Incremental Whole Plate ALPR Under Data Availability Constraints

Markus Russold[a], Martin Nocker[b] and Pascal Schöttle[c]
MCI The Entrepreneurial School, Innsbruck, Austria

Keywords: Automatic License Plate Recognition, Continual Learning, Synthetic Data Generation, Computer Vision.

Abstract: In the realm of image processing, deep neural networks (DNNs) have proven highly effective, particularly in tasks such as license plate recognition. However, a notable limitation in their application is the dependency on the quality and availability of training data, a frequent challenge in practical settings. Addressing this, our research involves the creation of a comprehensive database comprising over 45,000 license plate images, meticulously designed to reflect real-world conditions. Diverging from conventional character-based approaches, our study centers on the analysis of entire license plates using machine learning algorithms. This novel approach incorporates continual learning and dynamic network adaptation techniques, enhancing existing automatic license plate recognition (ALPR) systems by boosting their overall confidence levels. Our findings validate the utility of machine learning in ALPR, even under stringent constraints, and demonstrate the feasibility and efficiency of recognizing license plates as complete units.

1 INTRODUCTION

Automatic License Plate Recognition (ALPR) is a cornerstone in applications like toll collection and traffic management (Goncalves et al., 2018; Jiang et al., 2023). System operators are increasingly relying on ALPR for automation, hence, systems' ability to perform consistently and accurately under diverse conditions like moving vehicles and weather becomes increasingly important (Gao and Zhang, 2021; Schirrmacher et al., 2023).

ALPR algorithms provide both the license plate number and a confidence level associated with each recognition. This confidence level serves as a critical determinant in deciding whether the data requires manual post-processing, shaping the overall efficiency and reliability of ALPR systems. Variability in conditions can challenge the confidence level and may require human verification (Ahmad et al., 2015; Shashirangana et al., 2021; Bulan et al., 2015).

In this paper, we first create a database of over 45,000 license plates and subsequently evaluate a deep learning-based method to improve the ALPR pipeline. Our approach aims to reduce manual checks by adding an advanced post-processing step. This involves using deep neural networks (DNNs) to adapt to new data and requires preserving previous knowledge. Because of data privacy regulations like the European General Data Protection Regulation (GDPR)[1] or other availability constraints such as limitations in storage capacity, the algorithm must be efficient as retraining with old data is not possible (Seunghui, 2021; Van de Ven et al., 2020; Ke et al., 2023).

Figure 1: Example of Six Variants of the Same License Plate Number.

To summarize, we make the following contributions:

- We provide a synthetic data generation method for license plate images in ALPR applications and create a comprehensive and diverse database of over 45,000 license plates, which is a valuable resource for training and testing ALPR algorithms under various real-world conditions.
- Our research introduces an advanced DNN-based approach, designed to extend existing ALPR

[a] https://orcid.org/0009-0009-1650-4733
[b] https://orcid.org/0000-0002-6967-8800
[c] https://orcid.org/0000-0001-8710-9188

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

Russold, M., Nocker, M. and Schöttle, P.
Incremental Whole Plate ALPR Under Data Availability Constraints.
DOI: 10.5220/0012566400003654
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2024)*, pages 131-140
ISBN: 978-989-758-684-2; ISSN: 2184-4313
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

pipelines. This approach uniquely operates on whole license plate images rather than focusing on single characters, thereby augmenting, rather than replacing, the current methodologies in the field.

- We propose a continual learning methodology specifically designed for ALPR systems, allowing them to adapt and evolve with new data while efficiently preserving previous knowledge.

The rest of this paper is structured as follows. Section 2 briefly introduces ALPR and continual learning of DNNs and discusses related work. Section 3 defines the problem we tackle and our specific constraints. Our method to generate synthetic license plate images is described in Section 4. Next, Section 5 covers our experimental setup and methods before we present our results in Section 6. We conclude with a summary and discussion in Section 7.

2 BACKGROUND & RELATED WORK

This section provides a very short overview of relevant related work, we provide a more extensive list in the supplementary material.

2.1 The ALPR Pipeline

Despite its long history, dating back to the 1970s, ALPR remains a difficult and complex problem, partly because of changing color and lighting conditions influenced by dynamic weather variations. ALPR algorithms continue to move away from traditional techniques based on computer vision and optical character recognition (OCR) towards the use of deep neural networks, e.g., convolutional neural networks (CNNs) (Li and Shen, 2016) and You Only Look Once (YOLO) models (Al-batat et al., 2022).

Traditional ALPR processing is typically broken down into four sub-tasks following a so-called "multi-stage" approach: image pre-processing, license plate detection (localization), character segmentation and character recognition (Pustokhina et al., 2020; Shashirangana et al., 2021; Izidio et al., 2018; Jiang et al., 2023).

The basic output of the recognition process is the combination of recognized characters and the computed confidence level per character. Therefore, an overall level of confidence is to be calculated. The literature discusses several ways to accomplish this, whereas it recognizes the importance of the level of uncertainty for further processing (Schirrmacher et al., 2023). There are several approaches to calculate the overall level of confidence and ultimately, the choice of which method to use will depend on the specific application and the desired level of accuracy (Goncalves et al., 2016; Hendry and Rung-Ching, 2019; Schirrmacher et al., 2023).

2.2 Continual Learning of DNNs

Continual learning is the ability of a model to learn new data without forgetting the previously learned data (De Lange et al., 2022). In the context of this work, this means that the model can learn to recognize new license plates without losing the ability to recognize previously seen license plates. The emergence of new data may necessitate the adaptation of the model. Adaptation is the ability of a model to adjust to changes in the data distribution, for example, when new output layers need to be added. Fixing or adjusting deep neural networks is a complex challenge (Jinwook et al., 2022).

The effect of a model losing its ability to accurately remember and perform well on previously learned data when trained on new data is referred to as catastrophic forgetting. This issue is particularly prominent in systems that learn sequentially or incrementally, as they often need to adapt and improve their performance over time (Kirkpatrick et al., 2017). A variety of strategies have been proposed to counter catastrophic forgetting, from flexible network structures to memory models and gradient-based optimizations. The main aim is to balance knowledge transfer and interference. The lack of a clear trend suggests that further research is needed to solidify effective methods (Kirkpatrick et al., 2017; French, 1999; Van de Ven et al., 2020).

2.3 Open Problems

One of the main challenges in conducting ALPR research for this particular study is the lack of a suitable dataset. Specifically, we require a large number of images featuring the same license plates under varying conditions. To the best of the authors' knowledge, no available dataset meets this requirement. For example, Goncalves et al. (2018) propose a new dataset with 6,775 frames with 8,683 different license plates thus indicating, that license plates do not reappear often. Also Špaňhel et al. (2018) proposes a new dataset called CamCar6k, but this one also lacks the reappearance of license plates as they have recorded 7.5 hours of license plates on a vehicle passing, resulting in a total of 6,064 images of license plates. Furthermore, most popular approaches to ALPR rely on OCR and

classify based on individual characters (Shashirangana et al., 2021; Du et al., 2013). In contrast, our approach of considering entire license plates as individual classes has largely been overlooked by the research community. This approach also necessitates adapting the network to accommodate to new license plates. Moreover, the assumption that past data is unavailable for training creates the need for a continual learning approach. Collectively, the sum of these challenges has not been comprehensively addressed by previous works.

3 PROBLEM DEFINITION

Based on the gaps in current research stated above, we generate a dataset of labeled license plates to answer the following research question:

Can deep learning algorithms be used to automatically recognize license plates given that,

1. the algorithm is performed on the whole license plate, rather than on single characters, and
2. the imagery data becomes available incrementally, whilst old data shall be deleted frequently due to data privacy (or other) reasons?

This question addresses the enhancement of the performance of ALPR pipelines with the usage of machine learning algorithms in a post-processing kind of operation, given the following specific constraints:

1. Image data shall be assumed only available for a short time period, as it shall be deleted as soon as possible in order to respond to privacy regulations (such as the GDPR). As a consequence, the neural network can not be retrained using the original set of data, but must get enhanced instead without losing previously learned data.
2. License plate recognition shall be based on the whole image of the license plate. Most ALPR algorithms recognize and process the individual characters of the license plate. This research explores an alternative method for achieving the same outcome. In this approach, we eliminate the need for character segmentation but require the neural network to have one output node per license plate. As a consequence, the artificial neural network must be able to adapt over time as it needs to learn and process new license plates.

4 GENERATION OF SYNTHETIC DATA

One of the main characteristics of the analyzed use case is the recurring appearance of license plates and the subsequent continual training of the model. Thus, the used image set requires the very same license plates to appear multiple times in order to measure the recognition accuracy as the number of collected images increases.

To the best of the authors' knowledge, an image set matching the criteria of a set of identical license plate numbers appearing several times in the data was not available. Hence, synthetic data was used for the experimental implementation. Previous research also faced problems of not having suitable data available and used synthetic license plate images or captured own data in order to train and test their proposed algorithms. For example, Izidio et al. (2018), Andersson (2022), and Bulan et al. (2015) discussed the usage of synthetically generated data. Ke et al. (2023) though, captured their own data as described. Also, Schirrmacher et al. (2023) discussed the use of synthetic data for their work, as they mimicked the way Kaiser et al. (2021) proposed the generation of such data.

In this study, synthetic data was generated using our Python implementation, that uses the Pillow, pandas, numpy, openpyxl and matplotlib libraries. As a welcome side effect, the use of generated data eliminates any ethical concerns and any data privacy considerations (such as GDPR), since this study does not use any real license plate images.

4.1 Format and Syntax of the License Plates

The format of the license plate is similar to, but for simplicity reasons not identical to, what modern license plates in the Federal Republic of Germany look like.[2] There is no specific reason for choosing Germany license plates other than the need to follow some guidance in order to be as close to reality as possible.

The following main principles are obeyed: (1) The syntax of the license plate is matching the following pattern: Two uppercase letters (including umlauts Ä, Ö, and Ü), followed by a space, followed by two more uppercase letters, then another space and four digits. The letters are randomly selected from the complete list, resulting in a uniform distribution

[2]Federal Republic of Germany, Federal Ministry of Justice, https://www.gesetze-im-internet.de/fzv_2023/anlage_4.html

across the generated license plate. As a consequence, some letters (e.g., Ä, Ö, Ü) might appear more frequently than they typically would in reality. (2) Furthermore, black characters on white background, and (3) the used typeface is "FE-Schrift" (German: "Fälschungserschwerende Schrift"), a special typeset developed to make tampering difficult.

4.2 Implementation of the Data Generator

The algorithm generates a list of unique license plate numbers and iteratively produces JPG images with dimensions of 340×1130×3 from a random subset of this list. Sampled license plates occurring across different cycles simulate their occurrence at different points in time. Later, the model is trained using images from each new cycle and is extended as new license plates occur with new cycles. Additionally, a table with a row for each license plate and a column for each cycle is generated and saved to an Excel sheet, aiding in subsequent experiment simulations.

To mimic real-world conditions, the algorithm applies disturbances such as: (1) Random rotation between $-\varphi$ and $+\varphi$ degrees, (2) Gaussian blur with a radius between 0 and β, (3) random noise with a normal distribution of mean 0 and standard deviation 2, scaled by 30, to every pixel of the image and (4) Grainy noise as done by Schirrmacher et al. (2023) and Izidio et al. (2018).

Figure 1 shows an example set of six license plate images which have been synthetically generated using the described algorithm. Even though the license plate number is the same for all variants, each image is unique and shows specific characteristics such as angles, illumination, or level of noise.

4.3 The Dataset

The parameters for the data generation were configured as follows: The maximum rotation angle was set to $[-2,2]$ degrees, the Gaussian blur radius was set to $[0,2]$, the grains to add indicator was set to "Yes", the different illumination conditions value was also set to "Yes", the number of license plates to generate was set to 1,000, the number of cycles was set to 183, and the probability for a license plate to appear in a cycle was set to 0.25.

This configuration led to the creation of 45,751 different images for a total of 1,000 license plate numbers, randomly spread over 183 cycles given a probability of 0.25. These images consumed approximately 6,250 megabytes of disk space.

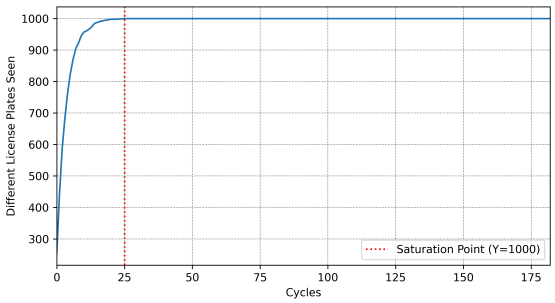

Figure 2: Saturation of License Plates Seen in the Data.

Due to the probability of 0.25 to appear in a certain cycle, the number of license plates per cycle for which an image was created is limited. This was done in order to delay the saturation threshold where all license plate numbers have at least been seen once by the model. The mean value of created images per cycle was 250.01 with a standard deviation of 12.37. The minimum and maximum values observed were 215 and 286, respectively. Again, those values were expected given the configured probability factor.

The saturation point, i.e., the cycle where all license plates occurred at least once, was observed at index 25. The evolution of the saturation can be seen in Figure 2.

When looking at the horizontal data distribution (number of images per license plate) the mean value can be found at 45.75, with a standard deviation of 5.75. The values range from a minimum of 28 to a maximum of 64. Also, these figures were to be expected given the distribution probability of 0.25.

The authors made the source code of the license plate generator available on GitHub under the Creative Commons license.[3]

5 WHOLE PLATE ALPR EXPERIMENT

The experiment was implemented using Python version 3.9.12 and was carried out on a Linux Ubuntu machine with version 5.4.0-77-generic on an Intel Xeon(R) CPU E5-2630 v4 operating at a frequency of 2.2 GHz. The CPU had 6 cores and the computer was equipped with 16.8 GB of RAM. The target computer was not equipped with a GPU, which would certainly have accelerated the execution.

The script required a total execution time of 17,145.52 seconds, equivalent to four hours and 46 minutes. During this process, 84% of the total duration accounted for model training, including both

[3]https://github.com/markusrussold/ALPRDataset

Table 1: Model Architecture.

Layer	Kernel Size	Activation Function	Output Shape
Input Layer	-	-	(64, 300)
Conv2D	(3, 3)	relu	(62, 298)
MaxPooling2D	(2, 2)	-	(31, 149)
Conv2D	(3, 3)	relu	(29, 147)
MaxPooling2D	(2, 2)	-	(14, 73)
Conv2D	(3, 3)	relu	(12, 71)
MaxPooling2D	(2, 2)	-	(6, 35)
Conv2D	(3, 3)	relu	(4, 33)
MaxPooling2D	(2, 2)	-	(2, 16)
Flatten	-	-	512
Dense	-	relu	512
Dense	-	softmax	[num_output_nodes]

the initial training phase and subsequent re-training iterations. The remaining portion of time was allocated to image pre-processing, predicting, and documenting the results. It should be mentioned that the script's feasibility was its main priority rather than performance optimization.

5.1 AI Network Architecture

The study uses a CNN, a method known for its effectiveness in image tasks. The model reshapes input images to 64×300 pixels and processes them through convolutional layers with rectified linear unit (ReLU) activation functions. Moreover, the model uses max pooling and fully connected layers, including a dense layer with 512 nodes. The architecture is dynamic, adapting the number of output nodes based on the license plates presented to the model during training. This versatile model, resulting from extensive testing, aims for balanced performance rather than optimizing for just prediction or training. Table 1 lists the detailed model architecture.

5.2 Approach to Continual Learning

The study utilized the principles of continual learning by implementing an iterative training process using the TensorFlow framework. In each cycle, the deep learning model was further trained on a new set of data without resetting the previously learned weights. This was made possible by TensorFlow's `tf.keras.Model` functionalities.

After the initial training phase, where the `compile()` method was used to configure the model's learning process, we employed the `fit()` method to initiate training.[4] Critically, in subsequent cycles, the `fit()` method was invoked again on the same model instance, thereby updating the model with new data while retaining and refining the knowledge it had already acquired. This approach effectively embodies the essence of continual learning, where the model dynamically adapts and evolves, improving its performance and generalization capabilities over time.

By continuously training the very same model after each cycle, we ensured that our model did not suffer from catastrophic forgetting, a common challenge in continual learning scenarios. Instead, it was able to incrementally build upon its previous knowledge, demonstrating a significant improvement in license plate recognition accuracy with each additional training cycle. This continual learning process was pivotal in enabling the model to adapt to new, previously unseen license plates, thereby enhancing the robustness and reliability of the ALPR system.

5.3 Approach to Dynamic Adaptation of Number of Output Nodes

In the study, the dynamic nature of new license plate numbers required a flexible model. A custom label encoder class maintained fixed mappings between license plates and output nodes to handle new labels. When new labels appeared, a new model was created with more output nodes. The weights and biases from the previous model were copied over, adjusting only the new output nodes with zero values. In the experiment, the model expanded 22 times, stabilizing after 26 cycles.

5.4 Image Pre-Processing

Image pre-processing is crucial for model performance, aligning with existing literature (Tarigana et al., 2017; Selmi et al., 2017; Špaňhel et al., 2018). We apply techniques to mitigate disturbances from the data generator and standardize images related to the same license plate. The pre-processing steps include resizing, grayscale conversion, Gaussian blur, angle correction, cropping, sharpening, and final resizing to 64×300×1 to fit the neural network input. Figure 3 shows a comparison between the original (left) images as they were created by the data generator and their equivalents after the pre-processing operation (right) which were used for training and prediction input.

5.5 Model Training and Adaptation Process

Initially, the model is trained utilizing pre-processed images acquired from cycle zero. Each image is associated with its corresponding label using the custom-built label encoder. For training, the Adam optimizer

[4] https://www.tensorflow.org/api_docs/python/tf/keras/Model\#fit

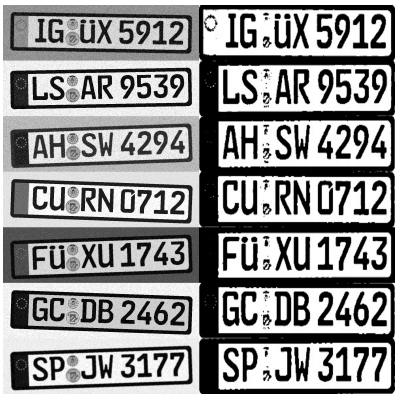

Figure 3: Examples of License Plate Images and Their Representation After Pre-Processing.

is used, and the loss is set to the sparse categorical crossentropy loss. Because of the necessary network adaptations when encountering new license plates, we use integer labels rather than one-hot encoded vectors. Therefore, the sparse categorical crossentropy loss is favorable.

For cycles one to 182 the following steps are repeated: (1) Loading license plate images of the current cycle, (2) pre-processing the images, (3) predicting the license plate for images of the current cycle using the model which is trained on data from the previous cycles, (4) analyzing the labels from the current cycle and adding the new labels to the list of labels without affecting the order of the previously seen labels, (5) adaptation of the model to the new number of output nodes corresponding to the total number of known labels, and (6) re-training of the model to enhance previously seen classes and to train new output nodes for the first time. Ten epochs of training are executed for re-training. This process yielded 182 execution time measurements and accuracy results as discussed in the following section.

6 RESULTS

A comprehensive analysis was conducted on the collected data, and various statistical inspections were performed to explore the relationships and patterns within the dataset.

First, a discussion on the evolution of the prediction accuracy is presented. Subsequently, the computation times for both, prediction and training operations are discussed including their implications and what can be learned from them.

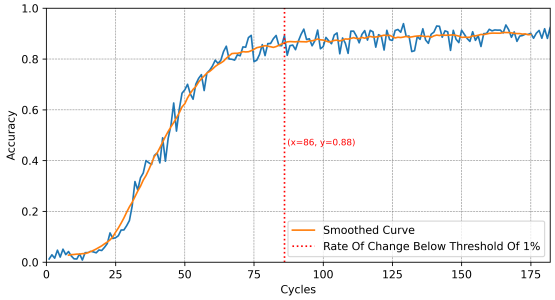

Figure 4: Evolution of the Prediction Accuracy.

6.1 Evolution of the Prediction Accuracy

A graphical illustration of the evolution of the prediction accuracy per cycle is shown in Figure 4.

The mean value across of the prediction accuracy over all 182 cycles is 0.6849 with a standard deviation of 0.3150. The minimum value is 0.0083, while the maximum value is 0.9393. The values for the n^{th} quartile $q(n)$ are the following: $q(25) = 0.5542$, $q(50) = 0.8528$, and $q(75) = 0.8943$, where $q(50)$ is also known as the median.

It is clearly visible in Figure 4 that the prediction accuracy increases over time, as one would expect from a gradually updated deep learning model. At some point, the curve appears to flatten out, however, individual results are strongly fluctuating after that point.

A smoothed curve was calculated in order to measure the rate of change and to determine the position where the curve can be assumed to have flattened out (where the model apparently converged or was about to converge). This was accomplished by calculating the moving mean of the accuracy values by applying a rolling window with the size of 15 elements. Subsequently, the rate of change was calculated and the first element where the rate is smaller than 0.01 was determined at cycle 86. Hence, this is the point where the model reached a more or less stable state and further training or iterations did not significantly improve its performance or reduce its loss.

Only looking at the data from index 86 onwards gives the following insights: The remaining 97 accuracy values exhibit a mean value of 0.8873. The dispersion of the data, represented by the standard deviation of around 0.0277, is relatively low. The dataset is spanning from a minimum value of 0.8147 to a maximum value of 0.9393 with $q(25) = 0.8719$, $q(50) = 0.8915$, and $q(75) = 0.9098$. When fitting a line into the accuracy values between cycle 86 to 182, the resulting function indicates a slight upwards trend,

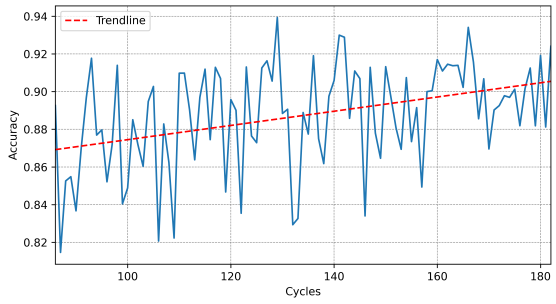

Figure 5: Evolution of the Prediction Accuracy per Cycle After Flattening.

thus, showing an increase in performance:

$$f(x) = 0.000377x + 0.836745\,, \tag{1}$$

where x represents the cycle index and $f(x)$ the corresponding accuracy value. This examined accuracy values together with the trend line is illustrated in Figure 5.

In addition, the results were analyzed vertically, i.e., the accuracy per license plate image instead of the accuracy per cycle. The lower blue curve in Figure 6 visualizes the results across all 182 cycles, with accuracy values arranged in descending order. This allows for a clear visualization of accuracies per license plate, with the highest accuracy located on the left side and the lowest accuracy positioned on the right side of the curve. The accuracy values range from 0.3214 (minimum) to 0.8857 (maximum). The mean accuracy is 0.6835 with a standard deviation of 0.0688, together with $q(25) = 0.6408, q(50) = 0.6889$, and $q(75) = 0.7308$.

Additionally, the upper orange curve in Figure 6 shows the ordered accuracy values per license plate after the model training converged, i.e., accuracy values after cycle 86. The mean value is 0.8863 with a standard deviation of 0.0681. Minimum and maximum values are 0.5556 and 1.0, respectively, with quantiles $q(25) = 0.8421, q(50) = 0.8929$, and $q(75) = 0.9333$. The orange curve exhibits a significant positive y-offset, indicating a performance increase when training is completed.

6.2 Correlation Between Number of Images Trained and Evolution of Accuracy

Considering that the prediction accuracy increases along the x-axis, it can be assumed that there is also a positive linear correlation between the number of images provided to train the model for each license plate number and the accuracy of the corresponding

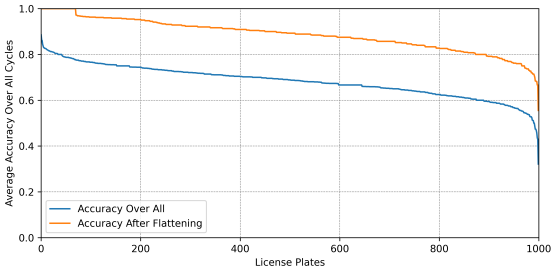

Figure 6: Accuracy Per License Plate.

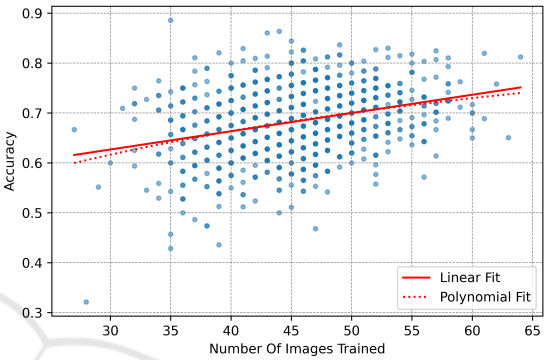

Figure 7: Correlation Between Number of Images and Evolution of Accuracy.

license plate number over the whole dataset, encompassing all cycles. The values are illustrated in Figure 7. The Pearson correlation coefficient shows indeed a weak positive correlation of $r = 0.306$ with a very low p-value of 3.754×10^{-23}, which indicates a strong statistical significance of the result: The linear correlation line can be expressed as

$$f(x) = 0.003657x + 0.517106\,. \tag{2}$$

In addition to evaluating a linear correlation, the data are also subject to a polynomial fit to investigate potential non-linear distributions. Furthermore, a second-order polynomial, as specified by Equation 3, provides a more suitable fit to the data.

$$f(x) = 1 \times 10^{-6}x^2 - 1.14 \times 10^{-4}x + 1.08 \times 10^{-2} \tag{3}$$

The linear and the polynomial fits are illustrated in Figure 7 by solid and dotted lines, respectively.

6.3 Computation Times

During the experiment, prediction and training execution time was recorded. This was done to examine how execution times evolved and whether the input data influenced these durations.

The execution times for predicting the license plates per cycle and per image are illustrated in Figure 8. The mean prediction time across all 182 ex-

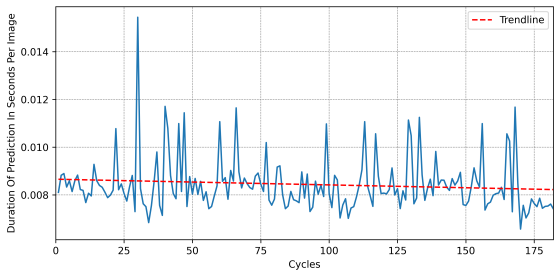

Figure 8: Average Duration of Prediction Per Cycle and Image.

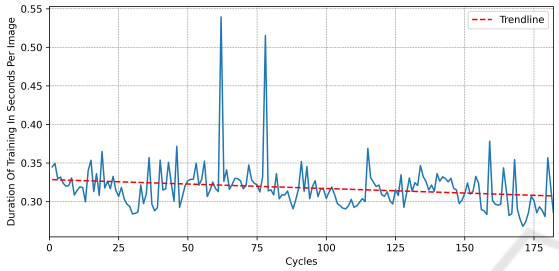

Figure 9: Average Duration of Training Per Cycle and Image.

ecutions is 0.0084 seconds per image, with a standard deviation of 0.0011 s. The minimum duration is 0.0066 s, the maximum duration is 0.0154 s. The quantile values are $q(25) = 0.0078\,\text{s}, q(50) = 0.0082\,\text{s}$, and $q(75) = 0.0087\,\text{s}$. As before, a linear trendline is fitted into the data, resulting in a slight downward trend:

$$f(x) = -0.00000239x + 0.01\,, \qquad (4)$$

where x represents the cycle index and $f(x)$ the corresponding prediction execution time.

The execution times for (re-)training the model per cycle and image are illustrated in Figure 9. The mean training time across all 182 cycles is 0.3178 seconds per image, with a standard deviation of 0.0299 s. The fastest processing took 0.2679 seconds, the slowest one 0.5393 seconds. The quantile values are $q(25) = 0.3013\,\text{s}$, $q(50) = 0.3151\,\text{s}$, and $q(75) = 0.3270\,\text{s}$. As with the prediction execution times, a slight downward trend is observable for the training:

$$f(x) = -0.0001168693x + 0.33\,, \qquad (5)$$

where x represents the cycle index and $f(x)$ the corresponding training execution time.

7 CONCLUSION

This chapter offers a final summary of the study's main findings and arguments, while also highlighting limitations and suggesting areas for future research.

7.1 Summary

The results from our study clearly indicate the effectiveness of machine learning in enhancing Automatic License Plate Recognition (ALPR) systems, particularly under specific constraints like access only to current-cycle images, absence of historical image data, and the requirement for a dynamic model capable of adapting to new license plates. Significantly, our approach, which focuses on recognizing entire license plates rather than individual characters, serves as a complementary extension to existing ALPR pipelines, not a replacement.

The analysis of our model demonstrates a positive linear and polynomial correlation between prediction accuracy and the volume of license plate images used for training. This suggests that the model had not fully converged by the final training cycle, indicating that further training could potentially lead to even greater accuracy.

Our experiment robustly supports the integration of machine learning into license plate recognition processes, especially when this methodology is applied in a post-processing context and focuses on the entire plate. Despite the challenges posed by data limitations, our study managed to achieve high levels of accuracy. This success underscores the capability of machine learning to adeptly manage the complexities inherent in license plate recognition tasks.

A key aspect of our experiment's success is attributed to the holistic analysis of entire license plates rather than individual characters. This comprehensive approach not only yields high recognition accuracy but also provides practical insights for the enhancement of ALPR applications. By incorporating such post-processing algorithms, there is a potential to significantly increase both accuracy and operational efficiency in automated image verification processes.

In conclusion, our experiment conclusively demonstrates the effectiveness of machine learning algorithms in license plate recognition, affirming the added value of considering the entire license plate as a unit for recognition. This finding offers crucial insights for the future development and implementation of ALPR systems, particularly as an augmentative strategy for existing technologies in the field.

7.2 Critical Discussion

The following limitations have been identified and shall be taken into consideration when interpreting the results of the study.

Room for Optimization. The experimental code was unoptimized, focusing only on task feasibility and the research question. Thus, both speed and accuracy can be significantly improved.

Impact of Model Adaptation on Performance. Figure 2 shows that the model saw all license plates by cycle 26, leaving questions about how further changes, like adding output nodes, would impact performance. This study offers initial insights, but more research is needed. Palnitkar and Cannady (2004) discuss methods for adapting DNNs for optimal performance.

Impact of Data Variety. We observed a slight decrease in computation times for both, predictions and re-training, but larger, real-world datasets may show different trends. Increased data variability could also alter computational behavior, suggesting an area for future research.

Implementation of Mechanisms to Prevent Overfitting. The model used the Adam optimizer to minimize overfitting risk, but it does not guarantee prevention. Although it likely did not overfit by the final cycle, real-world or future research should explore additional techniques like early stopping or dropout, as discussed by Steinwendner and Schwaiger (2020).

ACKNOWLEDGEMENTS

Martin Nocker and Pascal Schöttle are supported under the project "Secure Machine Learning Applications with Homomorphically Encrypted Data" (project no. 886524) by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) of Austria.

REFERENCES

Ahmad, I. S., Boufama, B., Habashi, P., Anderson, W., and Elamsy, T. (2015). Automatic license plate recognition: A comparative study. In *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 635–640.

Al-batat, R., Angelopoulou, A., Premkumar, S., Hemanth, J., and Kapetanios, E. (2022). An end-to-end automated license plate recognition system using yolo based vehicle and license plate detection with vehicle classification. *Sensors*, 22(23):9477.

Andersson, J. (2022). A study on automatic license plate recognition. Master's thesis, Åbo Akademi University, Faculty of Science and Engineering.

Bulan, O., Kozitsky, V., and Burry, A. (2015). Towards annotation free license plate recognition. In *IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1495–1499.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2022). A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 44(7):3366–3385.

Du, S., Ibrahim, M., Shehata, M., and Badawy, W. (2013). Automatic license plate recognition (alpr): A state-of-the-art review. *IEEE Transactions On Circuits And Systems For Video Technology*, 23(2):311–325.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135.

Gao, L. and Zhang, W. (2021). Research on license plate detection and recognition based on deep learning. In *International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)*, pages 410–415.

Goncalves, G. R., Diniz, M. A., Laroca, R., Menotti, D., and Schwartz, W. R. (2018). Real-time automatic license plate recognition through deep multi-task networks. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 110–117.

Goncalves, G. R., Menotti, D., and Schwartz, W. R. (2016). License plate recognition based on temporal redundancy. In *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2577–2582.

Hendry and Rung-Ching, C. (2019). Automatic license plate recognition via sliding-window darknet-YOLO. *Image and Vision Computing*, 87:47–56.

Izidio, D. M. F., Ferreira, A. P. A., and Barros, E. N. S. (2018). An embedded automatic license plate recognition system using deep learning. In *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 38–45.

Jiang, Y., Jiang, F., Luo, H., Hongyu, L., Jian, Y., Jiaxin, L., and Jia, R. (2023). An efficient and unified recognition method for multiple license plates in unconstrained scenarios. *IEEE Transactions On Intelligent Transportation Systems*, 24(5):5376–5389.

Jinwook, K., Heeyong, Y., and Min-Soo, K. (2022). Tweaking deep neural networks. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 44(9):5715–5728.

Kaiser, P., Schirrmacher, F., Lorch, B., and Riess, C. (2021). Learning to decipher license plates in severely degraded images. In *Pattern Recognition, ICPR Inter-*

national Workshops and Challenges, Part VI, pages 544–559. Springer Nature Switzerland AG.

Ke, X., Zeng, G., and Guo, W. (2023). An ultra-fast automatic license plate recognition approach for unconstrained scenarios. *IEEE Transactions On Intelligent Transportation Systems*, 24(5):5172–5185.

Kirkpatrick, J., Pascanua, R., Rabinowitza, N., Venessa, J., Desjardinsa, G., Rusua, A. A., Milana, K., Quana, J., Ramalhoa, T., Grabska-Barwinskaa, A., Hassabisa, D., Clopathb, C., Kumarana, D., and Hadsella, R. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Li, H. and Shen, C. (2016). Reading car license plates using deep convolutional neural networks and LSTMs.

Palnitkar, R. M. and Cannady, J. (2004). A review of adaptive neural networks. In *IEEE SoutheastCon - 2004 - Proceedings*, pages 38–47. IEEE SoutheastCon.

Pustokhina, I. V., Pustokhin, D. A., Rodrigues, J. J. P. C., Gupta, D., Khanna, A., Shankar, K., Seo, C., and Joshi, G. P. (2020). Automatic vehicle license plate recognition using optimal k-means with convolutional neural network for intelligent transportation systems. *IEEE Access*, 8:92907–92917.

Schirrmacher, F., Lorch, B., Maier, A., and Riess, C. (2023). Benchmarking probabilistic deep learning methods for license plate recognition.

Selmi, Z., Halima, M. B., and Alimi, A. M. (2017). Deep learning system for automatic license plate detection and recognition. In *14th IAPR International Conference on Document Analysis and Recognition*, pages 1132–1138.

Seunghui, Jang anf Kim, Y. (2021). A fast training method using bounded continual learning in image classification. In *2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 186–191.

Shashirangana, J., Padmasiri, H., Meedeniya, D., and Perera, C. (2021). Automated license plate recognition: A survey on methods and techniques. *IEEE Access*, 9:11203–11225.

Steinwendner, J. and Schwaiger, R. (2020). *Neuronale Netze programmieren mit Python*. Rheinwerk Verlag.

Tarigana, J., Nadiab, Diedanc, R., and Suryanad, Y. (2017). Plate recognition using backpropagation neural network and genetic algorithm. *Procedia Computer Science*, 116:365–372.

Van de Ven, G. M., Siegelmann, H. T., and Tolias, A. S. (2020). Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11:4069.

Špaňhel, J., Sochor, J., Juránek, R., and Herout, A. (2018). Geometric alignment by deep learning for recognition of challenging license plates. In *21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3524–3529.