FORENSIC LICENSE PLATE RECOGNITION WITH COMPRESSION-INFORMED TRANSFORMERS

Denise Moussa[1,2], Anatol Maier[2], Andreas Spruck[3], Jürgen Seiler[3], Christian Riess[2,]*

[1]Federal Criminal Police Office (BKA), Germany
[2]IT Security Infrastructures Lab, Computer Science, Univ. of Erlangen-Nürnberg
[3]Multimedia Communications and Signal Processing, Electrical Engineering, Univ. of Erlangen-Nürnberg

ABSTRACT

Forensic license plate recognition (FLPR) remains an open challenge in legal contexts such as criminal investigations, where unreadable license plates (LPs) need to be deciphered from highly compressed and/or low resolution footage, e.g., from surveillance cameras. In this work, we propose a side-informed Transformer architecture that embeds knowledge on the input compression level to improve recognition under strong compression. We show the effectiveness of Transformers for license plate recognition (LPR) on a low-quality real-world dataset. We also provide a synthetic dataset that includes strongly degraded, illegible LP images and analyze the impact of knowledge embedding on it. The network outperforms existing FLPR methods and standard state-of-the art image recognition models while requiring less parameters. For the severest degraded images, we can improve recognition by up to 8.9 percent points.[1]

Index Terms— License Plate Recognition, Image Forensics

1. INTRODUCTION

LP detection and recognition is an active field of deep learning research. FLPR in particular aims at supporting criminal investigations by recognizing visually unreadable LP characters from severely degraded footage. Hereby, the data mostly stems from uncontrolled sources, often from surveillance cameras of third parties. Unfortunately, many commercial security cameras are budget-constrained systems using strong compression and low resolution to achieve high memory and cost efficiency [1]. Besides, these techniques amplify quality loss introduced by environmental factors like a vehicle's velocity and distance to the camera or weather and lighting conditions. Despite all these complicating factors for LPR, pioneering works show that neural networks (NNs) are still able to extract some LP information from synthetic very low-quality images where human analysts and traditional forensic image enhancement fail [1–5].

In this work, we aim at pushing the boundary for FLPR even further. Since to our knowledge there is no real world LP dataset featuring visually unreadable LP images, we generate SynthGLP, a synthetic dataset covering strong compression and low resolution, the arguably most challenging degradation factors for criminal investigations. We apply the JPEG compression [6] algorithm as it is the most widely spread lossy compression technique used by cameras and image editing tools and thus often appears in forensic image

analysis [7]. Regarding the architecture, we propose to use a Transformer network [8] which includes prior knowledge about the compression strength to guide character recognition. It requires considerably less network parameters than the state-of-the-art, but outperforms existing works. As a simple scalar measure for compression strength we use the JPEG quality factor (QF). We demonstrate that our method is applicable for libjpeg QFs (coarsely) estimated from image data compressed with arbitrary quantization matrices. To the best of our knowledge, this is the first work that uses a Transformer on the task of FLPR. Hence, we demonstrate its applicability on a low-quality real world automatic license plate recognition (ALPR) dataset. To summarize, the specific contributions of this work are:

- We propose a parameter-efficient Transformer [8] model for FLPR and evaluate it on real world data
- We show the added benefit of embedding compression knowledge at the example of JPEG compression
- We analyze performance on severely degraded, visually illegible synthetic LPs in detail

The remainder of this work is structured in four parts. Section 2 summarizes the related work, Sec. 3 describes our problem formulation, proposed network architecture and the used datasets. Section 4 presents our experiments and Sec. 5 concludes this work.

2. RELATED WORK

We distinguish ALPR and FLPR systems for LP identification. Many existing works address ALPR systems, which meet other requirements than FLPR. The main focus of ALPR systems is to automatically identify LP strings from (at least partially) controlled acquisition systems of reasonably high quality. Applications include traffic monitoring, automatic toll collection or access control [9]. The systems typically consist of an LP detection step and an LPR step. For LP detection, commonly YOLO-based networks are used [9–12]. LP recognition is generally implemented via convolutional neural networks (CNNs) [9–11] or convolutional recurrent neural networks (CRNNs) [12–16]. CNN methods classify each character separately. CRNNs consider the LP as a stream, which allows to flexibly process character sequences of varying lengths.

Several works recently paved the way for FLPR, which aims at deciphering unreadable LPs. Špaňhel *et al.* [17] presented a CNN recognition method together with a low quality real world LP dataset of Czech LPs called ReId. However, although ReId includes low-quality samples, they are not severely degraded, i.e., indecipherable. Agarwal *et al.* [2] are the first to show that CNNs can recover characters from synthetic LP data that are unreadable due to strong noise and low resolution. Their network recognizes two groups of three

[1]The source code and datasets are available at https://www.cs1.tf.fau.de/research/multimedia-security/
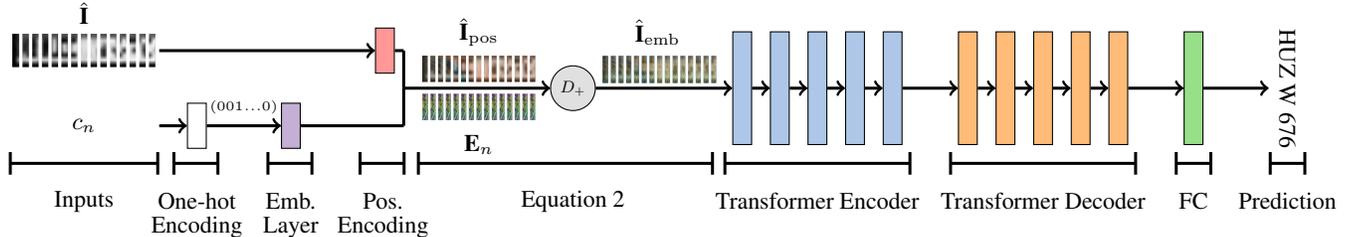
Fig. 1: The proposed Transformer architecture with knowledge embedding.

characters. Lorch *et al.* [3] generalize this approach to LPs with five to seven characters and remove constraints on the character format. Kaiser *et al.* [1] evaluate Lorch *et al.*'s [3] method for synthetic and real data and report lossy compression as one of the most challenging degradation factors. Rossi *et al.* [4] train a network that jointly consists of a U-NET denoiser and a CNN recognition module. The method outputs both a string of LP characters and a denoised version of the image. However, their image reconstruction requires images of slightly higher quality, and compression is not considered. Using a CRNN method was also proposed for FLPR [5]. An evaluation on synthetic data hereby showed increased recognition performance and higher robustness to out of distribution samples compared to CNNs.

Recently, Transformer networks [8] set the state of the art in natural language processing [18] and were subsequently adapted to image processing by Dosovitskiy *et al.* [19]. Transformers show very promising results on tasks like image classification, object detection or object segmentation [20].

In this work, we demonstrate the effectiveness of Transformer architectures for FLPR. Our proposed network uses available knowledge on the input compression strength. The integration of prior knowledge is similar to previous works on recommender systems [21, 22]. Here, additional information describing items is fed to the NN system to improve proposals for users. To the best of our knowledge, neither Transformers nor knowledge embedding have been investigated for FLPR.

3. METHODS

We model FLPR as a sequence-to-sequence (Seq2Seq) task that operates on an image $\mathbf{I} \in [0, 1]^{W \times H}$ of width W and height H. The input $\mathbf{I}$ is processed column-by-column as a series of image slices $\mathbf{i}_w \in [0, 1]^H$, yielding the input series $\hat{\mathbf{I}} = [\mathbf{i}_0, \mathbf{i}_1, ..., \mathbf{i}_W]$. An additional input to each $\hat{\mathbf{I}}$ is a scalar quantity $c_n \in [1, 100]$ that encodes the JPEG compression QF of $\hat{\mathbf{I}}$. Our specific task is, given c_n, to translate $\hat{\mathbf{I}}$ to a string output S of variable length with characters $a_m \in \mathcal{A}$ of an alphabet of all valid output tokens.

In this section, we briefly summarize the concept of Transformers, present our architecture and outline libjpeg [23] QF estimation.

3.1. Attention in Transformer Networks

Transformers [8] rely mainly on a scaled dot-product attention mechanism to grasp global dependencies between two sequences $\mathbf{S}_i$, $\mathbf{S}_j$. If self-attention is computed, $\mathbf{S}_i = \mathbf{S}_j$ are identical. The attention function maps a matrix triple consisting of a set of queries $\mathbf{Q}$, values $\mathbf{V}$ and keys $\mathbf{K}$ to an output. It is given by

$$\text{Att}(\mathbf{Q}, \mathbf{V}, \mathbf{K}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_K}}\mathbf{V}\right), \qquad (1)$$

where d_K is the dimension of the queries/keys and acts as normalizing factor. $\mathbf{Q}, \mathbf{K}$ and $\mathbf{V}$ are obtained by projecting input sequences with learnable weight matrices. Typically, Transformers learn multiple such projection matrices for inputs, yielding multiple $\mathbf{Q}, \mathbf{V}, \mathbf{K}$ representations. Attention is then computed in parallel over each triple $\mathbf{Q}, \mathbf{V}, \mathbf{K}$, which is called multi-head attention.

For solving Seq2Seq tasks, a transformer network consisting of an encoder and decoder is applied. The encoder processes the input sequence and outputs it to the decoder. The decoder further takes the output sequence elements from all previous time steps as input and yields the prediction. The encoder consists of several layers, each implementing self-attention and fully-connected (FC) sub-layers. Each sub-layer includes layer-normalization as well as a residual connection around itself. The decoder is constructed similarly, but additionally computes the encoder-decoder attention, where $\mathbf{K}$ and $\mathbf{V}$ stem from the encoder output, while $\mathbf{Q}$ stems from the decoder. For more details on Transformers, we refer to Vaswani *et al.* [8].

3.2. Network Architecture with Knowledge Embedding

Our Seq2Seq model is shown in Fig. 1. First, the input $\hat{\mathbf{I}}$ is combined with compression strength c_n as side information via an embedding layer. The result is fed to the network consisting of a Transformer encoder (blue) and decoder (orange), each of 5 layers with 8 attention heads. The dimension of each (FC) sub-layer is 2160. A final output FC layer (green) projects the decoder output in the vocabulary space of size $|\mathcal{A}|$, where $\mathcal{A}$ is the target vocabulary set. In our case it is dependent on the LP format and covers 36/41 tokens for Czech/German LPs plus three special Seq2Seq tokens.

The knowledge embedding with c_n is computed as follows. First, c_n is one-hot encoded and projected with a linear layer to the dimension of H. This vector is replicated W times to form $\mathbf{E}_n \in \mathbb{R}^{W \times H}$. The final embedded input sequence $\hat{\mathbf{I}}_{\text{emb}}$ is given by

$$\hat{\mathbf{I}}_{\text{emb}} = \text{Dropout}(\hat{\mathbf{I}}_{\text{pos}} + \mathbf{E}_n) \ , \qquad (2)$$

where $\hat{\mathbf{I}}_{\text{pos}}$ is the input $\hat{\mathbf{I}}$ after positional encoding [8]. Dropout regularization is applied with empirically chosen probability 0.5.

3.3. Estimating QFs

To acquire a measure for compression strength, we regress the scalar JPEG QF $\in [1, 100]$ of libjpeg's [23] quantization matrix ($\mathbf{M}_Q$). Embedding entire matrices leads to high combinatorial complexity when densely covering the whole $\mathbf{M}_Q$ space, which makes training infeasible. Following the approach of related work [24], we hence use the QF as surrogate value. In practice, the QF itself is generally not directly available from JPEG (meta)data and its value range depends on the specific $\mathbf{M}_Q$ used for compression. However, approaches for compression quality estimation exist. The $\mathbf{M}_Q$ may

(a) ReId set [17] example samples

r_w	JPEG quality factors		
	1	20	50
20			
30			
50			
70			
90			
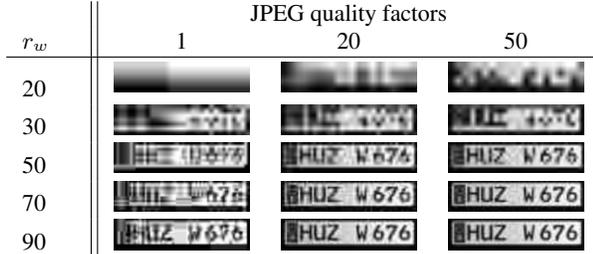
(b) Medium and low quality selection from our SynthGLP set

Fig. 2: Samples from featured real world and synthetic sets

be available from JPEG metadata or can be approximated [7]. The QF can then be estimated by orthogonal projection on the libjpeg standard $\mathbf{M}_Q$. To achieve this, we refer to the implementation of Cozzolino *et al.'s* [24] method.

4. EVALUATION

We train all networks for 100 epochs and take the best run. We use the Adam optimizer and progressively reduce the initial learning rate $\eta = 1e^{-4}$ by a multiplicative factor of 0.1 with a patience of 3 epochs. A train and validation batch size of 64 is employed. Seq2Seq models are trained with teacher forcing [25]. We evaluate our approach on ReId [17] and SynthGLP and report both the accuracy per LP (acc_{lp}) and character error rate (CER).

4.1. Datasets

We use ReId [17] to show the general applicability of our method to low-quality real images. To our knowledge, there exists no real dataset for severely degraded, unreadable LPs. Hence, we evaluate severe degradations in controlled experiments on synthetic data.

ReId consists of $105\,924$ train and $76\,412$ test samples of mostly Czech LPs from video cameras filming highways (Fig. 2a). The SynthGLP dataset is newly generated for this work. It consists of 900k/100k/1k images for the train/validation/test split. Each image shows a grayscale LP in German format [26], rendered at a resolution of 180×40 pixels by a framework similar to Spruck *et al.'s* [27]. LP characters are sampled from the valid alphabet per position but are otherwise random to prevent biases towards specific regions.

We compute degradations on the fly. The pipeline consists of normalization, bilinear down sampling to random pixel width $r_w \in [20, 180]$ with preserved ratio, JPEG compression with random QF $\in [1, 100]$ and bilinear upsampling to the original size (Fig. 2b).

4.2. Evaluation on Real World Data

We first show that Transformers are fit for the task of real world LPR by training our model without knowledge embedding (LP-Transf.) on the ReId [17] training split and evaluating on the test split. Table 1 shows the results of running our model and related work [1, 5], as well as Špaňhel *et al.'s* [17] reported values. We slightly outperform the related methods for FLPR [1,5] and share the same CER = 0.004

Type	Method	acc_{lp}	CER	Params
CNN	Špaňhel-S [17]	*98.3%*	**0.004**	∼ 8M
	Špaňhel-L [17]	**98.6%**	**0.004**	∼ 17M
	Kaiser [1]	97.3%	0.029	∼ 45M
CRNN	Moussa [5]	98.1%	**0.004**	∼ 4.5M
Transf.	LP-Transf.	*98.3%*	**0.004**	**∼ 1.9M**

Table 1: Performance on the ReId test split [17].

Type	Method	acc_{lp}	CER	Params
CNN	Špaňhel-S [17]	75.10%	0.0597	∼ 8M
	Špaňhel-L [17]	79.90%	0.0479	∼ 17M
	Kaiser [1]	80.15%	0.0449	∼ 45M
	EffNet-B0 [28]	83.89%	0.0378	∼ 40M
	EffNet-B7 [28]	87.25%	0.0304	∼ 132M
CRNN	Moussa [5]	92.48%	0.0221	∼ 4.5M
Transf.	LP-Transf.	92.41%	0.0215	
	LP-Transf.-5	92.34%	0.0211	
	LP-Transf.-10	92.54%	0.0206	∼ 1.9M
	LP-Transf.-25	92.51%	0.0206	
	LP-Transf.-50	**92.83%**	**0.0195**	
	LP-Transf-100	*92.66%*	*0.0202*	

Table 2: Results on SynthGLP$_{TFull}$. LP-Transf.–$\{10, 25, 50, 100\}$ surpass all other approaches, where $K = 50$ is best. LP-Transf.(–5) achieve higher CER but lower acc_{lp} than the best baseline CRNN.

as Špaňhel–$\{S,L\}$. Špaňhel–L achieves a slightly higher acc_{lp} than the other two models with $93.6\% > 93.3\%$. All in all, we report LP-Transf. to perform comparably, while being far more parameter-efficient with 1.9M trainable weights as opposed to 8M and 17M weights (Špaňhel–$\{S,L\}$).

4.3. Evaluation on Synthetic Data

We analyze the advantage of incorporating prior knowledge about compression levels on SynthGLP. For the train and validation split we use two fixed random seeds for degradation calculations to ensure identical data for all models. For testing, we generate SynthGLP$_{TFull}$. It contains 858 degraded variants of SynthGLP's test split, each featuring one of all possible (c_n, r_w) combinations, with $c_n \in \mathcal{C}_1 = \{i \cdot 4 + 1 | i \in [0, 24]\} \cup \{100\}$ and $r_w \in \mathcal{R}_1 = \{i \cdot 5 | i \in [4, 36]\}$. SynthGLP$_{TFull}$ is designed to evenly sample all degradation levels covered in training. To analyze performance on very low resolution images, we also generate SynthGLP$_{TLow}$ covering 156 test sets with $(c_n, r_w) \in \mathcal{C}_1 \times \mathcal{R}_2$, $\mathcal{R}_2 = \{i | i \in [20, 25]\}$.

4.3.1. Models

The evaluation covers various baseline approaches, and our Transformer model with and without knowledge embedding.

Baseline models include the same methods as for the experiments on ReId [1, 5, 17]. We also feature state-of-the-art Efficient-Net (EffNet) models B0 and B7 [28]. All models' input layers are adapted to input size 180×40. We extend Kaiser *et al.*'s model [1] to 9 output layers to account for longer possible lengths of German LPs and pre-initialize the EffNet models with ImageNet weights [29] prior to adding a classification head of 9 FC layers with size 2048.

LP-Transf.-K is the extension of LP-Transf. by a knowledge embedding layer with $K \in \mathbb{N}$ knowledge classes (Sec. 3.2, Fig. 1). $K = 100$ accounts for the full value range of the QF from libjpeg.
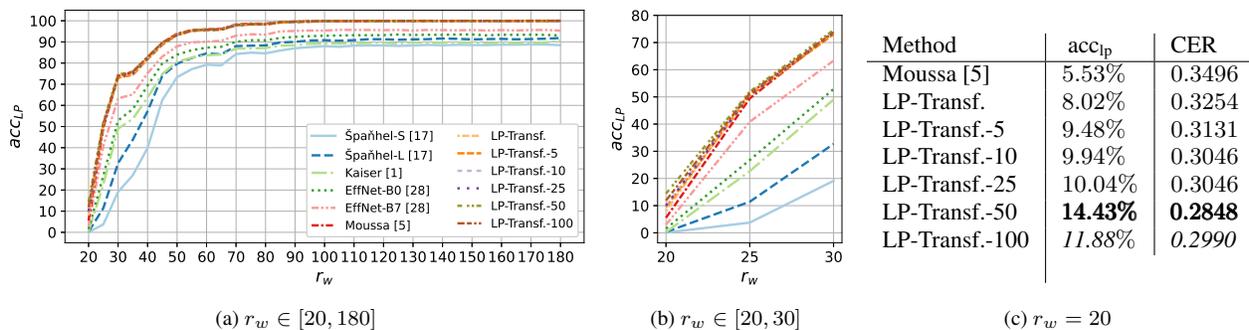
(a) $r_w \in [20, 180]$	(b) $r_w \in [20, 30]$	(c) $r_w = 20$

Fig. 3: acc_{lp} results averaged over $c_n \in \mathcal{C}_1$ on SynthGLP$_{TFull}$. Seq2Seq models perform comparably well for $r_w \geq 30$ and surpass all CNN methods. The advantage of knowledge embedding steadily enlarges with decreasing r_w and is highest for $r_w = 20$ as depicted in Tab. 3c.
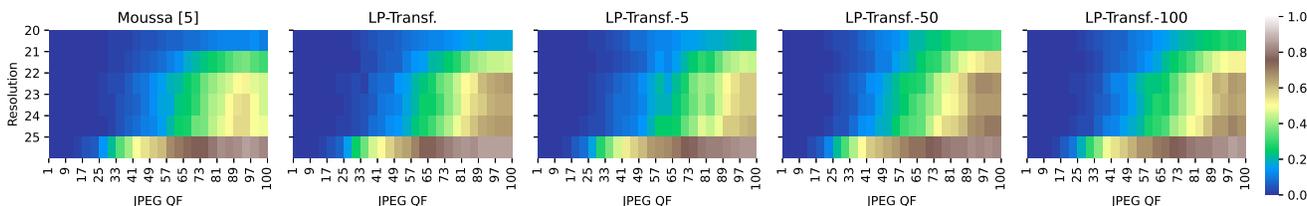
The table in Fig. 3c:

Method	acc_{lp}	CER
Moussa [5]	5.53%	0.3496
LP-Transf.	8.02%	0.3254
LP-Transf.-5	9.48%	0.3131
LP-Transf.-10	9.94%	0.3046
LP-Transf.-25	10.04%	0.3046
LP-Transf.-50	**14.43%**	**0.2848**
LP-Transf.-100	*11.88%*	*0.2990*

Fig. 4: acc_{lp} for the best performing baseline CRNN [5] and our method.

To simulate estimated QFs, we linearly map the values to a smaller range and set $c_n = \left\lfloor \frac{K \cdot QF}{100} \right\rfloor$ for $K \in \{50, 25, 10, 5\}$ classes. This simulates QFs that are estimated correctly within $l \in \{2, 4, 10, 20\}$ levels, accordingly. Hence, our hardest included scenario is covered by LP-Transf.-5, which deals with only 5 distinguishable classes, where one class includes $l = 20$ QF levels.

4.3.2. Results

The averaged performance over SynthGLP$_{TFull}$ is given by Tab. 2. Seq2Seq methods show to solve the task better than CNNs. The method of Špaňhel *et al.* did achieve less good results than on ReId (Tab. 1), which we attribute to SynthGLP$_{TFull}$ containing distinctly stronger degraded data. This is also consistent with the other CNN methods' results, where performance rises with parameter size. LP-Transf.–$\{10, 25, 50, 100\}$ outperform all other models, where $K = 50$ performs best with $acc_{lp} = 92.83\%$ and CER = 0.0195. LP-Transf.–100 performs second best. We hypothesize that the better performance for $K = 50 < 100$ is attributed to the nature of JPEG compression. Neighboring QFs yield similar compression quality, therefore halving the QF dimension does not lead to a significant information loss but simplifies the embedding layer's optimization. The performance of the standard LP-Transf. model and LP-Transf.–5 is comparable to the best baseline CRNN [5] model.

A further analysis showed that the advantage of incorporating estimated QF knowledge is particularly large for severely degraded samples. Fig. 3a shows the acc_{lp} for SynthGLP$_{TFull}$ averaged over $c_n \in \mathcal{C}_1$ per $r_w \in \mathcal{R}_1$, Fig. 3b shows the same for $r_w \in [20, 30]$. Seq2Seq approaches perform comparably down to $r_w = 30$ and surpass CNN approaches for all r_w by a great margin. Fig. 3b shows a considerable benefit of knowledge embedding for $r_w \leq 30$. Especially the performance boost for the lowest $r_w = 20$ is apparent as also indicated by the table in Fig. 3c. While all our models out-

perform the CRNN [5], LP-Transf.–50 gains the greatest advantage with $acc_{lp} = 14.43\% > 5.53\%$ and CER = 0.2848 < 0.3496. This shows that incorporated QF knowledge is of increasing importance for increasing compression strength. Especially for low r_w, where compression removes a large amount of information from the image content, the LP-Transf.-K models offer the biggest advantage.

To explore this phenomenon further, we analyze the performance on very low resolution samples via SynthGLP$_{TLow}$. Fig. 4 shows the acc_{lp} results for the best performing baseline CRNN model, LP-Transf. and for the weakest $K = 5$, best $K = 50$ and standard $K = 100$ LP-Transf.-K model. Obviously, LP-Transf. already handles the very low quality input better than the CRNN. The LP-Transf.-K models further increase the advantage and recognize very low quality images more robustly. For $K = 50$ the performance is best, where the acc_{lp} for the lowest $r_w = 20$ first surpasses 20% for QF = 61 and reaches 31.3% for QF = 100 while the CRNN remains constantly below 12%.

5. CONCLUSION

In this work, we showed the effectiveness of using compression levels of images as prior knowledge to a parameter efficient Transformer model for FLPR. We showed the applicability of Transformers for LPR on real world data and evaluated our knowledge embedding method on SynthGLP, our dataset specifically generated to fit common image degradation challenges in forensic investigations. We match the performance of the best existing FLPR approach on high and medium quality data while needing less parameters. In addition, we strongly outperform all existing methods for low quality data with QFs only estimated correctly down to 20 levels ($K = 5$).

For future work, we plan on incorporating more quality parameters from image (meta)data in our model to drive forward the research on FLPR.

6. REFERENCES

[1] P. Kaiser, F. Schirrmacher, B. Lorch, and C. Riess, "Learning to Decipher License Plates in Severely Degraded Images," in *Pattern Recognition. ICPR International Workshops and Challenges*, 2021, pp. 544–559.

[2] S. Agarwal, D. Tran, L. Torresani, and H. Farid, "Deciphering Severely Degraded License Plates," *Electronic Imaging*, vol. 2017, no. 7, pp. 138–143, 2017.

[3] B. Lorch, S. Agarwal, and H. Farid, "Forensic Reconstruction of Severely Degraded License Plates," *Electronic Imaging*, vol. 2019, no. 5, pp. 529–1–529–7, 2019.

[4] G. Rossi, M. Fontani, and S. Milani, "Neural Network for Denoising and Reading Degraded License Plates," in *International Conference on Pattern Recognition*, 2021, pp. 484–499.

[5] D. Moussa, A. Maier, F. Schirrmacher, and C. Riess, "Sequence-Based Recognition of License Plates with Severe Out-of-Distribution Degradations," in *International Conference on Computer Analysis of Images and Patterns*, 2021, pp. 175–185.

[6] G. K. Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[7] T. H. Thai, R. Cogranne, F. Retraint, et al., "JPEG Quantization Step Estimation and its Applications to Digital Image Forensics," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 123–133, 2016.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[9] R. Laroca and D. Menotti, "Automatic License Plate Recognition: an Efficient and Layout-Independent System Based on the YOLO Detector," in *Anais Estendidos do XXXIII Conference on Graphics, Patterns and Images*, 2020, pp. 15–21.

[10] 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," in *2018 IEEE International Joint Conference on Neural Networks*, 2018, pp. 1–10.

[11] S. Montazzolli Silva and C. R. Jung, "License Plate Detection and Recognition in Unconstrained Scenarios," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 580–596.

[12] L. Zhang, P. Wang, H. Li, Z. Li, C. Shen, and Y. Zhang, "A Robust Attentional Framework for License Plate Recognition in the Wild," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[13] B. Shi, X. Bai, and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and its Application to Scene Text Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[14] P. Shivakumara, D. Tang, M. Asadzadehkaljahi, T. Lu, U. Pal, and M. H. Anisi, "CNN-RNN Based Method for License Plate Recognition," *CaaI Transactions on Intelligence Technology*, vol. 3, no. 3, pp. 169–175, 2018.

[15] B. Suvarnam and V. Sarma Ch, "Combination of CNN-GRU Model to Recognize Characters of a License Plate Number Without Segmentation," in *5th International Conference on Advanced Computing & Communication Systems*, 2019, pp. 317–322.

[16] H. Zhang, F. Sun, X. Zhang, and L. Zheng, "License Plate Recognition Model Based on CNN + LSTM + CTC," in *International Conference of Pioneering Computer Scientists, Engineers and Educators*, 2019, pp. 657–678.

[17] 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 *14th IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2017, pp. 1–6.

[18] D. W. Otter, J. R Medina, and J. K Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2020.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2020.

[20] S. Khan, Ms Naseer, Ms Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *ACM Computing Surveys*, 2021.

[21] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel Recurrent Neural Network Architectures for Feature-Rich Session-Based Recommendations," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 241–248.

[22] E. Fischer, D. Zoller, A. Dallmann, and A. Hotho, "Integrating Keywords into BERT4Rec for Sequential Recommendation," in *German Conference on Artificial Intelligence*, 2020, pp. 275–282.

[23] Independent JPEG Group, "Libjpeg," accessed: 2022-01-15, `http://libjpeg.sourceforge.net/`.

[24] D. Cozzolino and L. Verdoliva, "Noiseprint: a CNN-Based Camera Model Fingerprint," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 144–159, 2019, Implementation for JPEG QF Estimation, accessed 2021-11-01: `https://github.com/grip-unina/noiseprint/blob/afd0b8e7dd6c33a345ef9f09eb0b456cf0a28ee6/noiseprint/utility/utilityRead.py`.

[25] R. J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.

[26] BMVBS, "Verordnung über die Zulassung von Fahrzeugen zum Straßenverkehr," 2011.

[27] A. Spruck, M. Hawesch, A. Maier, C. Riess, J. Seiler, and A. Kaup, "3D Rendering Framework for Data Augmentation in Optical Character Recognition," in *2021 International Symposium on Signals, Circuits and Systems*, 2021, pp. 1–4.

[28] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[29] Pytorch Vision Models, "torchvision.models — Torchvision 0.11.0 documentation," accessed: 2021-12-02, `https://pytorch.org/vision/stable/models.html`.