Digit String Recognition

Lucas Steinmann, Leandro Piekarski, Hesham Hendy, Manuel Olk

Praktikum Neuronale Netze
Karlsruhe Institute of Technology, Germany

Abstract

This is a layout specification and template definition for the paper of the IWSLT 2015 Conference. The format is essentially the one used for the IEEE ICASSP conferences.

1. Introduction

In recent years the recognition of handwritten digits has been systematically improved. Benchmarked on the MNIST (Mixed National Institute of Standards and Technology) dataset that contains about 70.000 images of handwritten digits along with their corresponding labels, numerous papers were published that apply convolutional neural networks (CNNs) to achieve error rates of 0.23% [1] or even 0.21% [2].

As these results almost cope with human performance and a good dataset does need to represent a sufficiently challenging problem to stay useful and to ensure its longevity, various adjustments can be contemplated. One way could be the extension of the underlying dataset [3].

Another approach would be the extension of the task itself by not only recognizing single digits but whole digit strings. In 2014 the ICFHR (International Conference on Frontiers in Handwriting Recognition) announced a competition to attend that matter: Handwritten Digit String Recognition in Challenging Datasets (HDSRC2014) [4].

In course of that they proposed a new benchmark framework that consists of two real world datasets and respective evaluation measures. The task's complexity is not only founded on the connected digits but also on the additional challenges that come along with the images' real world nature as document layout, background texture or noisy strokes.

Traditional approaches to digit string recognition often used segmentation. The string image is segmented to pieces that in best case represent single digits. The recognition results of these single pieces are then combined to get global optimal results. Since these approaches in practice suffer from various handwritten styles, connected or even overlapped characters and noises, [5] proposes an segmentation free approach.

Their model uses ResNet with convolutional layers [6] as an discriminative sequence extractor and feature decoder, combines it with an bidirectional LSTM and calculates the loss with connectionist temporal classificiation (CTC) [7].

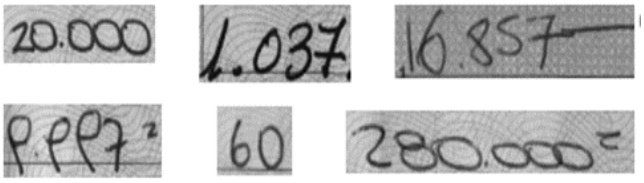

Figure 1: Sample images of ORAND-CAR-A

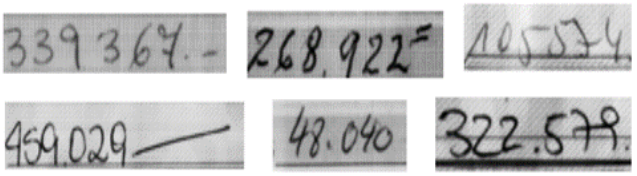

Figure 2: Sample images of ORAND-CAR-B

Since this approach achieved remarkable results on the given datasets we used it as fundament for our own model.

The rest of this paper is organized as follows. In Section 2 we briefly describe the real world datasets given by the HDSRC2014. ..... Then, details of our experiments are presented in Section 3. Section 4 concludes this paper and discusses potential future work.

2. Databases

There are two different databases for recognizing handwritten digit strings we used in our experiments.

The first database, named ORAND-CAR, is a real world database with 11719 images of the 'Courtesy Amount Recognition (CAR)' field of bank checks. It originates from two different sources, one bank of Chile and one of Uruguay, and therefore shows different characteristics in terms of check layout, image quality and further noise. That is the reason the database is splitted into two subsets called ORAND-CAR-A (CAR-A) and ORAND-CAR-B (CAR-B). Samples of both datasets are shown in Figure 1 and 2.

CAR-A's training set consists of 2009 images whereas the testing set delivers 3784 images. In CAR-B there are 3000 training images and 2926 images for testing.

The second database we worked on is called Computer Vision Lab Handwritten Digit String (CVL HDS). It con-

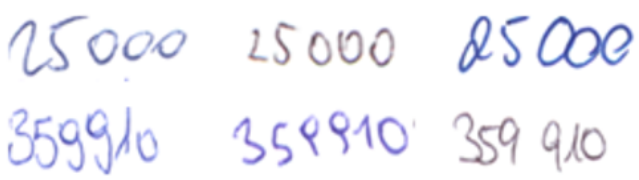

Figure 3: Sample images of CVL HDS

Table 1: *Summary of string length distribution.*

	Training			Testing		
len	CAR-A	CAR-B	CVL	CAR-A	CAR-B	CVL
2	36	0	0	22	0	0
3	387	5	0	204	0	0
4	1425	69	0	704	63	0
5	1475	1241	789	903	1200	125
6	363	1452	4144	145	1599	758
7	87	157	1765	29	137	379
8	11	2	0	2	1	0

tains handwritten digits of 300 different writers without any background noise. The training set includes only 10 different digit strings from about 125 writers which leads to overall 1262 images for training. In the testing set we have the same 10 digit strings from the remaining writers and 16 new strings from all 300 writers to reach 6698 testing images. CVL HDS provides us with large variability with respect to handwriting styles but lacks diversity in the digit strings themselves. Examples of the CVL HDS database are shown in Figure 3.

The datasets differ with respect to their string lengths. CVL has images with string lengths from 5 to 7 whereas CAR-B offers digit strings from length 4 to 8. In this regard CAR-A comes with the biggest variety and covers a string length range from 2 to 8 digits. A summary of the string length distribution can be seen in Table 1.

Since the images of the used databases come in lots of different aspect ratios and we want them to fit our model, we resize them to a fixed width of 120 and an height of 50.

We furthermore make use of the different tools given by Pytorch to augment our data. Dynamic data augmentation is achieved with help of the functions ColorJitter() and RandomAffine(). How the data is transformed, can be seen in Figure 4.

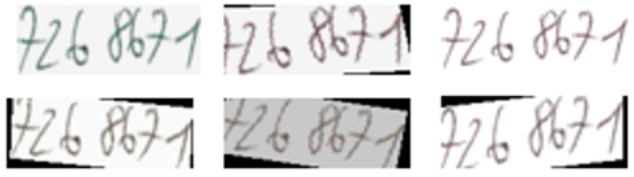

Figure 4: Examples of an image after preprocessing and data augmentation.

3. Page layout and style

Authors should should observe the following rules for page layout. A highly recommended way to meet these requirements is to use a predefined template and check details against the corresponding example file.

3.1. First page

The first page should have the paper title, author(s), and affiliation(s) centered on the page across both columns. The remainder of the text must be in the two-column format, staying within the indicated image area.

3.1.1. Paper Title

The paper title must be in boldface. All non-function words must be capitalized, and all other words in the title must be lower case. The paper title is centered across the top of the two columns on the first page as indicated above.

3.1.2. Authors' Name(s)

The authors' name(s) and affiliation(s) appear centered below the paper title. If space permits, include a mailing address here. The templates indicate the area where the title and author information should go. These items need not be confined to the number of lines indicated; papers with multiple authors and affiliations may require two or more lines. Note that the submission version of technical papers *should be anonymized for review*.

3.1.3. Abstract

Each paper must contain an abstract that appears at the beginning of the paper.

3.2. Basic layout features

- Proceedings will be printed in A4 format. The layout is designed so that files, when printed in US Letter format, include all material but margins are not symmetric. Although this is not an absolute requirement, if at all possible, **PLEASE TRY TO MAKE YOUR SUBMISSION IN A4 FORMAT.**
- Two columns are used except for the title part and possibly for large figures that need a full page width.
- Left margin is 20 mm.
- Column width is 80 mm.
- Spacing between columns is 10 mm.
- Top margin 25 mm (except first page 30 mm to title top).
- Text height (without headers and footers) is maximum 235 mm.

Table 2: *This is an example of a table.*

ratio	decibels
1/1	0
2/1	≈ 6
3.16	10
10/1	20
1/10	-20

- Headers and footers must be left empty (they will be added for printing).

- Check indentations and spacings by comparing to this example file (in pdf format).

3.2.1. Headings

Section headings are centered in boldface with the first word capitalized and the rest of the heading in lower case. Subheadings appear like major headings, except they start at the left margin in the column. Sub-sub-headings appear like subheadings, except they are in italics and not boldface. See the examples given in this file. No more than 3 levels of headings should be used.

3.3. Text font

Times or Times Roman font is used for the main text. Recommended font size is 9 points which is also the minimum allowed size. Other font types may be used if needed for special purposes. While making the final PostScript file, remember to include all fonts!

LaTeX users: DO NOT USE Computer Modern FONT FOR TEXT (Times is specified in the style file). If possible, make the final document using POSTSCRIPT FONTS. This is necessary given that, for example, equations with non-ps Computer Modern are very hard to read on screen.

3.4. Figures

All figures must be centered on the column (or page, if the figure spans both columns). Figure captions should follow each figure and have the format given in Fig. 5.

Figures should preferably be line drawings. If they contain gray levels or colors, they should be checked to print well on a high-quality non-color laser printer.

3.5. Tables

An example of a table is shown as Table 2. Somewhat different styles are allowed according to the type and purpose of the table. The caption text may be above or below the table.

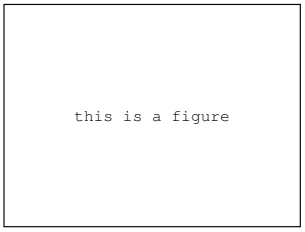

Figure 5: *Schematic diagram of speech production.*

3.6. Equations

Equations should be placed on separate lines and numbered. Examples of equations are given below. Particularly,

$$x(t) = s(f_\omega(t)) \tag{1}$$

where $f_\omega(t)$ is a special warping function

$$f_\omega(t) = \frac{1}{2\pi j} \oint_C \frac{\nu^{-1k} d\nu}{(1 - \beta\nu^{-1})(\nu^{-1} - \beta)} \tag{2}$$

A residue theorem states that

$$\oint_C F(z)dz = 2\pi j \sum_k Res[F(z), p_k] \tag{3}$$

Applying (3) to (1), it is straightforward to see that

$$1 + 1 = \pi \tag{4}$$

Make sure to use `\eqref` when refering to equation numbers. Finally we have proven the secret theorem of all speech sciences (see equation (3) above). No more math is needed to show how useful the result is!

3.7. Hyperlinks

Hyperlinks can be included in your paper. Moreover, be aware that the paper submission procedure includes the option of specifying a hyperlink for additional information. This hyperlink will be included in the CD-ROM. Particularly pay attention to the possibility, from this single hyperlink, to have further links to information such as other related documents, sound or multimedia.

If you choose to use active hyperlinks in your paper, please make sure that they present no problems in printing to paper.

3.8. Page numbering

Final page numbers will be added later to the document electronically. *Please don't make any headers or footers!.*

3.9. References

The reference format is the standard for IEEE publications. References should be numbered in order of appearance, for example [**?**], [**?**], and [**?**].

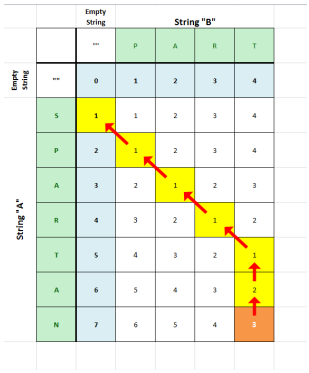

Figure 6: *Example of edit distance for words "part" and "spartan" by [8].*

4. Experiments

Please make sure to give all the necessary details regarding your experimental setting so as to ensure that your results could be reproduced by other teams.

5. Evaluation

5.1. Metrics

To evaluate the performance of the models two metrics are utilized. The first one is the Levenshtein distance, also known as edit distance. The edit distance as depicted by Fig. 6 calculates how many changes among insertions, deletions and substitutions are necessary to transform one string into another [8]. In this work all changes are weighted the same. The second metric is accuracy, which considers a prediction correct only if all characters match the label characters in the correct position. Originally [4] considers from TOP-1 to TOP-3 submissions in case of accuracy, however this work considers only the TOP-1 network output. For the final results of the trained models in this section only accuracy is displayed since it is a harder metric and the only one which can be compared with the baseline results.

5.2. Model performance

The implemented model is similar to the one proposed by Zhan et al. [5], therefore its results are used as a reference of what can be achieved with an architecture using CNN + RNN-CTC. For each of the provided datasets, the model is trained for 100 epochs with Adam as optimizer, learning rate of 1×10^{-4} and batch size equal to 16. The provided sets are already separated into training and test data, thus the training data was split further into 80% of it for training and 20% for validation. Table 3 summarizes the comparison between the base line and the used model. In each case of this comparison models were trained and tested only with data of the referenced dataset.

Given the fact that the model is able to perform well on CAR-A and CAR-B, experiments with cross-validation with data from different sets are made to evaluate if the reason for poor performance on CVL is the architecture or the model. Table 4 contains the results for models which were trained with datasets on the first line and tested against datasets on the first column. Considering only CAR-A and CAR-B, the best results are obtained by training and testing with data within a same set. The more interesting results are for CVL tests, as it has the worst performance when using its own training set and receives big performance improvements achieving almost 60% if tested on a model which was trained exclusively with CAR-B data.

A further idea to improve the CVL performance follows the principle "more data is always better". As observed in the previous experiments as well as for reasons discussed in the data analysis in section 1, the CVL test data is not represented well by its training data, especially regarding the small amount of different string labels. The idea is therefore to adopt the whole training data of CAR-A plus CAR-B to train the model from scratch and additionally fine tune it on CVL. Taken into account from the previous experiments that the architecture is able to generalize well and to maximize the amount of training data, no validation data is used in these experiments. Apart from the lacking train-validation split, all the parameters remain the same. For fine tuning, all weights are trained for extra 20 epochs exclusively on CVL.

The model trained on CAR-A+CAR-B without CVL data surpasses both the performances of models trained exclusively on these sets as well as the previously best CVL performance. By fine tuning the model on CVL however, the CVL test performance achieved almost 76%, which compared to the results in [4] means a second best CVL performance losing only to BeiJing et al. [4].

5.3. Error cases

In order to have a better understanding of the mistakes the network makes, some of the errors during test are collected. Most of the errors fall into four classes. The first class is represented by Fig. 7a, in which the left border is misclassified as the digit "1", resulting in a prediction "175" versus label

Table 3: *Overall comparison between implementations.*

Methods	CAR-A	CAR-B	CVL
Zhan el al. 2017	0.8975	0.9114	0.2707
Our "interpretation"	0.9014	0.9084	0.2550

Table 4: *Performance on different sets.*

Tested \ Trained	CAR-A	CAR-B	CVL
CAR-A	**0.9014**	0.7315	0.002
CAR-B	0.8434	**0.9084**	0.006
CVL	0.3019	**0.5804**	0.2550

Table 5: *Performance using more than one set + fine tuning.*

Tested \ Trained	CAR-A + CAR-B	CVL (fine tuning)
CAR-A	**0.9127**	0.0565
CAR-B	**0.9408**	0.4733
CVL	0.5913	**0.7579**

(a) Border (b) Bad digits (c) Symbols (d) Receptive field

Figure 7: *Examples of test errors*

”75”. Right borders can also be misclassified. A second class is as depicted by Fig. 7b, in which the digits cannot be easily classified correctly even by humans. In this case the network predicts ”400” for a label ”500”. The third class of identified errors is illustrated by Fig. 7c. Due to the fact that the pictures do not include exclusively digits, symbols such as commas can be confused with the digit ”1”, despite the training to classify all strange symbols as ”blank”. The last error class is pictured by Fig. 7d. Given the CNN output, the input picture is divided in 15 parts to be classified by the RNN. A problem also occurs thereby if irrelevant symbols occupy too much space in a way that a single digit is too small inside one of these parts. For the example the prediction was ”381”, which suggests that the digits ”29” are merged during the feature extraction.

Most of these errors can be probably avoided by investing more time into preprocessing the images, which is neglected in this work in favor of an end-to-end approach. Any extra preprocessing for such images requires feature engineering to remove irrelevant symbols by for example, searching for contours which are more likely to belong to digits as in Fig. 8. As a consequence the deep learning model would require less effort and complexity.

6. Lessons learned

Of the many lessons learned during the development of this work, the most important is to recognize when overfitting is happening. Divergences between training and validation performance tend to happen but if they are too big a thoroughly analysis of the possible causes is necessary. Data augmentation and regularization methods do help but are not enough to overcome highly divergent performances, which are more likely caused by a wrong approach. Models for more complex tasks have so many parameters that is easy for them to memorize the training set completely without any generalization capability given enough epochs.

A second lesson is to analyze carefully which type of information is passed to the LSTM input gates. Just reshaping tensors output by the CNN to fit the LSTM input shape without any reasonable criteria does not deliver any valuable results. At first it is not clear what to use as input in the presented problem since the sequence aspect is not instantly recognized as in problems which have for example image sequences as input or text sentences. Following the intuition of splitting the classification along the image width dimension, the LSTM time steps must also match this idea for learning effectively.

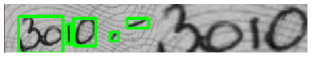

Figure 8: *Digit contour detection.*

7. Conclusions

This paper has described a novel approach for doing wonderful stuff such as ...

8. Acknowledgements

The IWSLT 2015 organizing committee would like to thank the organizing committees of INTERSPEECH 2004 for their help and for kindly providing the template files.

9. References

[1] D. C. Ciresan, U. Meier, and J. Schmidhuber, “Multi-column deep neural networks for image classification,” *CoRR*, vol. abs/1202.2745, 2012. [Online]. Available: http://arxiv.org/abs/1202.2745

[2] V. Romanuke, “Training Data Expansion and Boosting of Convolutional Neural Networks for Reducing the MNIST Dataset Error Rate,” accessed: 2019-02-19.

[3] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, “EMNIST: an extension of MNIST to handwritten letters,” *CoRR*, vol. abs/1702.05373, 2017. [Online]. Available: http://arxiv.org/abs/1702.05373

[4] “ICFHR 2014 Competition on Handwritten Digit String Recognition in Challenging Datasets,” http://www.orand.cl/en/icfhr2014-hdsr/#datasets, accessed: 2019-02-19.

[5] H. Zhan, Q. Wang, and Y. Lu, “Handwritten digit string recognition by combination of residual network and RNN-CTC,” *CoRR*, vol. abs/1710.03112, 2017. [Online]. Available: http://arxiv.org/abs/1710.03112

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unseg-

mented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 369–376. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143891

[8] "Edit Distance Algorithms," https://www.clear.rice.edu/comp130/12spring/editdist/, accessed: 2019-02-19.