images with unsupervised feature learning', in *2011 International Conference on Document Analysis and Recognition*, pp. 440–445. IEEE, (2011).

[4] Teófilo Emídio de Campos, Bodla Rakesh Babu, and Manik Varma, 'Character recognition in natural images.', in *VISAPP (2)*, pp. 273–280, (2009).

[5] Piotr Dollar, Ron Appel, Serge Belongie, and Pietro Perona, 'Fast feature pyramids for object detection', *IEEE Trans. Pattern Anal. Mach. Intell.*, **36**(8), 1532–1545, (August 2014).

[6] Piotr Dollár, Serge Belongie, and Pietro Perona, 'The fastest pedestrian detector in the west', in *BMVC*, (2010).

[7] John Duchi, Elad Hazan, and Yoram Singer, 'Adaptive subgradient methods for online learning and stochastic optimization', *J. Mach. Learn. Res.*, **12**, 2121–2159, (July 2011).

[8] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 'Reading text in the wild with convolutional neural networks', *International Journal of Computer Vision*, **116**(1), 1–20, (2016).

[9] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman K. Ghosh, Andrew D. Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny, 'Icdar 2015 competition on robust reading.', in *ICDAR*, pp. 1156–1160. IEEE Computer Society, (2015). relocated from Tunis, Tunisia.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, 'Imagenet classification with deep convolutional neural networks', in *Advances in neural information processing systems*, pp. 1097–1105, (2012).

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, **86**(11), 2278–2324, (1998).

[12] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng, 'End-to-end text recognition with convolutional neural networks', in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 3304–3308. IEEE, (2012).

[13] Christian Wolf and Jean-Michel Jolion, ' Object count/Area Graphs for the Evaluation of Object Detection and Segmentation Algorithms', *International Journal of Document Analysis and Recognition*, **8**(4), 280–296, (April 2006).

[14] Gkhan Yildirim, Radhakrishna Achanta, and Sabine Ssstrunk, 'Text recognition in natural images using multiclass hough forests', in *In Proc. VISAPP*, (2013).