License Plate Recognition and Super-resolution from Low-Resolution Videos by Convolutional Neural Networks

Vojtěch Vašek[1]
vojtech.vasek@eyedea.cz
Vojtěch Franc[2]
xfrancv@cmp.felk.cvut.cz
Martin Urban[1]
urbanm@eyedea.cz

[1] Eyedea Recognition
[2] Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

Abstract

The paper proposes Convolutional Neural Network (CNN) for License Plate Recognition (LPR) from low-resolution videos. The CNN accepts arbitrary long sequence of geometrically registered license plate (LP) images and outputs a distribution over a set of strings with an admissible length. Evaluation on 31k low-resolution videos shows that the proposed CNN significantly outperforms both baseline methods and humans by a large margin. Our second contribution is a CNN based super-resolution generator of LP images. The generator converts input low-resolution LP image into its high-resolution counterpart which i) preserves the structure of the input and ii) depicts a string that was previously recognized from video.

1 Introduction

Automatic LPR is a mature technology with a wide range of applications. The commercial LPR systems rely on specialized cameras designed specifically for the task as the quality of captured images largely determines the overall performance. In this paper we address a different and under-explored scenario when the input of the LPR system is a low resolution (LR) video e.g. captured by a common camera or a mobile phone.

We propose a CNN with a novel architecture, denoted as LprNet, which accepts a sequence of geometrically registered LP images obtained from a tracker and outputs distribution over a set of strings with an admissible length. The LprNet architecture has three components: i) a CNN extracting features from each image in the sequence, ii) an aggregation layer shrinking the feature sequence into a single vector and iii) another CNN converting the output of the aggregation layer into a distribution over strings. The aggregation layer allows the training and the testing sequences to have a different number of images. We compared the LprNet with a common approach based on recognizing the strings in each image of the sequence independently and then aggregating the individual predictions to a single hypothesis. Evaluation on 31k low-resolution image sequences shows that the LprNet significantly outperforms both the baseline approach and human performance by a large margin. We

also demonstrate that the LprNet performance monotonically improves with the number of images in the sequence unlike the baseline approach.

Our second contribution is a CNN based generator of super-resolution LP images. The generator has two inputs: i) a string recognized by the LprNet from image sequence and ii) a single LR image. The generator outputs HR image depicting LP with the desired string and closely matching the structure the LR input like pose, background, lighting conditions, etc.

The paper is organized as follows. A brief summary of the relevant prior work is given in Section 2. The proposed LprNet architecture is a subject of Section 3. Section 4 describes the proposed architecture generating super-resolution LP images. The empirical evaluation is presented in Section 5 and Section 6 concludes the paper.

2 Previous work

The classical approach to LPR involves three stages: detection of LP region, segmentation of the characters and recognition of each character separately. A survey of works implementing the classical approach can be found e.g. in [1, 4]. Here we briefly review only the recent works solving the LPR problem in end-to-end fashion by a deep neural network trained from examples. In [12] a sequence of features is extracted by a CNN sliding across the input image depicting geometrically registered LP. Then a Recurrent Neural Network [8] is used to label the sequential features. Finally, the Connectionist Temporal Classification layer [6] converts the label sequence into a character string. The approach of [10] uses a CNN with the last layer representing N different classifiers each predicting particular character in the string. The classifier outputs either the character class or "NULL" symbol if the character is not present which allows to model LP with different number of characters. The model is made spatially invariant by placing the Spatial Transformer module [9] before the CNN. The method of [13] solves both the license plate detection and recognition simultaneously by a single CNN. The work of [18] considers LPR from images captured by a moving camera. Their approach uses a architecture similar to [12]. However, the CNN is trained from synthetically generated training examples without manual annotation. The examples are generated from a grammar model producing noise-free image. The Cycle-GAN [21] is used to translate the synthetic graphics to real looking photos.

Most existing LPR systems, like those listed above, process still images. To our best knowledge the CNN based method for LPR from video sequences has not been published yet. Hence we briefly review only non-CNN works. There are two main approaches to exploit the video sequences. The first approach is based on using a super-resolution reconstruction [3, 15, 17, 20] to create a single high-quality image which is then passed to the still-image LPR recognizer. The second approach is based on recognizing LP strings in each image of the sequence independently and aggregating the predicted strings to a single hypothesis. For example [2] recognizes each frame by SVM classifier and then applies simple majority voting on each position in the character string. The approach of [16] is based on first registering the independently recognized character strings by the Levenstein distance and then averaging posterior distribution of the registered characters.

The architecture proposed in our paper extends the CNN for number recognition of street view images [5]. Namely, we employ the same last layer for representing distribution over character strings of variable length. Our network uses a layer aggregating sequence of features which are extracted from the individual images in the sequence. It differs from [2, 16] who directly aggregate either the predicted characters or their distribution. The CNN ar-

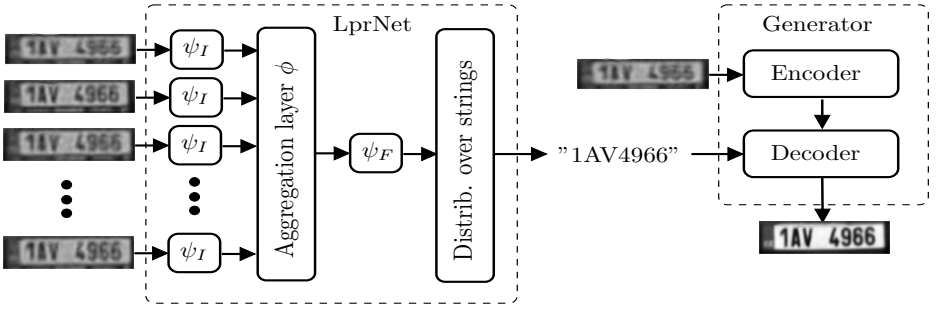

Figure 1: Architecture of the proposed LprNet and the super-resolution generator.

chitecture proposed in this paper uses averaging and max-pool aggregation layers to obtain fixed size representation from a sequence of features extracted from video frames. A similar CNN architecture with max-pool aggregation was proposed [19] for 3D shape analysis from multiple views. The CNN architecture with the averaging aggregation was used in [7] to learn patch similarity for depth estimation.

3 LPR from image sequences

3.1 Proposed LPR-NET

LP image $x \in \mathcal{X}$ depicts a string $(c_1,\ldots,c_L) \in \mathcal{C}^L$ of $L \in \mathcal{L}$ characters. We use $\mathcal{X}$ to denote a set of admissible input images, $\mathcal{L} = \{L_{min},\ldots,L_{max}\}$ is a set of admissible lengths of the strings and $\mathcal{C} = \{"0",\ldots,"9","A",\ldots,"Z",\ldots\}$ is an alphabet of characters. Let further $\mathcal{C}^* = \cup_{L\in\mathcal{L}}\mathcal{C}^L$ denote a set of all strings composed of characters from the alphabet $\mathcal{C}$ which have an admissible length $L \in \mathcal{L}$. Let $\mathcal{X}^*$ be a power set of $\mathcal{X}$.

Our goal is to design a predictor $h\colon \mathcal{X}^* \to \mathcal{C}^*$ which accepts an image sequence $\overline{x} = (x_1,\ldots,x_N) \in \mathcal{X}^*$ and returns a string $\overline{c} = (c_1,\ldots,c_L) \in \mathcal{C}^*$. The elements of the sequence $\overline{x}$ are geometrically registered LP images obtained by tracking the plate in a video, i.e. the images depict the same string $\overline{c} \in \mathcal{C}^*$. Note that the string length is unknown and has to be predicted as well. Given a sequence $\overline{x} \in \mathcal{X}^*$, we predict the character string by the MAP rule

$$h(\overline{x}) = \operatorname*{argmax}_{\overline{c}\in\mathcal{C}^*} p(\overline{c} \mid \overline{x};\theta) \tag{1}$$

$$p(\overline{c} \mid \overline{x};\theta) = p(L \mid \overline{x};\theta) \prod_{i=1}^{L} p_i(c_i \mid \overline{x};\theta) \tag{2}$$

where $p(L \mid \overline{x};\theta)$ is the probability that the string length is L and $p_i(c \mid \overline{x};\theta)$ is the probability that character c is at the i-th position of the string. We use the following parametric models

$$p(L \mid \overline{x};\theta) \propto \exp\langle w_L, \psi(\overline{x})\rangle \quad \text{and} \quad p_i(c \mid \overline{x}) \propto \exp\langle w_{i,c}, \psi(\overline{x})\rangle \;, i \in \{1,\ldots,L_{max}\}\,, \tag{3}$$

where $\psi\colon \mathcal{X}^* \to \mathbb{R}^D$ is a function extracting features from the image sequence and $w_L \in \mathbb{R}^D$, $L \in \mathcal{L}$, $w_{i,c} \in \mathbb{R}^D$, $i \in \{1,\ldots,L_{max}\}, c \in \mathcal{C}$, are parameters. The feature extractor ψ is a CNN composed of three parts (see Fig 1)

$$\psi(\overline{x}) = \psi_F(\phi(\psi_I(x_1),\ldots,\psi_I(x_N))) \tag{4}$$

where $\psi_I \colon \mathcal{X} \to \mathbb{R}^K$ and $\psi_F \colon \mathbb{R}^K \to \mathbb{R}^D$ are CNNs with a chain architecture composed of convolution, max-pooling, fully-connected and ReLU layers. The function $\phi \colon \mathbb{R}^{K \times N} \to \mathbb{R}^K$ is an aggregation layer converting a sequence of N K-dimensional vectors to a single K-dimensional vector. We consider two different aggregation layers performing element wise averaging and maximization, respectively, i.e.,

$$\phi_{avg}(\mathbf{F}) = \begin{bmatrix} \frac{1}{N}\sum_{n=1}^{N} F_{1,n} \\ \vdots \\ \frac{1}{N}\sum_{n=1}^{N} F_{K,n} \end{bmatrix} \quad \text{and} \quad \phi_{max}(\mathbf{F}) = \begin{bmatrix} \max_{n\in\{1,\ldots,N\}} F_{1,n} \\ \vdots \\ \max_{n\in\{1,\ldots,N\}} F_{K,n} \end{bmatrix}. \tag{5}$$

Let all model parameters, i.e. the vectors $w_L, w_{i,c}$ and the convolution filters of the CNNs ψ_I and ψ_F, be encapsulated in θ. We learn θ by maximizing the log-likelihood

$$L(\theta) = \sum_{j=1}^{m} \left(\log p(L^j \mid \bar{x}^j; \theta) + \sum_{i=1}^{L^j} \log p_i(c_i^j \mid \bar{x}^j; \theta) \right) \tag{6}$$

defined on a training set $\{(\bar{x}^j, \bar{c}^j) \in (\mathcal{X}^* \times \mathcal{C}^*) \mid j = 1, \ldots, m\}$. The training set is composed of m image sequences $\bar{x}^j = (x_1^j, \ldots, x_N^j)$ each annotated by a string $\bar{c}^j = (c_1^j, \ldots, c_{L^j}^j)$ depicted on the images. We maximize the log-likelihood (6) by ADAM [11].

The distribution of variable length character strings, equation (2), is adopted from [5]. In case the video sequence $\bar{x}$ is a single image, $N = 1$, our architecture becomes equivalent to the chain CNN proposed in [5] for recognition of house numbers from still images.

3.2 Baseline: aggregation of independent predictions

In this section we describe a baseline approach that has been used for LPR from video sequences [2, 16]. The approach is based on predicting the character string from each image in the sequence independently and aggregating the individual predictions to a final one.

Let $(x_1, \ldots, x_N) \in \mathcal{X}^N$ be a sequence of LP images and let $q^n(y)$ denote a distribution over a generic label $y \in \mathcal{Y} = \{1, \ldots, Y\}$ which is extracted from the n-th image. By $g \colon \mathbb{R}^{Y \times N} \to \mathcal{Y}$ we denote an aggregation strategy used to predict a single label based on the sequence of distributions $Q = (q^1(y), \ldots, q^N(y))$. The aggregation is applied to prediction of characters at individual positions in which case $\mathcal{Y} = \mathcal{C}$ and $q^n(y)$ equals to $p_i(c \mid x_n)$. Similarly, it can be used to predict the string length in which case $\mathcal{Y} = \mathcal{L}$ and $q^n(y)$ equals to $p(L \mid x_n)$. In our experiments, $p_i(c \mid x_n)$ and $p(L \mid x_n)$ is defined by (3) with the feature descriptor $\psi(x)$ being a CNN with a chain architecture [5]. We consider the following aggregation strategies:

Averaging The label is predicted based on the highest average probability

$$g(Q) = \operatorname*{argmax}_{y \in \mathcal{Y}} \sum_{n=1}^{N} q^n(y). \tag{7}$$

Voting The labels is predicted based on the sum of votes weighted by the label probability

$$g(Q) = \operatorname*{argmax}_{y \in \mathcal{Y}} \sum_{n=1}^{N} q^n(y) [\![y = \operatorname*{argmax}_{y' \in \mathcal{Y}} q^n(y')]\!]. \tag{8}$$

Maximization The label is predicted based on the highest probability

$$g(Q) = \operatorname*{argmax}_{y \in \mathcal{Y}} \max_{n \in \{1,\ldots,N\}} q^n(y). \tag{9}$$

The video-based LPR system using Averaging aggregation was proposed in [16]. A variant of the Voting aggregation was described in [2]. In particular, they consider all votes to have equal weight, i.e. $g(Q) = \operatorname{argmax}_{y\in\mathcal{Y}} \sum_{n=1}^{N} [\![y = \operatorname{argmax}_{y'} q^n(y')]\!]$. However, we found the strategy (8) to work considerably better because the weights resolve decision making in case of even number of votes often occurring when the video sequence is short.

4 Generator of super-resolution LP images

Assume we predicted a string $\bar{c} = h(\bar{x};\theta)$ from a LR image sequence $\bar{x} = (x_1,\ldots,x_N)$ using the approach from Sec 3.1. The task now is to create a HR and well perceivable image of a LP that closely resembles images in the sequence $\bar{x}$ themselves being unreadable. Let x be a single image taken from $\bar{x}$. We compute its HR counterpart $\hat{x}$ by the super-resolution generator (see Fig 1)

$$\hat{x} = d(e(x;\omega_e),\bar{c};\omega_d) \tag{10}$$

where $d\colon \mathbb{R}^Z \times \mathcal{C}^* \to \mathcal{X}$ is a decoder generating the synthetic image and $e\colon \mathcal{X} \to \mathbb{R}^Z$ is an encoder the task of which is to extract a low-dimensional description of x. The descriptor $z = e(x;\omega_e)$ encodes all information not contained in the string $\bar{c}$, e.g. a pose, lighting condition, background color and so on. The decoder $d(\cdot;\omega_d)$ and the encoder $e(\cdot;\omega_e)$ are CNNs whose convolution filters are encapsulated in the vectors ω_d and ω_e, respectively.

In order to measure quality of the images generated by (10) we introduce a discriminator $\ell\colon \mathcal{X}\times\mathcal{C}^* \to [0,1]$ similarly to CGANs [14]. The discriminator's output $\ell(x,\bar{c};\omega_l)$ corresponds to the probability that image x depicts a real LP conditioned the string is $\bar{c}$. The value $1-\ell(x,\bar{c};\omega_l)$ is then the probability that x is synthetically generated. The discriminator itself is a CNN with filters ω_l. Let $\{(x^j,\hat{x}^j,\bar{c}^j) \in \mathcal{X}\times\mathcal{X}\times\mathcal{C}^* \mid j = 1,\ldots,m\}$ be a training set where x^j is the input LR image, $\hat{x}^j$ is a desired HR counterpart of x^j and $\bar{c}^j$ is the string to be depicted on $\hat{x}^j$. To measure the quality of HR images generated by (10) we define a function

$$\begin{aligned} F(\omega_d,\omega_e,\omega_l) \quad = \frac{1}{m}\sum_{j=1}^{m} \Big(& \|\hat{x}^j - d(e(x^j;\omega_e),\bar{c}^j;\omega_d)\|_1 \\ & + \log(1-\ell(d(e(x^j;\omega_e),\bar{c}^j;\omega_d),\bar{c}^j;\omega_l)) + \log \ell(\hat{x}^j,\bar{c}^j;\omega_l) \Big) . \end{aligned} \tag{11}$$

The first term in (11) is L_1 distance between the ground-truth HR image $\hat{x}^j$ and the generated one $d(e(x^j;\omega_e),\bar{c}^j;\omega_d)$. The second and the third term corresponds to the adversarial loss whose value is low if the discriminator cannot distinguish the generated image from the real one. The parameters of the generator are then learned by solving the min-max problem

$$(\omega_d^*,\omega_e^*) \leftarrow \min_{\omega_d,\omega_e} \max_{\omega_l} F(\omega_d,\omega_e,\omega_l) . \tag{12}$$

We solve (12) iteratively by alternating minimization w.r.t (ω_d,ω_e) using ADAM [11] and maximization w.r.t. ω_l using the Stochastic Gradient Descend.

5 Experiments

5.1 Data

For empirical evaluation we use LP images originating from three sources:

Video tracks We captured videos of cars using a common camera and a mobile phone installed on a tripod. The videos were processed by a commercial LP tracker producing sequences of geometrically registered LP images. In total we have 31k sequences with ≈72 images on average. The image resolution in each sequence is either monotonically increasing (car driving towards the cam) or monotonically decreasing (car going away from the cam). The LP strings were manually annotated using the well readable frames with the highest resolution.

Still images We used a proprietary database of 1.4M high-resolution images of geometrically registered LPs. The LPs originate from various European countries. Each image has annotation of the LP string. The annotation is created manually (40%) and automatically (60%). The automatic annotation is done by recognizing the string and the car make and model, both by a trainable CNN, and verifying the recognized entries in the car register.

Synthetic images We have a precise description of LP format of 16 European countries. We used the information to generate a database of 4M synthetic LP images.

The video tracks, still images and synthetic images were used to create data for benchmarking the LprNet and the super-resolution generator as follows:

Image sequences for training and testing the LPR system We randomly selected 5.7k video tracks for testing and 1.5k for validation. The 1.5k validation tracks were split in a sliding window fashion into 79k image sequences each having 5 frames. The validation set serves for hyper-parameter tuning of the CNNs. The remaining 23.8k video tracks were used for training. In addition, the training set was extended by artificially generated image sequences. The first frame of the artificial sequence was taken from the still image or the synthetic image database. The consecutive frames in the sequence were created by applying a distortion transformation on the first frame. The distortion transforms involve application of motion blur, additive Poisson noise, and bilinear down-scaling with randomly selected parameters. In total, the training set contains 8.3M image sequences (1.5M video tracks + 2.8M still images + 4M synthetic images) each having 5 images.

Training examples for the super-resolution generator The triplets $(x^j, \hat{x}^j, \overline{c}^j)$ for training the super-resolution generator were created from the still images and the synthetic images only. The database image distorted by a random affine transform was used as the desired generator's output $\hat{x}^j$. The corresponding input image x^j was obtained by applying a distortion transform (the same as described above) on $\hat{x}^j$. Altogether we use 1.6M training triplets (1.4M still images + 0.2M synthetic images).

Human annotation We randomly selected 500 images from the video tracks. The distribution of the horizontal resolution of the selected images is uniform on the range from 60 to 100 pixels. We asked 7 financially motivated annotators to estimate the LP string on each of the 500 images. Note that the ground-truth annotation is also created manually, however, using the well readable high-resolution images (>120px) of each video track.

5.2 Implementation

We implemented the proposed CNN for LP recognition based on image sequences as described in Sec 3.1. The variants using averaging ϕ_{avg} and maximization ϕ_{max} aggregation

layer are further denoted by LprNet-Avg(N) and LprNet-Max(N), respectively. The argument N denotes the number of images in the input sequence. Both variants were trained on 8.3M annotated image sequences each having 5 frames. Note that the number of images in the test sequences can be arbitrary (i.e. not just 5) thanks to the aggregation layer.

As the first baseline, denoted as SfCnn, we use a CNN recognizing the LP from a single frame. The SfCnn is trained from the same examples as LprNet. The architectures of the SfCnn, LprNet-Avg and LprNet-Max are exactly the same up to the aggregation layer. In case of SfCnn the aggregation layer is just the identity function, i.e. the layers of SfCnn form a chain like in [5].

As the next baseline we applied SfCnn on images in the sequence independently and aggregated the individual predictions by the strategies described in Sec 3.2. The baselines are denoted as SfCnn+Avg, SfCnn+Voting and SfCnn+Max corresponding to the averaging (7), voting (8) and maximization (9) aggregation strategy, respectively. Note that the original works [2, 16] use the SVM classifier as the method of single-image classification. Replacing SVM by SfCnn allows us to fairly measure the effect of different aggregation strategies.

All evaluated methods use a CNN (including the super-resolution CNN generator) processing gray-scale images normalized to resolution 128×32 pixels, i.e. $\mathcal{X} = \mathbb{R}^{32\times128}$. The alphabet $\mathcal{L}$ has 48 characters. The LP string length varies from $L_{min} = 5$ to $L_{max} = 9$.

5.3 Results

Accuracy versus number of frames We split the test image sequences into two subsets. First, a low-resolution subset, containing 2,318 sequences in which the widest LP image has the horizontal resolution 65 pixels. Second, a higher-resolution subset, containing 1,751 sequences with the widest image resolution 105 pixels. We varied the number of images in the sequence from 2 to 20 and for each setting we computed the accuracy of the competing methods. The accuracy is a fraction of test sequences for which all characters in the string and its length are predicted correctly. We performed the experiment twice: image resolution in the sequence is increasing (car goes towards cam) and decreasing (car goes away).

The results are summarized in Figure 2. The LprNet-Avg consistently outperforms LprNet-Max, and both methods are better than the baselines SfCnn+Avg/Max/Voting by a large margin. The improvement of LprNet-Avg/Max over the baselines is more significant on LR sequences. The ordering of image resolution matters. In case of increasing resolution the accuracy of all methods monotonically improves w.r.t. the number of images. In the opposite case accuracy of baselines starts to deteriorate when the number of images is more than 6, and this effect is more pronounced on LR sequences. The prediction of later added LR images is more likely to be erroneous which spoils the aggregation. In contrast the accuracy of LprNet-Avg/Max only stagnates but is not decreasing.

Accuracy versus image resolution We compare the LprNet-Avg, performing best according to the comparison in Sec 5.3, against human performance and the baseline SfCnn. The results are summarized in Tab 1. For humans we report the average accuracy and the top 7 accuracy. The average accuracy is the fraction of all $500 \times 7 = 3.5$k human predictions that are correctly predicted. The top 7 accuracy is the fraction of 500 test images for which the correct string is predicted at least by one of the 7 human annotators.

We further split the test sequences into groups according to the horizontal resolution of the widest image in the sequence. We then evaluated the prediction accuracy separately on sequences with the given resolution. The results are shown in Fig 3. It is seen that the

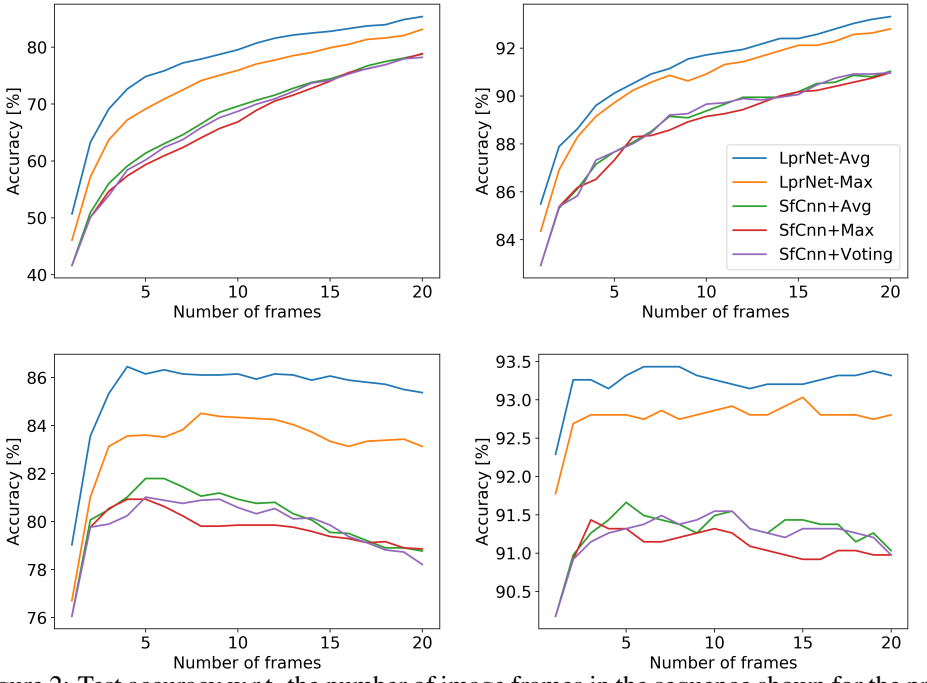

Figure 2: Test accuracy w.r.t. the number of image frames in the sequence shown for the proposed LprCnn-Avg/Max and the baselines SfCnn-Avg/Max/Voting. The left column shows results on low-resolution sequences and the right column on higher-resolution ones. The top row is for sequences with increasing image resolution and the bottom for the decreasing.

LprNet-Avg significantly outperforms both the baseline SfCnn and the human performance. The difference is most significant on the lowest resolution images. In particular, while 60px wide LPs are unreadable for humans (top-7 accuracy below 10%) the accuracy of LprNet-Avg using sequences with 10 images reaches 80%. The top-7 human accuracy approaches the computer performance at resolution ≈100px. However, the average human performance at 100px resolution is still only 60% while the LprNet-Avg(10) reaches 95%.

Method	**Acc** [%]
LprNet-Avg(10)	90.6
LprNet-Avg(5)	88.9
LprNet-Avg(2)	85.6
SfCnn	78.1
Human-top7	64.4
Human-avg	39.3

Table 1: Accuracy of the proposed LprNet-Avg(N) (N is the number of images in the sequence), the baseline SfCnn using a single image and the human accuracy.

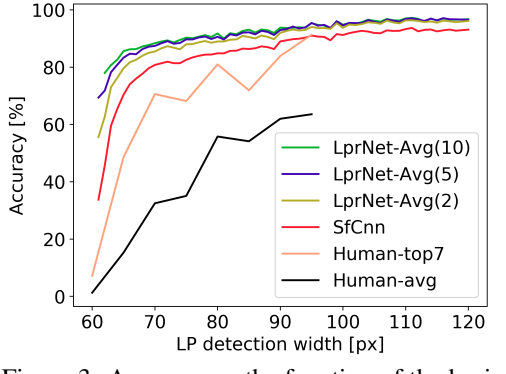

Figure 3: Accuracy as the function of the horizontal resolution of the LP image.

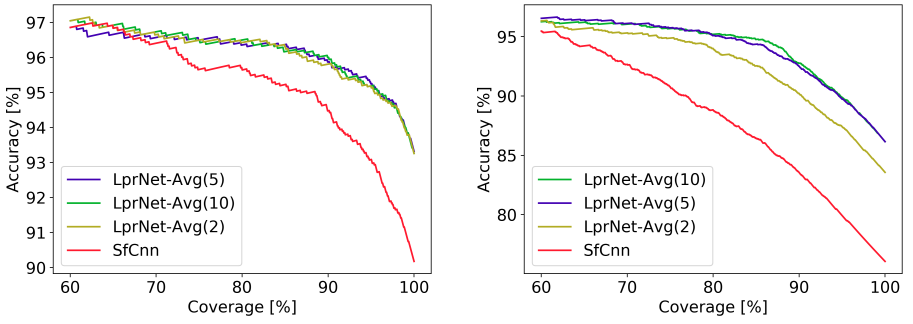

Figure 4: Accuracy vs. coverage for the proposed LprNet-Avg(N) and the baseline SfCnn when both methods are extended to have the reject option. The results are shown separately on the high-resolution sequences (left) and the low-resolution sequences (right).

Prediction with reject option The probabilistic output of the LprNet allows to extend the predictor by the reject option, i.e. $h(\overline{x})$ rejects to decide if $\max_{\overline{c}\in\mathcal{C}^*} p(\overline{c} \mid \overline{x};\theta) < \alpha$ where $\alpha \in [0,1]$ is a desired minimal confidence. We evaluated the prediction with the reject on low-resolution and high-resolution sequences separately. We compared the proposed LprNet-Avg and the baseline SfCnn. Recall that SfCnn provides the same probabilistic output. We varied the value of α and for each setting we computed the coverage and accuracy. The coverage is the fraction of sequences for which the predictor does not reject to decide. The results are summarized in Fig 4. The LprNet-Avg has consistently higher coverage for a fixed accuracy as compared to the baseline SfCnn that uses only a single image. As expected the coverage increases with the number of images in the sequence.

Generator of super-resolution LP images Examples of super-resolution images created by the proposed CNN generator are shown in Fig 5. Please read the caption for more details.

6 Conclusions

We have proposed end-to-end CNN architecture, called LprNet, recognizing character string from a sequence of geometrically registered images. Empirical evaluation on LR videos shows that the LprNet significantly outperforms both baseline methods and humans. E.g. the human prediction accuracy on LP images at 60px resolution is below 10% while accuracy of the LprNet reaches 80%. The performance of the LprNet improves monotonically with the number of images in the sequence in contrast to the baselines. Our second contribution is a CNN based super-resolution generator converting LR images into their HR counterparts closely matching the structured of the input. In contrast to previous works, our generator allows to explicitly control the content of the generated image, namely, the depicted string.

Acknowledgments

VF was supported by Czech Science Foundation grants 16-05872S. VV and MU were supported by the Technology Agency of the Czech Republic project TE01020415.

Figure 5: The first and the fourth columns show a sample of input images taken from the first frame of LR (60px) test sequences. The red strings denote the ground-truth. Other columns are super-resolution images created by the proposed CNN generator. The input of the generator was the corresponding LR image and the black strings shown above. The black strings were predicted from test sequences using LprNet-Avg(5) and they correspond to the MAP estimate and the second most probable hypothesis. The samples for which the MAP estimate of the string is correct are above the dash line. The samples below the line show errors. Note that the generated super-resolution images preserve structure of the LP inputs.

References

[1] C.N. Anagnostopoulos, I.E. Anagnostopoulos, and I.D. Psoroulas. License plate recognition from still images and video sequences: A survey. *IEEE Trans. on Intelligent Transportation Systems*, 9(3):377–391, 2008.

[2] Clemens Arth, Florian Limberger, and Horst Bischof. Real-time license plate recognition on embedded dsp-platform. In *Proc. of Conf. on Computer Vision and Pattern Recognition*, 2007.

[3] L. Dlagnekov. Recognizing cars. Technical Report CS2005-0833, Depart. Comp. Science Eng. University California, 2005.

[4] Shan Du, Mahmoud Ibrahim, and Mohamed Shehata. Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(2):211–325, 2012.

[5] Iam Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sasha Arnoud, and Vinay Shet. Multi-digit number recognition from stree view images using deep convolutional neural networs. In *ICLRP*, 2014.

[6] Alex Graves, Santiago Fernandez, Faustino Gomez, and Jurgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurent neural networks. In *Proc. of International Conference on Machine Learning*, 2006.

[7] W. Hartmann, S. Galliani, M. Havlena, L.V. Gool, and K. Schindler. Learned multi-patch similarity. In *ICCV*, 2017.

[8] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–80, 1997.

[9] Max Jaderberg, karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proc. of Neural Information Processing Systems*, 2015.

[10] Vishal Jain, Zitha Sasindran, Anoop Rajagopal, Soma Biswas, Bharadwaj Harish S, and K.R. Ramakrishnan. Deep automatic license plate recognition system. In *Proc. of 10'th Indian Conference on Computer Vision, Graphics and Image Processing*, 2016.

[11] D.P. Kingma and J.L. Ba. ADAM: A method for stochastic optimization. In *ICLR*, 2014.

[12] Hui Li and Chunhua Shen. Reading car license plates using deep convolutional neural networks and lstms. Arxiv, 2016.

[13] Hui Li, Peng Wang, and Shunhua Shen. Towards end-to-end car license plates detection and recognition with deep neural networks. arxiv, 2017.

[14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. Arxiv, 2014.

[15] K.V. Suresh, Maheres Kumar, and A.N. Rajagopalan. Superresolution of license plates in real traffic videos. *IEEE Trans. on Intelligent Transportation Systems*, 8(2):321–331, 2007.

[16] Nicolas Thome, Antoine Vacavant, Lionel Robinault, and Serge Migue. A cognitive and video-based approach for multinational license plate recognition. *Machine Vision and Applications*, 22(2):389–407, 2011.

[17] Patrick Vandewalle, Luciano Sbaiz, and oos Vandewalle. Super-resolution from unregistered and totally aliased signals using subspace methods. *IEEE Transactions on Signal Processing*, 55(7):3687 – 3703, 2017.

[18] Xinlong Wang, Zhipeng Man, Mingyu You, and Chunhua Shen. Adversarial generation of training examples: Applications to moving vehicle license plate recognition. arxiv, 2017.

[19] O. Wiles and A. Zisserman. SilNet: Single- and multi-view reconstruction by learning silhouettes. In *BMVC*, 2017.

[20] Jie Yuan, Si-Dan Du, and Xiang Zhu. Fast super-resolution for license plate image reconstruction. In *Proc of Int. Conference on Pattern Recognition*, 2008.

[21] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. arxiv, 2017.