2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Geometric Alignment by Deep Learning for Recognition of Challenging License Plates

Jakub Špaňhel, Jakub Sochor, Roman Juránek, Adam Herout
Graph@FIT, Brno University of Technology
Brno, Czech Republic
{ispanhel,herout}@fit.vutbr.cz

***Abstract*—In this paper, we explore the problem of license plate recognition in-the-wild (in the meaning of capturing data in unconstrained conditions, taken from arbitrary viewpoints and distances). We propose a method for automatic license plate recognition in-the-wild based on a geometric alignment of license plates as a preceding step for holistic license plate recognition. The alignment is done by a Convolutional Neural Network that estimates control points for rectifying the image and the following rectification step is formulated so that the whole alignment and recognition process can be assembled into one computational graph of a contemporary neural network framework, such as Tensorflow. The experiments show that the use of the aligner helps the recognition considerably: the error rate dropped from $9.6\,\%$ to $2.1\,\%$ on real-life images of license plates. The experiments also show that the solution is fast – it is capable of real-time processing even on an embedded and low-power platform (Jetson TX2). We collected and annotated a dataset of license plates called *CamCar6k*, containing 6,064 images with annotated corner points and ground truth texts. We make this dataset publicly available.**

***Index Terms*—License Plate Recognition, CNN, License Plate Dataset, Image Alignment, Intelligent Transportation Systems**

I. Introduction

Automatic License Plate Recognition (ALPR) is the backbone for many applications in traffic surveillance and intelligent transportation systems (automatic parking systems, security surveillance systems, toll gates, etc.). In many such applications, the cameras are fixed and positioned so that the license plates share a common size (image resolution), orientation and they are not skewed. In such scenarios, the existing recognizers of license plates achieve almost perfect results. However, mobile monitoring platforms are used more frequently for parking enforcement and other applications, and also the availability and properties of PTZ (pan-tilt-zoom) cameras offer for much less restricted scenarios. Existing solutions of automatic license plate recognition (an overview is available in Section II) are not designed for these unconstrained cases and tend to achieve poor results.

Špaňhel et al. [22] proved that holistic (i.e. refraining from segmenting the characters) recognition outperforms other available recognition methods. However, their work does not deal with challenging license plates captured from different viewpoints. In this work, we propose a solution how to overcome that limitation. We designed a new convolutional neural network (CNN) whose purpose is to predict four corner points of the license plate in the unaligned image.

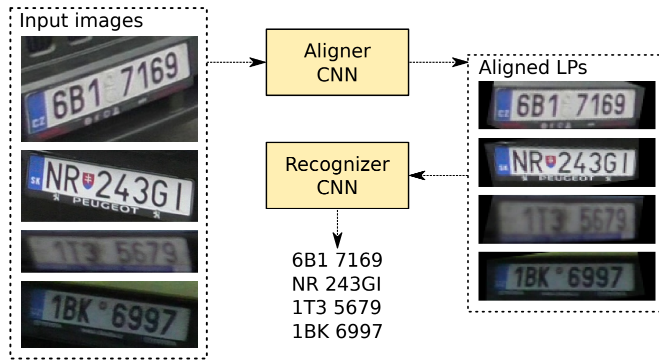

Fig. 1. Overview of our approach. The input license plates of various resolutions, padding, rotation, and skew are analyzed by the aligner CNN, rectified and processed by a holistic license plate recognizer. The whole process can be incorporated into one computational graph and computed as a whole, for example in a GPU.

These points define the transformation which rectifies the image for subsequent processing by the holistic license plate recognizer. Although the neural networks are trained separately, they can be assembled into one computational graph so that the solution works as a whole, its use is no more complicated than using only the recognizer.

We carried out experimental evaluation of the proposed approach. The proposed aligner network decreases the recognition error on real-life license plates considerably. We evaluated the sensitivity of the newly proposed pipeline to various distortions of the input images: rotation, skew, blur, noise, etc. Again, the aligner network helps the stability notably. We also evaluated the speed of the proposed recognition pipeline. Thanks to the fact that it can be assembled with the recognizer into one computational graph, it can be efficiently (in real time) executed even on low-power and embedded devices. All these findings make the proposed approach interesting in applications of traffic enforcement and intelligent transportation systems.

The contributions of this paper are the following:

- A method for aligning license plates before recognition based on CNN.
- Novel **CamCar6k** dataset of in-the-wild license plates.
- Detailed evaluation of aligner/recognizer performance influenced by different distortions.

II. Related Work

Rotation and distortion causes *segmentation-based* ALPR methods [9], [14], [17], [19]–[21], [23] to fail in most cases. Masood et al. [17] proposed to first detect and segment out the characters and then recognize them by a convolutional neural network and they integrated their approach to *Sighthound Cloud API*. Even recent work from Laroca et al. [14] uses character segmentation combined with individual character recognition based on CNN. From this class of methods, only one group can be used for in-the-wild license plate recognition under limited conditions. These methods are based on connected component techniques (CC-based) and they can deal with rotation and distortion of the license plate [1], [2], [5], [25]. CC-based techniques label blocks of pixels from binarized LPs, depending on 4- or 8-neighborhood connectivity and they use these blocks to segment the license plate. Unfortunately, these methods cannot segment characters correctly if they are connected together or broken.

Another approach is to use OCR systems designed for reading multi-character text in the wild, which are used to recognize digits and characters in different types of applications (e.g. reading house numbers and other texts on facades), where character segmentation of the input data can be also difficult. Goodfellow et al. [7] proved that neural nets are capable of recognizing multi-character texts without segmenting the characters in unconstrained natural photographs, which was confirmed by Jaderberg et al. [12].

Finally, *segmentation-free* methods [3], [11], [13], [15], [22] have state-of-the-art results in license plate recognition in past few years. Li and Shen [15] developed the first segmentation-free license plate recognition method. Their method is based on CNN for feature extraction and bi-directional recurrent neural network with LSTM units and connectionist temporal classification (CTC) [8] for sequential data labeling. Li et al. [16] improved their previous work using region proposal network for license plate detection and output of BRNNs is linearly transformed before CTC is involved.

Špaňhel et al. [22] proved that it is possible to have state-of-the-art results in license plate recognition, even on low quality data without segmenting individual characters. Unfortunately, their proposed method was only developed for recognizing license plates that are axis-aligned (with a small tolerance). Therefore, this approach is not directly applicable to license plate recognition in the wild.

III. Methodology

This section presents our processing pipeline, mostly the newly added aligner CNN and the way the whole system is assembled and learned. It should be noted that this work is not focused on license plate detection as we assume that the license plates are detected by an existing technique in software [1], [5], [11], [15], [21], hardware [24], or GPUs [10]. The license plate detector does not need to be very sophisticated and the system can tolerate its relatively high false positive rate, since the candidate locations are verified by the aligner/recognizer presented in this work.

A. Hourglass Network for Keypoint Detection

The proposed Aligner CNN is shown in Figure 3. It is based on the stacked hourglass neural network as designed by Newell et al. [18]. The Aligner CNN is a fully convolutional network which for each image point evaluates the probability that the point contains one of the four corner points. The network contains three hourglass modules which downsample and upsample the features in the spatial dimension. For better gradient flow, the network contains skip connections and an additional output layer with MSE or Binary Cross-Entropy losses after each hourglass module. The hourglass design allows the network to process and consolidate features across different scales. It has the capacity to capture all of these features together and create per-pixel predictions at the output. For further details about the model, we refer readers to the original paper [18].

The network is trained using randomly rotated images of license plates and ground truth probability maps for each corner point are used as the supervision. To reduce the computational complexity, we used only three hourglass modules and feature size 64.

B. License Plate Processing Pipeline

The hourglass network estimates the locations of the inner corner points of the license plate. Its outputs are four probability maps for the four corner points in a specified order (top left, top right, bottom right, bottom left). If a keypoint is not present in the input image (occlusion, damage, image does not contain a license plate), the peak is missing in the appropriate heatmap. An example of the predicted heatmaps merged into one can be found in Figure 2 middle (*merged predictions*) and in Figure 4.

The maximum of each predicted heatmap is used as the detected keypoint, shown in Figure 2 (*detected keypoints*). The detected keypoints are then used to normalize the license plate using homography $\mathbf{H}$ between the estimated points and predefined axis-aligned corner points.

Finally, aligned license plates depicted in Figure 2 (*aligned license plate*) are passed to the recognizer network to provide the final recognition.

C. Recognizer Network

Recognition of the aligned license plates is based on a CNN proposed by Špaňhel et al. [22]. The proposed network processes the whole image at once without character segmentation. During training, the CNN learns the presumed location for each output layer (each character of the LP) from the training data. The network is composed of three blocks sharing the same structure: three identical subblocks containing the convolutions and nonlinearity (3×3 convolution layer + ReLU + Batch normalization) followed by one 2×2 max pooling. At the end, the network predicts eight different characters independently.

During inference for *aligner+recognizer* network, both variants are evaluated (even the variant without the aligner)

Fig. 2. The whole license plate processing pipeline. The aligner outputs four heatmaps based on which the LP's corner points are found and the license plate is transformed/aligned. Then, the existing state-of-the-art recognizer processes it.

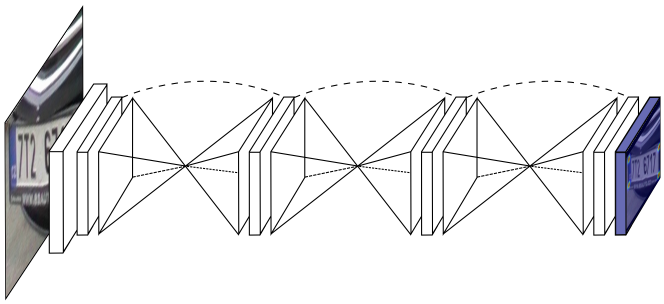

Fig. 3. Hourglass architecture of the proposed Aligner CNN for estimation of the corner points. It consists of three stacked hourglass modules, allowing repetitive top-down, bottom-up inference with skip connections. The Aligner CNN estimates the probability map for each corner point independently, in the case of a missing corner points, the probability map is (close to) zero over all its surface.

Fig. 4. Examples of the alignment process. From left to right: input image, merged predictions, detected keypoints, aligned license plates.

and the result with higher confidence is selected as the final output. It results in an increase of recognition accuracy; however, the processing speed decreases, which can be seen in Section V, Table II.

IV. Datasets for ALPR

Publicly available datasets of license plate images are a scarce commodity globally. Recently, Špaňhel et al. [22] published two datasets designed for license plate recognition – **ReId** dataset (182,336 images), composed primarily from low quality license plate images captured in a surveillance-like setup and **HDR** dataset (657 images), containing hand-cropped samples of license plates from HDR images taken from arbitrary viewpoints (i.e. slightly rotated and distorted images). Unfortunately, none of these datasets contains ground truth annotations of license plate corners' positions. They are therefore not suitable neither for training or evaluation of license plate alignment.

Very recently, Laroca et al. [14] published the *UFPR-ALPR* dataset with 4,500 images FullHD images with annotated license plates. However, the data was captured from a camera fixed at a vehicle's windshield, thus with minimal distortions of the LPs. The same is true for the dataset created by Gonçalves et al. [6] which is even smaller (only 2,000 images).

A. *CamCar6k* – *Public Dataset of License Plates in the Wild*

We recorded 7.5 hours of video of license plates from different viewpoints taken by four cameras mounted on a vehicle passing among vehicles parked on streets and/or parking lots. The recordings cover almost all possible styles of parking (parallel, angle, perpendicular) both outside (streets, outdoor parking lots) and inside (parking garages).

In each frame of the video, license plates were detected by a Boosted Soft Cascade detector [4]. In order to deal with the range of rotations and perspective projections, we used three detectors, each tuned to a different range of deformations. The outputs of these detectors were merged and formed the final set of detections. Each detection was processed with a preliminary version of the recognizer in order to filter out false detections.

Original image frames with at least one detection were stored and $5,000$ frames with detected license plates were chosen randomly and further processed by users. Their task was to annotate the *inner corners* of each license plate and transcribe the ground truth text for each image. Thus created dataset **CamCar6k** contains $6,064$ images of license plates.

The locations of annotated keypoints allow us to generate more training samples by rotation, translation, shear and zoom. The distribution of the rotations of the original license plates in the collected data is shown in Figure 6. It can be seen that majority of license plates are rotated in an interval of $\pm 20°$.

The **CamCar6k** dataset was divided into a *training* and a *test* split, containing $2,750$ and $3,314$ images of license plates, respectively. In our experiments (Section V), we use **ReId**, **HDR** and **CamCar6k** datasets mixed together for learning the LP recognizer. We are making the **CamCar6k** dataset publicly available for non-commercial purposes[1].

[1] https://medusa.fit.vutbr.cz/traffic

Fig. 5. Samples of license plate with annotated keypoints used for training aligner and recognizer. **Left:** Samples from *CamCar6k* dataset. **Right:** Samples from synthetic data (Sec. IV-B).

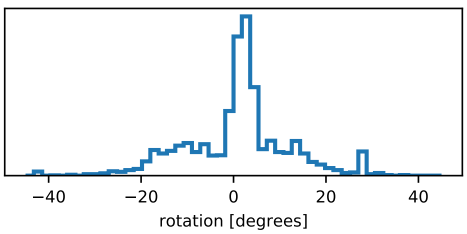

Fig. 6. Distribution of rotation of license plates in the newly collected **CamCar6k** dataset. The majority of LPs are almost horizontal, but the dataset also contains a good coverage of angles between $\pm 20°$.

TABLE I
ACCURACY OF LICENSE PLATE RECOGNITION WITH DIFFERENT VERSIONS OF ALIGNER (ROWS)/RECOGNIZER (COLUMNS).

	RECOGNIZER		
ALIGNER	Real	Real+Synth	OpenALPR
None	87.5	90.4	69.4
Real	96.0	97.9	73.6
Real+Synth	95.8	97.7	71.4

B. Dataset of Synthetic Images of Cars with License Plates

With image transformations, the texts of the annotated license plates remain unchanged. It is impossible to collect all valid variants of license plates, or combinations of all the allowed characters. To avoid misclassification of license plates with previously unseen characters and their combinations, we developed a tool for generating synthetic data. Given real images with annotated corner points of the license plates, a template of the country-specific license plate layout, and an appropriate font for characters, the tool is able to generate any requested number of license plates with different text and rotation/distortion of the source image. We generated $100\,\mathrm{k}$ synthetic license plates of multiple countries (denoted by *synth* in the following text), which are used only in the training phase (we consider it appropriate to only evaluate on real-world data).

V. EXPERIMENTS

In our experiments, we evaluated different combinations of the aligner and the recognizer networks. We trained two variants of the **aligner** network, different in the datasets used for training. The **Real** variant is trained on the *training* split of the *CamCar6k* dataset only because this real-world dataset contains the annotations of the LP corners. The **Real+Synth** option expands the training set by adding the synthetic data in the training phase. The networks were trained with learning rate 2.5e-4, batch size 16, and input resolution 128×128. The output probability maps have shape $32 \times 32 \times 4$ as we are estimating 4 corner points. The network was shown 5 millions randomly transformed samples during the training.

On the other hand, the **recognizer** network trained on the real-world data (denoted by **Real**) is trained using all data from *ReId* and *HDR* and the *training* part of the *CamCar6k* dataset because the corner point annotations are not necessary. The second variant **Real+Synth** expands the training set by synthetically generated images just like in the case of the aligner network. The networks were trained with learning rate 0.01 and Adam optimizer for 20 epochs. The input image shape is 128×35. During the training, the input license plates were randomly rotated in range $\pm 15°$, shifted, and resized. These different variants are denoted in the form of **ALIGNER / RECOGNIZER** throughout the whole paper.

For evaluating the recognition accuracy and the processing speed, only the *test* split of the *CamCar6k* dataset was used. The evaluation of each variant of the aligner/recognizer networks can be found in Table I. The results show that using the aligner CNN considerably improves the recognition rate on the in-the-wild data: error rate reduced from $12.5\,\%$ to $4.0\,\%$. Extending the real-world dataset by synthetic data did not improve the aligner's performance (it even slightly degraded). On the other hand, using the synthetic data when learning the recognizer network helped visibly: error rate reduced from $4.0\,\%$ to $2.1\,\%$. Two publicly available license plate recognizers were also evaluated. OpenALPR[2] which is the most famous open-source solution for license plate recognition and license plate recognition Sighthound Cloud API[3]. Both solutions suffer from being based on character segmentation (Sec. II). Interestingly, Sighthound was able to detect at least some characters only on $1\,\%$ of license plates in the dataset so we omitted it from Table I.

A. Robustness of the Proposed Method

Robustness of the method against various distortions and influences is crucial for license plate recognition in the wild, so we made experiments with different types of distortions.

All the distortions were applied on the original image data (testing part of the *CamCar6k* dataset). The detailed

[2] OpenALPR – https://www.openalpr.com
[3] Sighthound Cloud API – https://www.sighthound.com/products/cloud

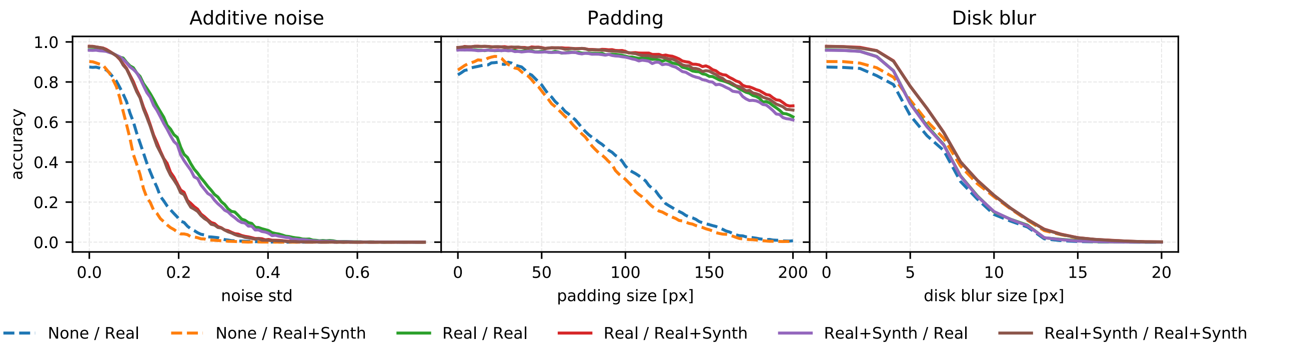

Fig. 7. Three different distortions added to the real-world images (testing part of the *CamCar6k* dataset). In all cases, the aligner improves the recognition performance considerably. In the case of added padding (the LP does not cover the whole region of interest, but a margin is added around), the recognizer helps tremendously – allowing for a high tolerance for the detector of the LPs.

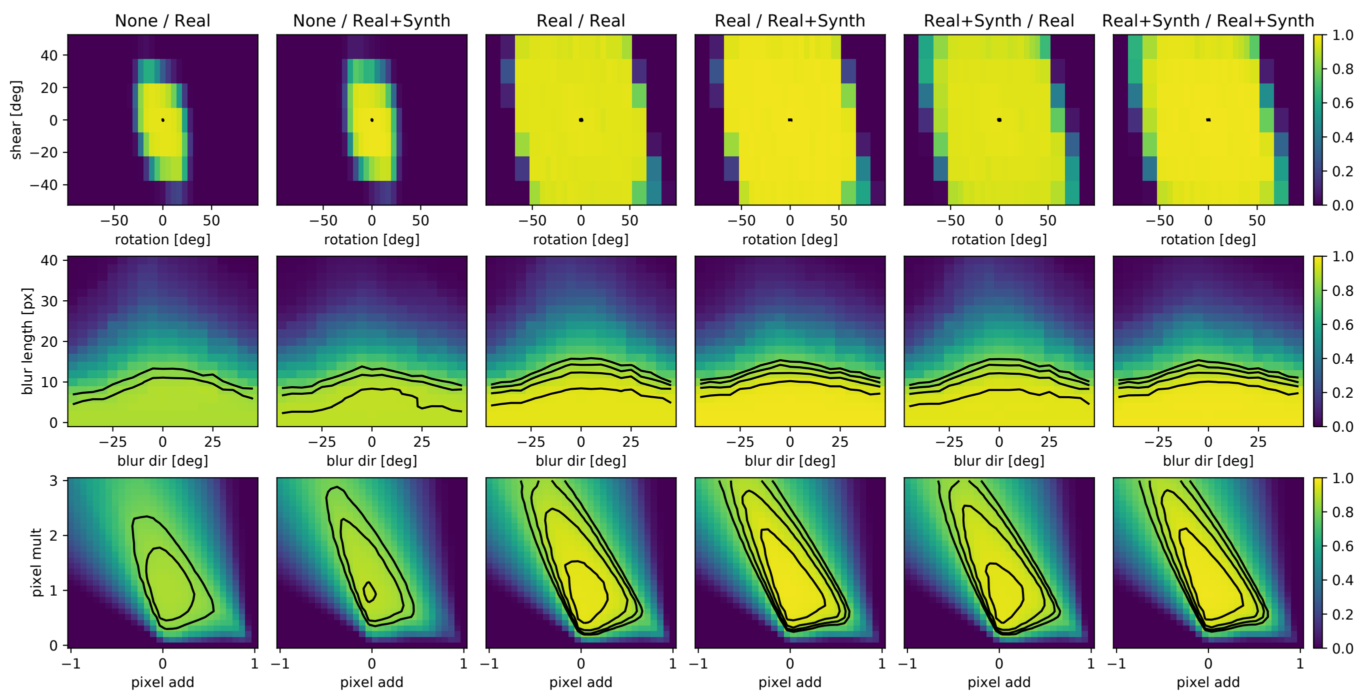

Fig. 8. Three different distortions added to the real-world images (testing part of the *CamCar6k* dataset), top to bottom: rotation vs. shear, simulated motion blur (direction + length), brightness / contrast adjustment. The versions with the aligner at work (columns 4 to 6) are clearly superior in all cases. Also, it should be noted that the Aligner CNN was trained for rotations $\pm 45°$.

results of these tests are shown in Figure 7 from left to right, respectively.

Additive noise: Noise added to the source image simulates varying quality of the used camera or its setting. Use of the aligner keeps the performance constantly better than without it and the drop with the increasing noise comes later. Interestingly, recognizer trained on purely real data does not drop its performance as quickly as the one trained also with synthetic data (though the second one performs slightly better when the noise is not added).

Padding: As expected, the aligner detects the extra padding around the license plate and the recognizer then can focus on the license plate instead of on its surroundings. The performance with the aligner remains almost unchanged and it starts to fall only with such a large padding which was not covered in the training data.

Out of focus blur: The recognition seems to be equally sensitive to disc blurring regardless of the use of the aligner.

The following distortions have two parameters, so they are visualized by using 2D graphs in Figure 8.

Rotation/Shear: The purpose of the aligner is to compensate for random rotations and skewed license plates. The test verifies that the aligner is very successful in this task and broadens the tolerance to these distortions greatly.

Motion blur: The motion blur (important factor in surveillance and monitoring of moving vehicles and/or from a moving vehicle) is simulated by applying blur of given length (in pixels) in a given direction (in degrees). Use of the aligner improves the results also in this case.

Brightness/Contrast changes: The last test studies how the performance is influenced by brightness/contrast changes,

TABLE II
PROCESSING SPEED OF RECOGNIZER ON DIFFERENT PLATFORMS.

Platform	Model	Recognizer ms	Recognizer FPS	Aligner+Recognizer ms	Aligner+Recognizer FPS
CPU	i5-6500	7.944	125.9	25.850	38.7
GPU	GTX 1080	0.686	1,457.3	1.877	532.7
SoC	Jetson TX2	4.456	224.4	16.466	60.7

simulated by adding and multiplying the individual pixel values.

B. Processing Speed

The processing speed of the recognizer and the combination of the aligner plus the recognizer was evaluated on different platforms: on CPU, GPU, and on an embedded device (SoC). The results are presented in Table II. The results show that the solution is able to greatly benefit from using contemporary GPUs – the coupled aligner with recognizer can process over 500 LPs per second. Even an embedded system, represented by NVIDIA Jetson TX2 platform (Tegra X2 chip, Pascal architecture, 256 CUDA cores), is capable of processing at least 60 LPs per second, with power consumption of only 7.5 Watts. This performance is sufficient for processing data from a connected camera in real time and it is suitable for usage in real-world applications.

VI. CONCLUSIONS

In this paper, we are dealing with the problem of recognizing license plates in the wild – rotated, skewed, blurred, noised, etc. Recognition of such license plate images is crucial for many applications from the domain of traffic enforcement and intelligent transportation systems.

We propose to precede the LP recognizer by an aligner having the Hourglass CNN architecture. The experimental results show that harnessing the aligner helped considerably: the error rate dropped from $9.6\,\%$ without it to $2.1\,\%$ on a real-world dataset captured by cameras mounted on a vehicle. The aligner is designed so that it can be assembled with the recognizer into one computational graph used by contemporary neural network platforms. This greatly helps the efficiency and the whole solution is able to work in real time even on low-power and embedded architectures (represented by Jetson TX2 in the experiments). The speed of processing 60.7 LPs per second on an embedded device outperforms current solutions significantly.

Along with this research, we collected a dataset of $6,064$ in-the-wild license plates and annotated their ground truth texts and four inner corner points. We are making the dataset **CamCar6k** publicly available for non-commercial use.

Acknowledgements This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project IT4Innovations excellence in science – LQ1602. Also, this work was supported by TACR project "SMARTCarPark", TH03010529.

REFERENCES

[1] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas. A license plate-recognition algorithm for intelligent transportation system applications. *IEEE T-ITS*, 7(3):377–392, 2006.
[2] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen. Automatic license plate recognition. *IEEE T-ITS*, 5(1):42–53, 2004.
[3] T. K. Cheang, Y. S. Chong, and Y. H. Tay. Segmentation-free vehicle license plate recognition using ConvNet-RNN. *arXiv preprint arXiv:1701.06439*, 2017.
[4] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Tran. PAMI*, 36(8):1532–1545, Aug 2014.
[5] I. Giannoukos, C.-N. Anagnostopoulos, V. Loumos, and E. Kayafas. Operator context scanning to support high segmentation rates for real time license plate recognition. *Pattern Recog.*, 43(11):3866–3878, 2010.
[6] G. R. Gonçalves, S. P. G. da Silva, D. Menotti, and W. R. Schwartz. Benchmark for license plate character segmentation. *Journal of Electronic Imaging*, 25(5):053034, 2016.
[7] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks, 2013.
[8] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Tran. PAMI*, 31(5):855–868, 2009.
[9] J.-M. Guo and Y.-F. Liu. License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques. *IEEE Tran. Vehicular Tech.*, 57(3):1417–1424, 2008.
[10] A. Herout, R. Jošth, R. Juránek, J. Havel, M. Hradiš, and P. Zemčík. Real-time object detection on CUDA. *JRTIP*, 6(3):159–170, 2011.
[11] G. S. Hsu, J. C. Chen, and Y. Z. Chung. Application-oriented license plate recognition. *IEEE Tran. Vehicular Tech.*, 62(2):552–561, Feb 2013.
[12] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *IJCV*, 116(1):1–20, 2016.
[13] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj, and K. R. Ramakrishnan. Deep automatic license plate recognition system. In *ICVGIP*, pages 6:1–6:8. ACM, 2016.
[14] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti. A robust real-time automatic license plate recognition based on the YOLO detector. *CoRR*, 2018.
[15] H. Li and C. Shen. Reading car license plates using deep convolutional neural networks and LSTMs. *arXiv preprint arXiv:1601.05610*, 2016.
[16] H. Li, P. Wang, and C. Shen. Towards end-to-end car license plates detection and recognition with deep neural networks. *arXiv preprint arXiv:1709.08828*, 2017.
[17] S. Z. Masood, G. Shu, A. Dehghan, and E. G. Ortiz. License plate detection and recognition using deeply learned convolutional neural networks. *arXiv preprint arXiv:1703.07330*, 2017.
[18] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016.
[19] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose. A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition*, 38(11):1961–1975, 2005.
[20] S. Qiao, Y. Zhu, X. Li, T. Liu, and B. Zhang. Research of improving the accuracy of license plate character segmentation. In *FCST*, pages 489–493. IEEE, 2010.
[21] S. Rasheed, A. Naeem, and O. Ishaq. Automated number plate recognition using Hough lines and template matching. In *WCECS*, volume 1, pages 24–26, 2012.
[22] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík, and P. Zemčík. Holistic recognition of low quality license plates by CNN using track annotated data. In *IWT4S (AVSS 2017)*, pages 1–6, 2017.
[23] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi. An algorithm for license plate recognition applied to intelligent transportation system. *IEEE T-ITS*, 12(3):830–845, 2011.
[24] P. Zemcik, R. Juranek, P. Musil, M. Musil, and M. Hradis. High performance architecture for object detection in streamed videos. In *FPL*, pages 1–4, Sept 2013.
[25] L. Zheng, X. He, B. Samali, and L. T. Yang. Accuracy enhancement for license plate recognition. In *CIT*, pages 511–516. IEEE, 2010.