Ethiopic Natural Scene Text Recognition Using Deep Learning Approaches

Direselign Addis(✉), Chuan-Ming Liu, and Van-Dai Ta

Department of Computer Science and Information Engineering,
National Taipei University of Technology, Taipei, Taiwan
{t106999405,cmliu,t104999002}@ntut.edu.tw

Abstract. The success of deep learning approaches for scene text recognition in English, Chinese and Arabic language inspired us to pose a benchmark scene text recognition for Ethiopic script. To transcribe the word images to the cross bonding text, we use a segmentation free end-to-end trainable Convolutional and Recurrent Neural Network (CRNN) hybrid architecture. In the network, robust representation features from cropped word images are extracted at convolutional layer and the extracted representations features are transcribed to a sequence of labels by the recurrent layer and transcription layer. The transcription is not bounded by lexicon or word length. Due to it is effective uses to transcribe sequence-to-sequence tasks, CTC loss is applied to train the network. In order to train the proposed model, we prepare synthetic word images from Unicode fonts of Ethiopic scripts, besides the model performance is evaluated on real scene text dataset collected from different sources. The experiment result of the proposed model, shows a promising result.

Keywords: Scene text recognition · Deep learning · Ethiopic script

1 Introduction

Extracting and analyzing scene text information found in the natural image, which carries high-level semantics has an extensive variety of uses including content-based image retrieval, tourist assistance, instant translation, assist visually impaired person, unmanned ground vehicle navigation, etc. Due to these, scene text recognition in computer vision and document analysis fields has recently received increasing attention. However, the diversity and variability of texts in natural images, such as written in different languages, available in different font colors, font styles, font sizes, text orientations, and shapes are the main challenges to develop a robust scene text detection method. Moreover, the complexity of unpredictable backgrounds and poor imaging conditions due to low resolution and severe distortions [1] are another challenges.

To ease the challenges of scene text recognition, several methods are proposed using traditional and deep learning techniques. The traditional algorithms were tested while the results were not satisfactory. Recently, the limitations of traditional algorithms in different areas are addressed using deep learning techniques and it shows a promising

N. G. Habtu et al. (Eds.): ICAST 2019, LNICST 308, pp. 502–511, 2020.
https://doi.org/10.1007/978-3-030-43690-2_36

result. In addition to its recognition performance, the deep learning techniques facilitate end-to-end trainable methods by freeing researchers from the exhaustive work of repeatedly designing and testing hand-crafted features.

The task of scene text recognition method in traditional approach possesses preprocessing, character segmentation and character recognition phases whereas in deep learning approach text detection and recognizing the detected text is the main tasks. In both traditional and deep learning approaches, several methods are proposed for scene text detection including Connected Components Analysis [2, 3], Sliding Window [4], MSER [5], EAST [6], SegLink [7], Corner localization [8], PixelLink [9] and TextSnake [10]. Using these scene text detection methods, several researchers proposed recognition methods such as RNN stacked with CNN [11], sequence prediction with attention-based models [17]. And they got promising scene text detection and recognition performance for Latin, Arabic, and Chinese scripts. However, there is no research for Ethiopic script as far as the researchers' knowledge is concerned.

Ethiopic script which is previously known as Ge'ez (ግዕዝ) is one of the oldest writing systems in the world [12]. It uses as a writing system for more than 43 languages including Amharic, Geez, Tigrigna, and others. The script has been largely used by Ge'ez and Amharic language, which are the liturgical and official languages of Ethiopia and some states of USA, respectively. The script is written down in a tabular format in which the first denotes the base character and the other columns are vowels derived from the base characters by slightly deforming or modifying the base characters. Ethiopic script has a total of 466 characters, out of these, twenty characters are digits, nine characters are punctuation marks, and the remaining 437 characters are alphabets. Developing a scene text recognition system for Ethiopic script is challenging, due to availability of similar characters especially between base characters and the derived vowels and there are number of many characters. Furthermore, unavailability of training and testing datasets are another limitation in the development of a model that can detect and recognize scene texts written in Ethiopic scripts.

In this paper, we use a modified version of CRNN [11] hybrid network that can train in an end-to-end manner. In addition, we prepare a synthetic and real dataset for training and testing the proposed model, respectively. In summary, the contributions of this paper are listed as follows:

- For training the proposed model, synthetic scene text datasets are prepared by changing the background textures and images, colors of text and other parameters.
- For testing the proposed model and to benchmark the recognition performance, hundreds of real scene text dataset was prepared.
- A segmentation free and end-to-end trainable CRNN model is employed.

The rest of the paper is presented as follows; previous related works are introduced in Sect. 2. In Sect. 3, we discuss about the proposed CRNN hybrid network. The experiment set-up, dataset and experiment results with discussions are conveyed in Sect. 4. To end, the conclusion and recommendations are drawn in Sect. 5.

2 Related Work

Scene text detection and recognition is currently the active area of research in computer vision and document analysis. In this section, we provide a short introduction to previous related works on methods of text detection and recognition.

2.1 Text Detection

Scene text detection is the sub process of text reading problem from natural images. Its main objective is to detect text areas from the given natural input image using different methods. Researchers uses different approaches to detect text areas from a natural image. Sliding window [4] and connected component [2, 3] methods are the most common conventional approaches for detecting scene text by considering the text as a composition of characters. Recently, deep learning techniques are applied to directly detect words from a natural image. As stated in [13], a vertical anchor mechanism was used to predict the fixed width sequential proposals and then connect them. Ma et al. [14] presents a novel rotation-based framework based on region proposal architecture for detecting arbitrary oriented texts from natural images. He et al. [7] presents deep direct regression methods for multi-oriented scene text detection. The model predicts words or text lines of arbitrary oriented and quadrilateral shapes in full images. In this paper, we focus on the recognition part of cropped scene images.

2.2 Text Recognition

In the text reading phases of natural images, text recognition is the second phase next to scene text detection. This method can be implemented independently or after scene text detection phases. In the scene text recognition phase, the cropped text regions are feed either from the scene text detection phase or from prepared input dataset and sequence of labels are decoded. Previous attempts were made by detecting individual characters and refine misclassified characters. Such a methods require training a strong character detector for accurately detecting and cropping each character out from the original word. These type of methods are more difficult for Ethiopic scripts due to its complexities. Despite to character level methods, word recognition [15], sequence to label [16], and sequence to sequence [17] methods are presented. Liu et al. [18] and Shi et al. [19] presents a spatial attention mechanism to transform a distorted text region from irregular input images into canonical pose suitable recognition. Shi et al. [11] presents a unique end-to-end trainable method by combining the robust convolutional features of CNN and transcription abilities of RNN. We use this design where the hybrid CNN-RNN network with a Connectionist Temporal Classification (CTC) loss is trained end-to-end.

3 CRNN Hybrid Architecture

The proposed CRNN hybrid network architecture has three fundamental components, including convolutional layers, recurrent layers, and transcription layers. The proposed CRNN hybrid network architecture is shown in Fig. 1. In the architecture, the image that contains scene texts are fed into the first convolutional layer then the sequences of features are extracted automatically using CNN network. Using the extracted sequence of features, the recurrent layer built prediction for each frame of the feature sequence. Finally, the transcription layer translates per-frame prediction into a label sequence by recurrent layers. The details of each layer are presented in the following subsections.

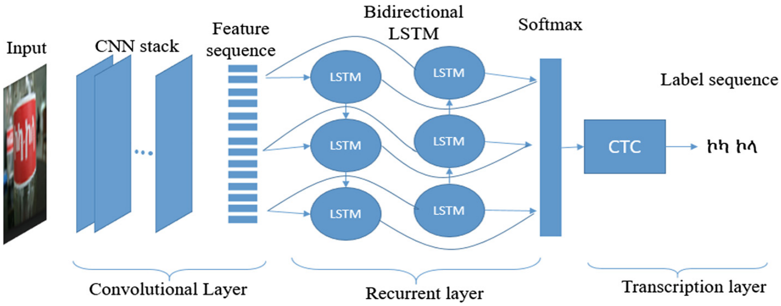

Fig. 1. The architecture of hybrid CRNN network

3.1 Convolutional Layer

The first layer of CRNN hybrid network architecture called convolutional layer is built based on the VGG [20] network architecture, with the exception of fully connected layers. The layer has a number of convolution and max-pooling layers. The convolution layer of CNN network extracts silent features of the input image such as edges, corners, and endpoints by applying convolution operation. From this operation, k-feature maps are extracted with the size of $(M - N + 1) \times (M - N + 1)$ from input images that have $M \times M$ square neuron nodes by using $N \times N$ convolutional kernel. The non-linearity decision function and the entire network is increased using the activation function in our case we use ReLU. Most commonly, the input images come in different sizes so that they have to be rescaled to the same size before being fed into the model. Max-pooling layer is another layer in CNN part of the model, which is periodically added between successive convolution layers and spatial invariance is achieved. In addition, the size of the representation feature maps, as well as the number of parameters and amount of computation in the network, are reduced, whereby overfitting is controlled. The extracted sequence of features from the input image uses as an input to the next recurrent layer.

3.2 Recurrent Layer

Next to convolutional layer, the recurrent layer is built using bidirectional RNN. The recurrent layer predicts a label y_t for each input frame x_t in the feature sequence $x_1, x_2, \ldots, x_T$. The advantages of using recurrent layer are can capture contextual information within a sequence, back propagate error differentials to its inputs, and able to work on a sequence of arbitrary lengths. Instead of treating each symbol independently, it is more stable and helpful to use contextual guide for image-based sequence recognition. In RNN, there are a number of self-connected hidden layers between input and output layers. As we discuss above the input for the recurrent layer is the output of the convolutional layer and after performing several recurrent operations, the recurrent layer predicts an output y_t. The predicted label y_t is based on the internal state h_t and also the value of hidden state h_t is determined based on the current input x_t, previous hidden state h_{t-1} and non-linear function. Mathematically described as $h_t = g(x_t, h_{t-1})$.

However, RNN have a limitation on learning long-term dependency. Because during training RNN with the gradient-based Backpropagation through time (BPTT) technique, it is difficult to build a precise model due to the vanishing and exploding gradient problems. To address this problem, LSTM networks was introduced by Hochreiter and Schmidhuber in 1997 [24], and improved the network structure by Gers et al. [25], to avoid the long-term dependency problem. LSTM network is a specific RNN architecture and state-of-the-art deep learning algorithm which introduces gates (i.e. input, output, and forget gate memory cells) to prevent gradient problems from vanishing and exploding. As the name indicates, the LSTM network can remember long-term values from several time steps as well as the short-term values. In this paper, we use two bidirectional LSTM designed by combining one forward and one backward LSTM.

3.3 Transcription Layer

Transcription layer is the last layer of the proposed CRNN network which takes the predictions from the recurrent layer and convert into label sequences by finding the highest probability. The transcription of label sequences can be performed either based on lexicon or lexicon free methods. In lexicon free methods, the prediction is made without lexicons whereas in lexicon based methods, highest probability label sequences are predicted from the collected lexicons. In this paper, we use a lexicon free transcription method to translate the predictions into label sequences for the target language.

4 Experiments and Results

4.1 Dataset Preparation

In any machine learning technique, dataset plays an important role to train and obtain a better machine learning model. Especially, deep learning methods are data hungry than traditional machine learning algorithms. However, preparing a large dataset was also a challenging task specifically for under resource languages. In [15], a deep learning based synthetic scene text dataset generation method was proposed. In this paper, we use a synthetically generated scene text dataset and real scene text dataset for training and testing the proposed model, respectively.

Synthetic Scene Text Dataset

As stated in [15], deep learning techniques are used to generate synthetic scene text datasets. The generated scene text images match similar to real scene images. This technique is very important to get more training data for those scripts that don't have prepared real scene text datasets. Using this technique, several synthetic scene text datasets are prepared to train and build a scene text recognition model, for instance, Chinese, Arabic, Indian and Latin scripts. As far as the researchers' knowledge there is no prepared real scene text dataset for Ethiopic script. Therefore, using a similar approach, we prepare a synthetic scene text dataset for Ethiopic script.

To prepare the synthetic scene text dataset, 540,735 words (8, 250,800 characters) are collected from different social, political, and governmental websites written in Ethiopic script. Ethiopic script uses for more than 43 languages, even if we include all Ethiopic characters, the collected data is written in Amharic, Geez and Tigrigna languages. Using these collected words and freely available 72 Unicode fonts, words are rendered into the foreground layer of the image to form the synthetic scene images. To make more robust the generated synthetic scene images the font color, font size, rotation along horizontal line, skew and thickness parameters are tuned. In addition, we use different images as a background. Based on these, we generate 500K number of synthetic scene text word images. Sample images are shown in Fig. 2.

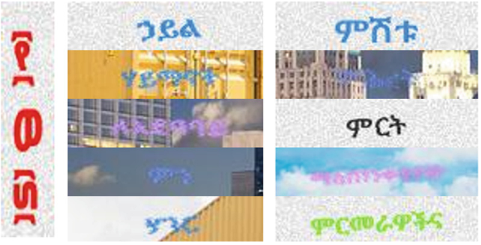

Fig. 2. Sample generated artificial scene text image

Real Scene Text Dataset

Besides to synthetic dataset, we prepare hundreds of real scene text benchmarking dataset collected from local markets, banners, navigation and traffic signs, billboards and image search using Google. After collecting the scene text images, equivalent texts of each image are annotated by bounding the text areas on the image. Based on this, we prepare around 2,500 scene text word images from hundreds of collected images. This dataset can also be used for the scene text detection task. In this paper, we only deal with the recognition of scene texts from the cropped word images. Sample real scene text images are shown in Fig. 3.

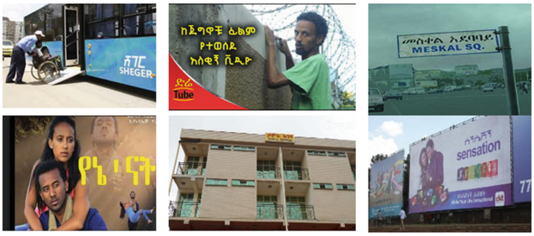

Fig. 3. Sample real scene text images before crop

4.2 Implementation Details

As discussed before, the hybrid CRNN network have convolution layer, recurrent layer, and transcription layer. The proposed CRNN network implementation detail is shown in Table 1. The convolution layer have a CNN network architecture which is designed based on the VGG architecture [20]. To get wider features from the input image and consecutive convolution layers, we use 1 × 2 pooling stride at the 3rd and 4th max-pooling layers instead of conventional 2 × 2 strides. To enable faster learning, the input images are converted into grayscale and rescaled into a fixed width and height (128 × 64) pixels. The training process for both CNN and RNN is very difficult in terms of computing power and time. To increase the training speed of the network, we add batch normalization [21] layers after 5th and 7th convolution layers. Even if batch normalization increases the training speed, the experiment shows that using at each convolution layer decreases the recognition performance.

After extracting features using convolution stacks, two BLSTM layers with 512 neurons are followed. The last BLSTM layer is fully connected with 467 number of label sequences, i.e. the number of characters in the script plus one additional extra label for blank. Finally, using the Softmax activation function on the outputs of the BLSTM layer, the CTC loss is calculated between the predicted probability and the actual value. Based on the variance between the predicted and the real values, the forward-backward [22] and backpropagation through time algorithm are applied to adjust the parameters of the network. The network parameters are optimized with AdaDelta optimization [23] while training the network using stochastic gradient descent (SGD).

The experiments are executed on Ubuntu machine containing Intel Core i7-7700 (3.60 GHz) CPU with 64 GB RAM and GeForce GTX 1080 Ti 11176 MiB GPU. For the implementation of the proposed system, we use Python 3.6 and library with TensorFlow backend.

Table 1. Configuration of the proposed network

Type	Configuration
Input	128 × 64 gray scale image
Conv2D	64, kernel 3 × 3, stride 1 × 1, padding 1 × 1
Max-pooling	Kernel 2 × 2, Stride 2 × 2
Conv2D	128, kernel 3 × 3, stride 1 × 1, padding 1 × 1
Max-Pooling	Kernel 2 × 2, Stride 2 × 2
Conv2D	256, kernel 3 × 3, stride 1 × 1, padding 1 × 1
Conv2D	256, kernel 3 × 3, stride 1 × 1, padding 1 × 1
Max-Pooling	Kernel 2 × 2, stride 1 × 2

(*continued*)

Table 1. (*continued*)

Type	Configuration
Conv2D	512, kernel 3 × 3, stride 1 × 1, padding 1 × 1, BN
Conv2D	512, kernel 3 × 3, stride 1 × 1, padding 1 × 1
Max-Pooling	Kernel 2 × 2, stride 1 × 2
Conv2D	512, kernel 3 × 3, stride 1 × 1, padding 1 × 1, BN
Map-to-sequence	–
BLSTM	#hidden 512
BLSTM	#hidden 512
Transcription	–

BN – stands for Batch normalization

4.3 Evaluation Metrics

The recognition performance of the hybrid CRNN network on the benchmarked real scene text dataset is evaluated at word level and character level using Word Recognition Rate (WRR) and Character Recognition Rate (CRR). CRR is the difference between total characters and the sum of Levenshtein distance between the Recognized Text (RT) and Ground Truth (GT) divided by the number of total characters.

$$CRR = \frac{C - \sum d(RT, GT)}{C} \tag{1}$$

where C is the number of total characters, d is the Levenshtein distance between RT and GT. Whereas WRR is computed by dividing the number of Correctly Recognized Text (CRT) by the number of Total Words (TW).

$$WRR = \frac{CRT}{TW} \tag{2}$$

4.4 Experiment Results and Discussion

The experiment is conducted to evaluate the recognition performance of the proposed hybrid CRNN network. Based on the network configurations stated above, the proposed model is trained using the prepared training dataset and its recognition performance is tested using testing datasets.

The recognition performance of the proposed model on the prepared real scene text dataset achieves 87.50% and 90.33%, WRR and CRR, respectively. In addition, to test the effects of using synthetic dataset, we test the proposed model using synthetic generated dataset and achieves a recognition performance of 94.35 and 97.23 for WRR and CRR, respectively. This shows that, using synthetic dataset has an important impact with some limitations because there is a big testing result difference between the real and synthetic testing dataset recognition results. This indicates that, using real scene text dataset for training the model will increase the current recognition performance of the proposed model. Sample recognition outputs are presented in Fig. 4.

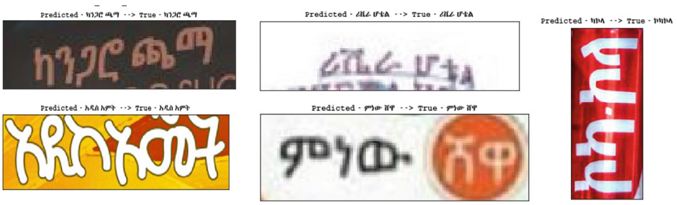
Predicted - ከንጋሮ ጫማ --> True - ከንጋሮ ጫማ
Predicted - ሪቨራ ሆቴል --> True - ሪቨራ ሆቴል
Predicted - ከኮላ --> True - ኮካኮላ
Predicted - አዲስ አምት --> True - አዲስ አምት
Predicted - ምነው ሸዋ --> True - ምነው ሸዋ

Fig. 4. Sample predicted outputs

5 Conclusion

In this paper, we present a scene text recognition method which is written in Ethiopic scripts using CRNN hybrid network. The convolution layer of the network automatically extracts sequences of features from the input image and fed as an input to the next recurrent layer. The recurrent layer maps the sequence of extracted features into a sequence of labels. Finally, the CTC computes the loss between the predicted labeled sequences and the actual value. Using synthetically generated scene text dataset, the model is trained in an end-to-end fashion and its performance is evaluated using real scene text dataset collected from different sources. The experiment results show that the proposed model is promising. The introduction of new dataset and initial experiment results may introduce other researchers to pursue and improve scene text recognition which is written in Ethiopic script. In the future, we will increase the size of the real scene text dataset and incorporate a scene text detector rather than using cropped images. In addition, most commonly natural images contain more than one script, so we will conduct an end-to-end trainable multi script scene text detection, scene text recognition, and text spotting system.

References

1. Liang, J., Doermann, D., Li, H.: Camera-based analysis of text and documents: a survey. Int. J. Docu. Anal. Recogn. (IJDAR) **7**(2–3), 84–104 (2005)
2. Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, pp. 2963–2970. IEEE (2010)
3. Neumann, L., Matas, J.: A method for text localization and recognition in real-world images. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6494, pp. 770–783. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19318-7_60
4. Neumann, L., Matas, J.: Scene text localization and recognition with oriented stroke detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 97–104 (2013)
5. Donoser, M., Bischof, H.: Efficient maximally stable extremal region (MSER) tracking. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), New York, NY, USA, pp. 553–560. IEEE (2006)
6. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W.: EAST: an efficient and accurate scene text detector. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 5551–5560. IEEE (2017)
7. He, W., et al.: Deep direct regression for multi-oriented scene text detection. In: Proceedings of the IEEE International Conference on Computer Vision (2017)

8. Lyu, P., Yao, C., Wu, W., Yan, S., Bai, X.: Multi-oriented scene text detection via corner localization and region segmentation. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, pp. 7553–7563. IEEE (2018)
9. Deng, D., Liu, H., Li, X., Cai, D.: PixelLink: detecting scene text via instance segmentation. In: Proceedings of the 2018 Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, pp. 7553–7563. IEEE (2018)
10. Long, S., Ruan, J., Zhang, W., He, X., Wu, W., Yao, C.: TextSnake: a flexible representation for detecting text of arbitrary shapes. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11206, pp. 19–35. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01216-8_2
11. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2017)
12. De Voogt, A.: The cultural transmission of script in Africa: the presence of syllabaries. Scripta **6**, 121–143 (2014)
13. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 56–72. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_4
14. Ma, J., et al.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans. Multimed. **20**(11), 3111–3122 (2017)
15. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227, 1–10 (2014)
16. Su, B., Lu, S.: Accurate scene text recognition based on recurrent neural network. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9003, pp. 35–48. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16865-4_3
17. Lee, C.Y., Osindero, S.: Recursive recurrent nets with attention modeling for OCR in the wild. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 2231–2239 (2016)
18. Liu, W., Chen, C., Wong, K.Y.K., Su, Z., Han, J.: STAR-Net: a SpaTial attention residue network for scene text recognition. In: Wilson, R.C., Hancock, E.R., Smith, W.A.P. (eds.) Conference 2016, BMVC, York, UK, vol. 2, p. 7 (2016)
19. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 4168–4176 (2016)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 1–14 (2014)
21. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167v3, vol. abs/1502.03167, 1–11 (2015)
22. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 2006 ICML Conference on Machine Learning, Pittsburgh, Pennsylvania, USA, pp. 369–376 (2006)
23. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)
24. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **8**(9), 1735–1780 (1997)
25. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to forget: continual prediction with LSTM. Neural Comput. **12**(10), 2451–2471 (2000)