Learning to Decipher License Plates in Severely Degraded Images *

Paula Kaiser[1,2], Franziska Schirrmacher[1,2][0000-0003-1511-7669], Benedikt Lorch[1][0000-0002-7843-4656], and Christian Riess[1][0000-0002-5556-5338]

[1] IT Security Infrastructures Lab, Computer Science, Friedrich-Alexander University Erlangen-Nürnberg
[2] Both authors contributed equally to this work.
franziska.schirrmacher@fau.de

Abstract. License plate recognition is instrumental in many forensic investigations involving organized crime and gang crime, burglaries and trafficking of illicit goods or persons. After an incident, recordings are collected by police officers from cameras in-the-wild at gas stations or public facilities. In such an uncontrolled environment, a generally low image quality and strong compression oftentimes make it impossible to read license plates. Recent works showed that characters from US license plates can be reconstructed from noisy, low resolution pictures using convolutional neural networks (CNN). However, these studies do not involve compression, which is arguably the most prevalent image degradation in real investigations.
In this paper, we present work toward closing this gap and investigate the impact of JPEG compression on license plate recognition from strongly degraded images. We show the efficacy of the CNN on a real-world dataset of Czech license plates.
Using only synthetic data for training, we show that license plates with a width larger than 30 pixels, an SNR above -3 dB, and a JPEG quality factor down to 15 can at least partially be reconstructed. Additional analyses investigate the influence of the position of the character in the license plate and the similarity of characters.

Keywords: License plate recognition · Deep learning · Compression

1 Introduction

Forensic investigations aim to reveal the identity of a suspect. This frequently involves the analysis of a surveillance video or picture of the suspect's vehicle. The vehicle license plate is oftentimes instrumental in the identification of suspects. However, investigators collect such images or videos from cameras

* We gratefully acknowledge support by the German Federal Ministry of Education and Research (BMBF) under Grant No. 13N15319, the German Research Foundation GRK Cybercrime (393541319/GRK2475/1-2019), and the German Research Foundation (146371743/TRR 89)

in-the-wild, which are in many cases low-cost, poorly maintained devices. The types of practically seen image degradations vary greatly. Additionally, image and video compression leads to an inherent loss of information, and the spectrum of compression algorithm further complicates the analysis. Furthermore, the resolution of license plates is oftentimes very low. In summary, the resulting license plates are in many cases not readable by a forensic investigator. In those cases, computational methods to partially or fully reconstruct a license plate are necessary.

Early methods for automatic license plate recognition operate with optical character recognition. The recognition rates of these systems considerably improved with the advent of deep neural networks [22, 21, 23]. As of today, deciphering license plates from good-quality images of controlled acquisition setups is well understood and implemented on many highways [3]. Deciphering low-quality images also progressed with the advent of neural networks, and by performing character segmentation and recognition within the same stage [22].

However, severely degraded images, which humans cannot read, are still an open challenge. Agarwal *et al.* showed that even such highly degraded images still contain useful information [2]. Lorch *et al.* extended this work by deciphering synthetic images of low-resolution, noisy US license plates [15]. While these first results are encouraging, they still do not suffice to operate on real images. As shown by recent works [3], additional sources of degradation can significantly hamper the recognition, particularly for images of low resolution, most notably the impact of lossy compression, which is present in virtually all recordings in the wild. In this work, we close this gap by studying the impact of compression in conjunction with low resolution and additional noise. Our study is performed on real and synthetic images of European license plates, more specifically on Czech license plates.

The contributions of this paper are three-fold:

1. We demonstrate the applicability of the CNN on real images of Czech license plates.
2. The influence of compression on the recognition rate is evaluated on synthetic Czech license plates.
3. We provide an in-depth analysis of the influence of similarity and position of characters.

The paper is organized as follows: Section 2 provides an overview of existing license plate recognition methods. Section 3 presents the network architecture. Section 4 reports the experiments and discusses the results. Section 5 concludes the paper.

2 Related Work

Automatic license plate recognition is a thoroughly studied research topic. In general, the techniques to extract the license plate from the image can be divided into pipeline-based recognition and learning-based approaches. However, the capability of these methods to operate on low-quality images varies considerably and will be discussed below.

2.1 Pipeline-based recognition

Pipeline-based license plate recognition commonly consists of three stages: First, license plate extraction, which locates and crops the license plate. Second, character segmentation, which splits the license plate into images of the individual characters. Third, character recognition, which labels the images of the individual characters.

1. Extraction Given an image of a car, the goal of the extraction stage is to locate and cut out the license plate [3]. The inspector can manually locate the license plate and determine the coordinates, since the number of images to analyze is typically small in forensic investigations. To obtain a perpendicular view of the license plate, the image is rectified [5].

2. Segmentation Projection profiles are a widely used feature to separate the characters. However, these features are sensitive to rotation and noise, and the number of license plate characters needs to be known beforehand. On the other hand, the computation is fast and independent of the position of the characters [4]. Liu *et al.* exploit additional prior knowledge about Chinese license plates for a successful segmentation. This method is very simple and straightforward, but is inherently limited by specific assumptions on Chinese license plates [13]. Overall, segmentation-based character recognition critically depends on a correct character segmentation. Especially in low-quality images, a good segmentation is oftentimes difficult to achieve.

3. Recognition Arguably the simplest recognition method is template matching. Template matching works well for single-font, non-rotated, non-broken, and fixed-size characters. Rotated characters can also be well recognized when templates under different inclination angles are used. However, this results in an increased computation time [4]. However, feature extraction is more robust than template matching. Wen *et al.* used direction features based on the connected relationship and direction strokes [24]. Other possible features are gradients [14] and local binary patterns [12, 3]. Hsieh *et al.* use a sliding window algorithm to avoid separating the characters [17]. However, this method requires explicit knowledge of font style and size, as well as precise character placement.

2.2 Learning-based approaches

Deep neural networks were successfully used for segmentation-free character recognition in more general environments [8, 9]. Li *et al.* adapted this approach to decipher license plates [11]. In this work, a combination of a convolutional neural network (CNN) and a bidirectional recurrent neural network (BRNN) with long

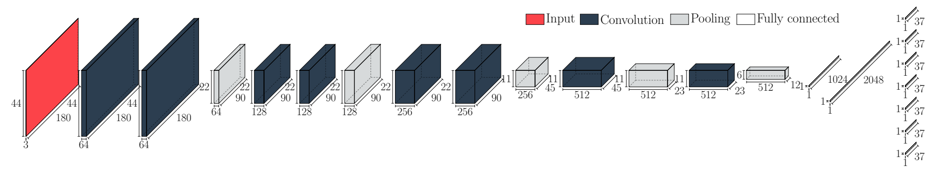


Fig. 1. Proposed architecture for severely degraded Czech license plates with seven characters.

short-term memory (LSTM) extracts features from the image. The connectionist temporal classification (CTC) converts the features to the final recognition result, namely the license number. The authors show that the method is able to recognize slightly distorted images, but low-quality data has not been studied. Following up on this work, several conceptually similar approaches have been developed [10, 19, 23].

A general recognition framework that is suitable for real-time applications is You Only Look Once (YOLO) [18]. Redmon *et al.* proposed this method to detect and classify different objects in real-time. Silva *et al.* modified YOLO to recognize letters and digits to fit their data. They demonstrate its use on Brazilian license plates [20] as well as license plates from other countries [21]. Abdullah *et al.* showed that YOLO can be retrained without architectural modification to recognize digits and detect Bangla characters in license plates [1].

Gonçalves *et al.* proposed the use of deep multi-tasking learning. Each task classifies one character of the license plate from the entire image. This method is suitable for real-time applications and recognizes multiple license plate characters in the same frame [7]. Špaňhel *et al.* [22] proposed a CNN that can predict eight characters of license plate including a blank fill character for shorter license plate numbers.

In forensic investigations, speed is oftentimes less important than accuracy when deciphering non-human-readable license plates. Agarwal *et al.* showed that useful information is present even in highly degraded images, distorted by noise and low resolution [2]. Two separate CNNs decipher three characters of the license plate each. Their network significantly outperformed human observers on synthetic US data. Lorch *et al.* further improved their method [15]. They deciphered all characters at once and introduced a null character to recognize license numbers of variable length.

In police investigations, officers oftentimes rely on data that was captured by low-cost surveillance cameras. Here, strong compression leads to a decrease in the image quality and a possible loss of information. So far, compression is not covered in the related work. This paper aims towards closing this gap. We examine the performance of a CNN deciphering severely degraded license plates on images with strong lossy compression. We also empirically determine a lower bound for deciphering at least some characters of a license plate in very strongly compressed images.

3 Methods

In this section, we describe the network architecture, training parameters, and the composition of the training data.

3.1 Network architecture

We adapt the CNN presented by Lorch *et al.* [15] to analyze the impact of the compression on license plate recognition. Figure 1 shows an overview of the used network. Compared to Lorch *et al.* [15], the input layer (red) of the network is adapted to 44×180 to better match the aspect ratio of European license plates. The remaining design decisions are identical, and briefly summarized below.

Eight convolutional layers (blue) extract features from the input image with kernel size 3×3, a stride of one and zero-padding for persisting spatial size. Pooling layers (gray) with kernel size 2×2 are inserted after the second, fourth, sixth, seventh and eighth convolutional layer. The first and third pooling layer cut the spatial size in half with a stride of two, while the second pooling layer does not affect the size by using a stride of one. The fourth pooling layer uses a stride of two horizontally and a stride of one vertically. The last pooling layer reduces the spatial size from 11×23 to 6×12 with a stride of two using zero-padding both on the right and on the bottom.

The resulting feature volume of size $6 \times 12 \times 512$ is flattened to a feature vector. This vector is passed to two consecutive fully-connected layers with 1 024 and 2 048 neurons. The fully-connected layers are followed by seven output layers, one for every character in the license plate. Each output layer consists of 37 units to represent 26 letters of the Latin alphabet, the digits from 0 to 9, and one null character. The null character is used for license plates with less than seven characters. The results of the output layers are passed to a softmax function, which normalizes the scores for each character to a probability distribution.

All convolutional layers and the first two fully-connected layers use the ReLU activation function. The weights of the convolutional layers and the output layers are initialized with the Xavier initialization [6]. The first two fully-connected layers are initialized with a truncated normal distribution with zero-mean and a standard derivation of 0.005. The biases of the output layers are initialized with 0, the other biases with 0.1.

The parameters are updated using mini-batch gradient descent with a batch size of 32. The magnitude of those updates is defined by the learning rate, which starts at 0.005 and decays stepwise exponentially. The decay step is the number of batches in one epoch and the decay rate is 0.9. Overfitting is reduced by using dropout in the two fully-connected layers and the output layers with a probability $p = 0.5$. The training is stopped if the validation accuracy does not change for 100 000 training iterations.

3.2 Synthetic Training Data

In order to obtain a high prediction rate, the network needs to be trained with a large number of training examples. However, obtaining many real-world images is

Fig. 2. The pipeline for generating license plates consists of five steps. First, the frame and font are printed and cropped. Then the image is degraded by downsampling, additive noise, and compression.

expensive. A good alternative is generating images of license plates synthetically. The proposed dataset is created similarly to the procedure described by Lorch *et al.* [15] and includes 10 million training, 2 000 validation and 750 000 test images.

Czech license plates are composed of seven characters with fixed positions. Exceptions to this are only historic vehicles and personalized plates, which are rare and will not be considered in this work. Czech regulations specify the font, the font size and measurements like gap sizes and offsets [16]. In this work, we only consider regular Czech license plates, which have the following constraints: At the first position only the digits one to nine are possible. However, in the real world there is a very low probability that a character is present in the first position. The second position represents the region, where the license plate is issued. Letters related to a region are A,B,C,E,H,J,K,L,M,P,S,T,U, and Z. The number of license plates per region is not considered. The third position does not have additional constraints, whereas at positions four to seven only digits are possible. Since the characters G, O, Q, and W are not used at all in Czech license plates, they do not appear in the training data. Grayscale images are used, since there is little additional information in the color channels.

Figure 2 provides an overview of the pipeline to generate the synthetic license plates. First, font and frame are printed to an image with random background. For every image a dark font on a bright background or a bright font on a dark background is randomly chosen. The contrast between fore- and background is randomly chosen as well. After tightly cropping the plate, image degradation is performed. Common forms of degradation, which notably lower the image quality are low resolution, noise, and compression.

Low resolution Low-cost cameras with small sensors result in low-resolution images. In addition, the license plate usually occupies only a few pixels of the image, and vehicles may pass a camera only at a large distance. Thus, we downsample the generated high-resolution images and consider an effective width of the license plate from 30 to 180 pixels with fixed aspect ratio. For training, a continuous range of license plate widths is used. For the test data, we consider seven levels of resolution (30, 50, 70, 90, 120, 150, and 180 pixels). To obtain a uniform size as input to the CNN, the images are upsampled to a size of 180 x 44 pixels with nearest neighbor interpolation after noise is added and compression is applied.

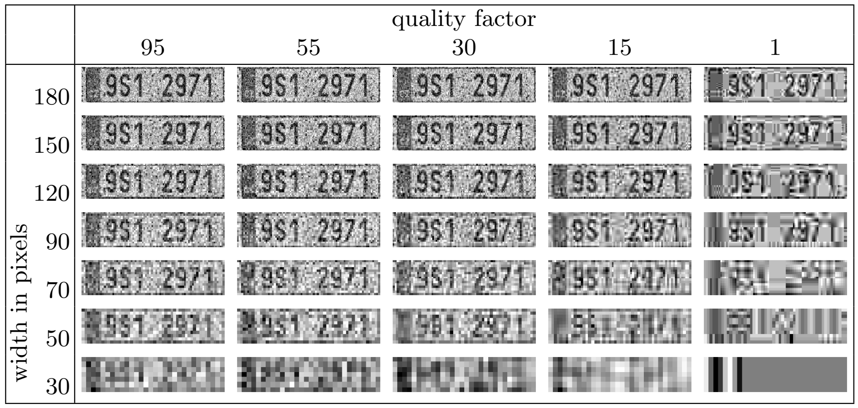


Fig. 3. Example images of the synthetic Czech dataset with an SNR of 3 dB and different sizes and JPEG quality factors. A quality factor of one leads to a huge loss of information, especially in small images.

Noise Noise is a widespread issue in real-world applications. A common assumption is that noise is mostly caused by the camera sensor during the acquisition. Oftentimes, unknown noise models are approximated by additive white Gaussian noise. Here, the noisy image $\boldsymbol{f} = \boldsymbol{s} + \boldsymbol{n}$ is the sum of a noise-free image $\boldsymbol{s}$ and a noise component $\boldsymbol{n}$. The magnitude of the noise is given by the signal-to-noise-ratio (SNR), which can be described with the power of signal and noise:

$$\mathrm{SNR} = \frac{\mathrm{power}(\boldsymbol{s})}{\mathrm{power}(\boldsymbol{n})} = \frac{\sigma_s^2}{\sigma_n^2} \quad , \tag{1}$$

where σ_s^2 denotes the power spectrum of the signal, and σ_n^2 denotes the power spectrum of the noise. For zero-mean noise and signal, the SNR is the fraction of the variances σ^2. In the experiments, we report the noise level in dB, which corresponds to the base-10 logarithm of the SNR. For the training data, the SNR ranges from -3 to 20 dB. For the test data, three levels of SNR (-3, 3 and 20 dB) are used, which corresponds to a severe, a moderate and a low noise level.

Compression Lossy compression is the most effective way of reducing the storage size, and arguably the most common type of image degradation. Such compression particularly reduces the image quality and causes a loss of information. While different compression formats exist, this work adopts JPEG compression, as JPEG is probably the most prominent storage format. The JPEG quality factor allows to trade image quality for file size. The JPEG quality factors reported in this work correspond to the quality argument used in the Python library Pillow. The JPEG quality factor in the training dataset ranges between 1 and 95. For testing, we use five levels of compression (1, 15, 30, 55, and 95).

See Fig. 3 for example images with an SNR of 3 dB and the different sizes and JPEG quality factors used for the test set.

4 Experiments

This section comprises the evaluation of the proposed method. The convolutional neural network (CNN) is trained and tested on different datasets in order to answer two questions: How does distortion, especially compression, influence the recognition? Is the proposed network capable of deciphering severely degraded Czech license plates?

Our evaluation consists of two parts. First, the network performance and generalizability of the training on the synthetic training data is quantitatively evaluated on the real-world dataset. Second, we show on the synthetic dataset an in-depth analysis of the impact of three distortion parameters, namely compression, low resolution, and noise, on the recognition rates.

4.1 Evaluation Metric

Analogously to previous works we report the top-n accuracies as figures of merit. In this metric, the network predictions for each character are sorted by their softmax output. If the correct character is within the n highest likelihoods, we count it as a hit, otherwise as a miss. The reported top-n accuracies are relative hit rates, averaged over all individual characters of all license plates. We specifically report top-1, top-3, and top-5 accuracies.

4.2 Performance on Degraded Real-World Test Data

Dataset Description The real-world Czech test set is used to evaluate the trained network on real data. Špaňhel *et al.* [22] published the dataset "ReId", containing 76 412 real pictures of European license plates. The authors placed Full-HD video cameras in eight different locations and under different conditions. The images are labeled and the plates cropped. The large majority of vehicles are Czech, the remaining ones are registered in other European countries. The images are converted to grayscale and either cropped or resized to size 180 x 44 pixels. The license plates in this dataset are mostly human-readable but of low quality, as seen in Fig. 4 [22]. While Czech license plates are limited to 7 characters, the small subset of non-Czech license plates may also contain an eighth character. To adjust our CNN for this potential failure case in this experiment, we added an eight output unit and trained the CNN with our synthetic 7-character license plates. Thus, it always outputs the empty character at the eighth position. In line with the ground truth annotations by Špaňhel *et al.*, this empty character is counted as hit on Czech 7-character license plates, and as miss on 8-character license plates.

Evaluation Results We compare the proposed network to the CNN presented by Špaňhel *et al.* [22]. The authors report a top-1 accuracy of 98.6% on the ReId dataset. To evaluate how realistic the synthetic dataset is in comparison to the real-world data, we train the proposed CNN on the synthetic training data and evaluate the performance on the real-world test data. The results are

Method	top-1	top-3	top-5
Špaňhel *et al.* [22]	98.6%	-	-
CNN synthetic	89.8%	95.6%	97.3%
CNN fine-tuned	97.3%	98.7%	99.0%

Fig. 4. Top-1, top-3 and top-5 accuracy (left) on the real-world Czech dataset (right) of the proposed network trained on synthetic data and fine-tuned on the real-world training data. The top-1 accuracy is compared to the values reported by [22]. The license plates are mostly human-readable, but distortions such as blur, low contrast, or occlusion can still occur.

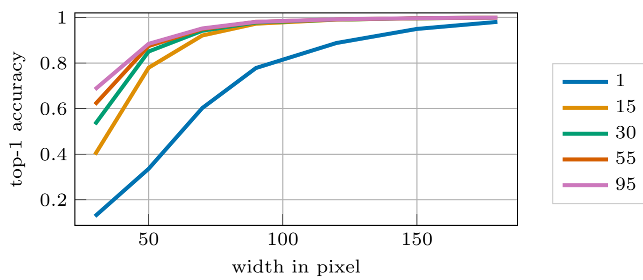
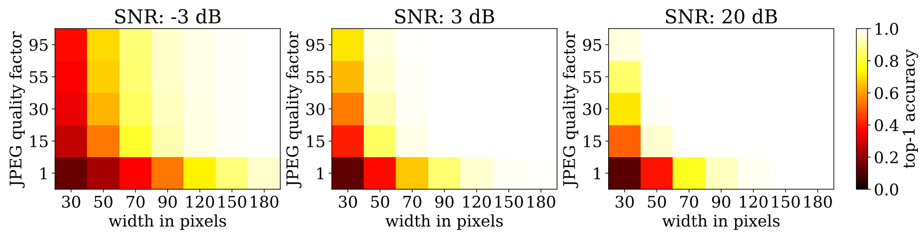


Fig. 5. Influence of the compression rate on the license plate recognition rate. We report the top-1 accuracy of five different compression levels in relation to the width of the license plate in pixels.

shown in the top and middle row of the table in Fig. 4. The proposed CNN achieves a top-1 accuracy of 89.8%. This result is 8.8% lower than the results reported by Špaňhel *et al.*. The performance gap can be attributed to a number of influences that are not modeled by the synthetic dataset. We only model Czech license plates, hence unseen appearances of non-Czech license plates are potential failure cases. Additionally, the real data contains notable amounts of blur and perspective rotation, which is also not modelled in our synthetic data.

To provide a reference on the overall network performance, we additionally fine-tune the network on the real-world training data by Špaňhel *et al.* with a learning rate of 0.005. This allows to fill gaps from training on synthetic data with dataset-specific training data. The results are shown in the bottom row of the table in Fig. 4. With fine-tuning, a top-1 accuracy of 97.3% is achieved, which is comparable to the results reported by Špaňhel *et al.* [22]. Thus, the proposed CNN itself has the capability to model the task of license plate recognition. This experiment shows that the CNN does not model all effects of real-world images when training exclusively on synthetic data. Nevertheless, we argue that the performance is sufficiently high for a further analysis of the impact of individual image degradations on license plate recognition.



SNR [dB]	quality factor	width in pixels 30	50	70	90	120	150	180
-3	95	0.38	0.69	0.86	0.94	0.98	0.99	1.00
	55	0.36	0.67	0.86	0.94	0.97	0.99	1.00
	30	0.33	0.63	0.84	0.94	0.97	0.99	1.00
	15	0.28	0.55	0.79	0.92	0.97	0.99	1.00
	1	0.14	0.23	0.37	0.54	0.73	0.87	0.95
3	95	0.71	0.96	0.99	1.00	1.00	1.00	1.00
	55	0.64	0.95	0.99	1.00	1.00	1.00	1.00
	30	0.55	0.93	0.99	1.00	1.00	1.00	1.00
	15	0.41	0.84	0.97	1.00	1.00	1.00	1.00
	1	0.13	0.37	0.66	0.86	0.95	0.98	1.00
20	95	0.96	1.00	1.00	1.00	1.00	1.00	1.00
	55	0.85	1.00	1.00	1.00	1.00	1.00	1.00
	30	0.71	0.99	1.00	1.00	1.00	1.00	1.00
	15	0.51	0.95	1.00	1.00	1.00	1.00	1.00
	1	0.12	0.39	0.78	0.93	0.98	1.00	1.00

Fig. 6. Top-1 accuracy on the synthetic Czech dataset at different degradation levels (top). Full table of all top-1 accuracies on the synthetic dataset, split by additive noise, JPEG quality factor and the pixel width of the license plate (bottom).

4.3 Experiments on the Impact of Image Degradations

All experiments in this section are performed on the synthetic dataset. The purpose of these controlled experiments is to assess the influence of various image degradations on the recognition rate. We specifically investigate the level of JPEG compression and additive noise, the influence of the character position on the recognition rate, and the similarity between characters. When averaging the recognition performances on the synthetic dataset over all license plates, all characters, all positions, and all degradation levels, then the top-1 accuracy of the CNN is 85.4%, the top-3 accuracy is 93.1%, and the top-5 accuracy is 96.0%.

JPEG Compression versus Pixel Resolution JPEG compression can lead to an inherent loss of information. Hence, it is insightful to investigate a lower bound for reliable character recognition under a given JPEG compression strength. Figure 5 shows the top-1 accuracy for different JPEG quality factors in dependence of the width of the license plate in pixels. The results are averaged over all noise levels.

We can observe that the JPEG compression has a considerable impact on the recognition accuracy for image widths below 70 pixels. That is not surprising, considering that for such small images the license plate is encoded in only very few JPEG blocks. Strong compression removes high-frequency information that

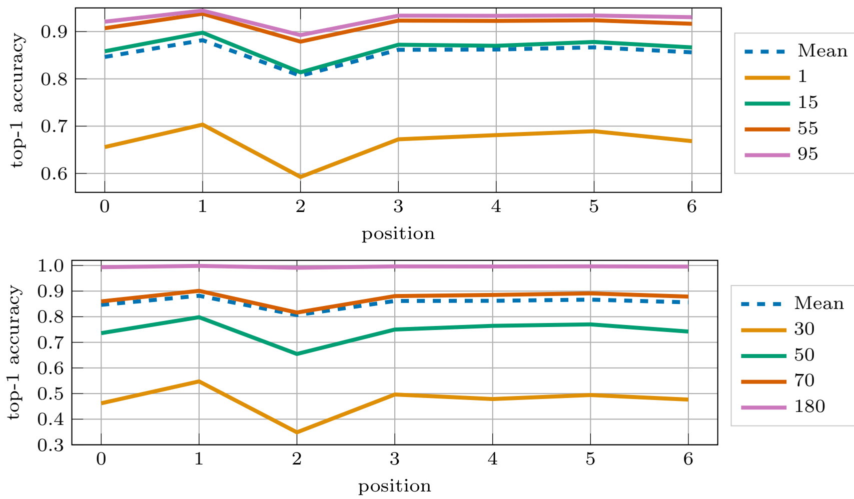


Fig. 7. Top-1 accuracy at all seven positions of the license plate evaluated for different compression rates (top) and widths of the license plate in pixels (bottom). Reported performances are averaged over all remaining degradations.

would be particularly important to distinguish characters at such low resolution. This suggests a trade-off between image resolution and JPEG compression strength. For example, the performances for JPEG qualities 55 and 95 are barely distinguishable for images that are wider than 50 pixels. On the other hand, a (rather extreme) JPEG compression quality of 1, which removes virtually all content within a JPEG block, requires image widths of 90 pixels and beyond to achieve acceptable accuracies. At JPEG quality 1 with the smallest image width of 30 pixels, the CNN performance is effectively at guessing rate.

Impact of Added Noise and Compression Figure 6 illustrates the top-1 accuracy for noise levels of -3 dB, 3 dB, and 20 dB averaged over all license plate positions. Each chart is subdivided by the license plate width and the JPEG quality factor. The exact numerical values corresponding to this experiment are listed in the table below.

The results show that the width of the license plates must be above 30 pixels to obtain reliable results in the presence of other degradations. Images with an SNR of -3 dB or a JPEG quality factor of one can be deciphered only if the remaining degradations are negligible.

Influence of the Character Position The rules for Czech license plates constrain for each position of the license plate the set of admissible characters. In this analysis, we investigate the recognition accuracy per position of the license plate.

Figure 7 shows the top-1 accuracy of the network for different compression rates (top) and different input sizes (bottom), averaged over all remaining distortions. For reference, the mean over all degradations is displayed in both plots.

char	top-1	p1	p2	p3	c1	c2	c3	char	top-1	p1	p2	p3	c1	c2	c3
0	0.86	0	8	6	0.82	0.03	0.03	H	0.86	H	M	U	0.82	0.03	0.03
1	0.89	1	3	7	0.87	0.03	0.02	I	0.83	I	1	T	0.79	0.03	0.03
2	0.88	2	7	3	0.86	0.03	0.03	J	0.91	J	U	Z	0.88	0.02	0.01
3	0.86	3	2	1	0.83	0.03	0.03	K	0.83	K	E	A	0.81	0.03	0.02
4	0.91	4	6	2	0.89	0.02	0.01	L	0.90	L	E	C	0.87	0.03	0.02
5	0.84	5	6	8	0.81	0.04	0.03	M	0.89	M	H	B	0.86	0.04	0.01
6	0.79	6	8	5	0.76	0.08	0.05	N	0.82	N	H	8	0.78	0.03	0.03
7	0.90	7	2	1	0.87	0.04	0.02	P	0.87	P	E	M	0.85	0.02	0.01
8	0.82	8	6	9	0.78	0.06	0.03	R	0.75	R	8	6	0.73	0.03	0.02
9	0.82	9	8	0	0.80	0.05	0.03	S	0.80	S	B	C	0.78	0.02	0.02
A	0.86	A	K	B	0.84	0.01	0.01	T	0.87	T	Z	Y	0.84	0.02	0.02
B	0.79	B	8	H	0.76	0.03	0.06	U	0.83	U	H	J	0.81	0.04	0.02
C	0.84	C	E	L	0.81	0.03	0.02	V	0.83	V	9	8	0.80	0.02	0.02
D	0.72	D	0	U	0.68	0.09	0.03	X	0.78	X	Y	K	0.75	0.02	0.02
E	0.85	E	C	L	0.81	0.03	0.02	Y	0.79	Y	T	1	0.76	0.04	0.03
F	0.81	F	P	E	0.78	0.05	0.04	Z	0.85	Z	2	T	0.83	0.03	0.02

Table 1. Average predictions for the different characters in the test data. The column char shows the characters in the font that is used in Czech license plates. The three highest confidences are shown in the columns c1 to c3, the corresponding characters are shown in the columns p1 to p3.

On top of Fig. 7, the accuracy for the JPEG quality factor 15 is already above the average, which can be attributed to the fact that larger license plates are less susceptible to compression. The higher JPEG quality factors 55 and 95 yield only about 5% higher performances. Qualitatively, the curves have similar shape. A quality factor of 1 leads to a drastic decrease of the top-1 accuracy, since only license plates with a high resolution can still be deciphered. On the bottom of Fig. 7, the image width exhibits a similar dependency as previously observed for JPEG compression, with a major performance drop for license plates with a width of only 30 pixels.

In both plots of Fig. 7, position 3 is particularly difficult to recognize. Here, Czech license plates exhibit the largest variability, as all digits and 22 letters can be used. The best top-1 accuracy is at position two, even though the number of possible characters is smaller for position one. The same effect can also be observed in the last and the second last positions, where only numbers are allowed. This can be attributed to the cropping of the image, since this affects characters at the license plate boundary.

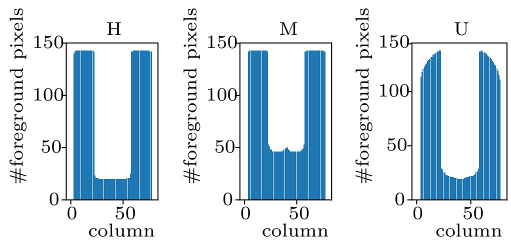


Fig. 8. Similarity of the horizontal projections of the characters H, M and U.

quality factor	char	p1	p2	p3	c1	c2	c3	char	p1	p2	p3	c1	c2	c3
1	C	C	E	U	0.58	0.04	0.04	P	P	E	H	0.62	0.04	0.03
15	C	C	E	Z	0.83	0.03	0.02	P	P	M	F	0.87	0.02	0.01
95	C	C	E	L	0.90	0.02	0.01	P	P	F	E	0.92	0.01	0.01

Table 2. Change of confusion order for different JPEG quality factors.

Similarity of Characters At any position, some characters are more difficult to recognize than others. Table 1 shows for every character the top-1 accuracy (top-1). Additionally, the three characters (predictions p1 to p3) with the highest average confidences (c1 to c3) are displayed. Those values are averaged over all degradation levels and positions. The column "char" shows the character in the font that is used in the license plates. It is important to notice that this table is averaged over all positions. The network learned which characters are possible at a certain position. For example, at positions four to seven, only digits are allowed. At those positions the network does not mistakenly predict a letter, even if they are very similar, like 8 and B. At position three, however, 8 and B are frequently confused. Digits appear at five positions, four of which allow only digits. Therefore, the top-3 predictions for digits only include digits.

Overall, recognition rates of unique characters are higher than recognition rates of groups of similar characters. However, similarity is measured w.r.t. the features extracted by the convolutional layers. Those features are learned by the network and are inherently difficult to analyze. However, possible features could be the direction and position of strokes and horizontal projection. Direction and position of strokes are similar if characters share parts of their strokes. For example, B and 8 are very similar, since parts of the characters are the same. The reason why H is confused with M and U could be the similarity of their horizontal projections, which can be seen in Fig. 8. Another factor of influence on the prediction rates is the number of occurrences in the training data. The letters D, F, I, N, R ,V, X, and Y only occur at position three. Therefore, the network has extracted the features of those letters less often. It is possible that those features are less sophisticated and therefore lower the prediction rates.

We additionally investigate the influence of the compression on the similarity of characters. While for letters such as F, D, I, L, and M the order of the confusion does not change for the three most frequent predictions, it changes for other characters. This can be explained with varying compression artifacts which change for certain characters with the JPEG quality factor. Two examples are shown in Tab. 2. The letter C is confused with the letter E for all quality factors. Just the prediction three changes with higher quality factor. On the other hand, the letter P is mixed up with the letters E, M, and F for the quality factors 1, 15, and 95 respectively. This shows that a confusion table of network predictions to support police work has to be conditioned on the compression strength, which we will investigate in future work.

5 Conclusion

This paper investigates the recognition of license plates where the images have been subject to JPEG compression and other forms of strong degradation. To this end, a synthetic Czech dataset is created with low-resolution, noisy, and compressed images. We show that training a network with this dataset can act as a surrogate for a real-world dataset for analyzing image degradations.

Our analysis on the impact of image degradations on character recognition shows that for smaller images compression has a greater impact than for larger images. A quality factor of 1 leads to a drastic decrease in the recognition rate, almost at guessing rate for small images. Overall, synthetic Czech license plates can be reliably reconstructed if the width is above 30 pixels, the SNR is above -3 dB, and the JPEG quality factor is at least 15. We also observed that image resolution has a larger impact than compression when the character is highly variable, such as at the third position of Czech license plates. Furthermore, police investigations might benefit from character confusion tables. We show that such tables have to be created in dependency of the compression strength of the input image. We will consider this in our future work to further improve license plate recognition from images in the wild, i.e., with uncontrolled image degradations.

References

1. Abdullah, S., Mahedi Hasan, M., Islam, S.: YOLO-based three-stage network for Bangla license plate recognition in Dhaka metropolitan city. In: International Conference on Bangla Speech and Language Processing. pp. 1–6 (2018)
2. Agarwal, S., Tran, D., Torresani, L., Farid, H.: Deciphering severely degraded license plates. In: Media Watermarking, Security, and Forensics. pp. 138–143 (2017)
3. Arafat, M.Y., Khairuddin, A.S.M., Khairuddin, U., Paramesran, R.: Systematic review on vehicular licence plate recognition framework in intelligent transport systems. IET Intelligent Transport Systems **13**, 745–755 (2019)
4. Du, S., Ibrahim, M., Shehata, M., Badawy, W.: Automatic license plate recognition (ALPR): A state-of-the-art review. IEEE Trans. on Circuits and Systems for Video Technology **23**(2), 311–325 (Feb 2013)
5. Farid, H.: Photo Forensics. MIT Press (2016)

6. Glorot, X., Bengio, Y.: Understanding the Difficulty of Training Deep Feedforward Neural Networks. In: Int. Conf. on Artif. Intell. and Statist. pp. 249–256 (2010)
7. Gonçalves, G.R., Diniz, M.A., Laroca, R., Menotti, D., Schwartz, W.R.: Real-time automatic license plate recognition through deep multi-task networks. In: 31st Conference on Graphics, Patterns and Images. pp. 110–117 (2018)
8. Goodfellow, I., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. In: Int. Conf. on Learning Representations (2014)
9. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. Int. J. of Computer Vision **116**(1), 1–20 (2016)
10. Kilic, I., Aydin, G.: Turkish vehicle license plate recognition using deep learning. In: Int. Conf. on Artif. Intell. and Data Processing. pp. 1–5 (2018)
11. Li, H., Wang, P., You, M., Shen, C.: Reading car license plates using deep neural networks. Image and Vision Computing **72**, 14–23 (2018)
12. Liu, L., Zhang, H., Feng, A., Wan, X., Guo, J.: Simplified local binary pattern descriptor for character recognition of vehicle license plate. Int. Conf. on Comput. Graphics, Imaging and Visualization pp. 157–161 (2010)
13. Liu, P., Li, G., Tu, D.: Low-quality license plate character recognition based on CNN. In: Int. Symp. on Comput. Intell. and Design. pp. 53–58 (2015)
14. Llorens, D., Marzal, A., Palazón, V., Vilar, J.M.: Car license plates extraction and recognition based on connected components analysis and HMM decoding. In: Pattern Recognition and Image Analysis. pp. 571–578 (2005)
15. Lorch, B., Agarwal, S., Farid, H.: Forensic reconstruction of severely degraded license plates. In: Electronic Imaging (Jan 2019)
16. Ministerstvo vnitra České republiky: Collection of laws no. 343/2014 (2014), https://aplikace.mvcr.cz/sbirka-zakonu/ViewFile.aspx?type=z&id=27609
17. Pei-Lun, H., Liang, Y.M., Liao, H.Y.M.: Recognition of blurred license plate images. IEEE Int. Workshop on Information Forensics and Security (2010)
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. IEEE Conf. on Computer Vision and Pattern Recognition (2016)
19. Shivakumara, P., Tang, D., Asadzadeh, M., Lu, T., Pal, U., Anisi, H.: CNN-RNN based method for license plate recognition. CAAI Trans. on Intell. Technology **3**(3), 169–175 (2018)
20. Silva, S.M., Jung, C.R.: Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In: Conf. on Graphics, Patterns and Images. pp. 55–62 (2017)
21. Silva, S.M., Jung, C.R.: License plate detection and recognition in unconstrained scenarios. In: European Conference on Computer Vision. pp. 580–596 (2018)
22. Špaňhel, J., Sochor, J., Juránek, R., Herout, A., Maršík, L., Zemčík, P.: Holistic recognition of low quality license plates by CNN using track annotated data. In: IEEE Int. Conf. on Advanced Video and Signal Based Surveillance. pp. 1–6 (2017)
23. Suvarnam, B., Ch, V.S.: Combination of CNN-GRU model to recognize characters of a license plate number without segmentation. Int. Conf. on Advanced Computing & Communication Systems pp. 317–322 (2019)
24. Wen, Y., Lu, Y., Yan, J., Zhou, Z., Von Deneen, K., Shi, P.: An algorithm for license plate recognition applied to intelligent transportation system. IEEE Trans. on Intelligent Transportation Systems **12**, 830–845 (2011)