Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 229 (2023) 149–157

Procedia Computer Science

www.elsevier.com/locate/procedia

12th International Conference Young Scientist Conference on Computational Science (YSC 2023)

CIS Multilingual License Plate Detection and Recognition Based on Convolutional and Transformer Neural Networks

Irek Saitov[a,*], Andrey Filchenkov[a]

[a] *ITMO University, 197101, 49 Kronverksky pr., St Petersburg, Russia*

Abstract

License plate detection and recognition is an urgent topic in traffic regulation due to many practical applications. Globalization and the development of worldwide delivery has contributed to the mixing of vehicles from different countries and regions. This, in turn, imposes additional difficulties for license plate recognition systems due to the difference in plate patterns. The problem of license plate recognition for the CIS countries has not been studied thoroughly. In our study, we propose a license plate detection model based on the YOLOv8 deep convolutional neural network and a multilingual license plate recognition model based on the TrOCR transformer for the CIS countries. The detection model shows a result of 0.983 for mAP@50 metric outperforming baseline in terms of speed and accuracy. The optical character recognition model reaches the best values for the CER metric for the the vast majority of countries of Armenia, Kazakhstan, Ukraine, Moldova, ex-USSR, Kyrgyzstan and also in Europe.

Keywords: license plate recognition ; deep learning; transformer; object detection; OCR;

1. Introduction

License plate recognition systems are in great demand in various practical applications, such as automatic collection of money on toll roads, traffic safety monitoring, access control to closed areas, and many others.

However, despite its popularity, license plate recognition can be quite complicated and requires high accuracy and reliability. One of the main problems in recognizing license plates is their diversity. License plates may vary in size, shape, color, font, and placement of symbols. Some license plates may be written in different languages or have special symbols or serial numbers. This means that license plate recognition systems must be configured to detect and recognize different types of license plates, which complicates the process and requires a lot of resources. Another

* Corresponding author. Tel.: +7-931-214-4420.
E-mail address: iasaitov@itmo.ru

1877-0509
10.1016/j.procs.2023.12.016

problem is the variability of lighting and weather conditions. With insufficient lighting, license plates may be difficult to detect, and bright sunlight or bad weather such as rain or fog might make them blurry or distorted.

All these problems make it difficult for a person to process incoming information, for example, from video cameras about license numbers in real time. Automated recognition is based on machine learning methods, which allow the system to learn recognizing license plates in various images and adapt to different lighting and environmental conditions. For more accurate recognition of license plates, additional information sources can also be used, such as license plate databases, information about the model and manufacturers of the vehicle, as well as location and time data. This allows the system to refine the recognition results and reduce the number of false positives. Despite the use of modern technologies, the recognition systems of number characters still cannot guarantee the highest recognition accuracy. Therefore, when transmitting information about license plates to law enforcement agencies, it is necessary to take into account the possibility of errors and additionally check the information received.

A standard approach to license plate recognition consists of the following steps:

1. License plate detection (bounding box / segmentation),
2. Character segmentation,
3. Optical character recognition (OCR).

License plate detection extracts bounding box or segmentation information about the location of the license plate in the image. In the next step, the position of each character is segmented using the resulting figure. In the final step, an OCR model tries to recognize each found character and return the resulting sequence.

Although transformer neural networks have proven to be effective in the field of natural language processing, their application in computer vision is still poorly understood. In image processing, the attention mechanism (a key component of the transformer architecture) is usually used in addition to the convolutional neural network (CNN) or replacing some of its parts. One of the main limitations of using the transformer architecture is the required amount of training data during training [8].

In addition, there is a problem with the lack of data for CIS license plate recognition. Most previous approaches utilized only one country data source and also they used some privately collected data, which does not allow comparison. One of the solutions is Nomeroff Net [3]. Nomeroff Net is an open-source Python license plate recognition framework based on the application of an object detection YOLOv5 [23] neural network and customized OCR module powered by GRU architecture. The Nomeroff Net approach was developed specifically for the target CIS benchmark and is considered as a baseline.

In our study, we propose a license plate localization model based on the YOLOv8 convolutional neural network and a multilingual license plate recognition model based on the TrOCR transformer model. The investigation of license plate recognition was made for the countries that previously belonged to CIS, such as Russian Federation, Kazakhstan, Armenia, Georgia, Kyrgyzstan, Moldova, Ukraine, and also for old license plate designs of ex-USSR.

2. Related works

In recent years, many approaches to solving the problem of recognition of characters on an image have been proposed. One of the challenges for license plates is the presence of noise in the image, which can be caused by various factors such as traffic, relief of the road, or obstacle on the road. Noise can significantly hinder the process of recognition of license plates as the system can erroneously identify the noise as part of the license plate. In addition, some license plates can be partially or completely hidden by other objects, such as other cars, trees or buildings. This creates additional difficulties for the license plate recognition system as it must be able to distinguish license plates from other objects in the image. To solve these problems, license plate recognition systems use different image processing techniques, such as noise filtering [19], improving the contrast, and finding the contours of objects.

The second stage of analysis in automatic license plate recognition systems (ALPR) is usually the segmentation of individual characters on an already allocated car number. For this, the construction of a vertical projection of a license plate previously reduced to black and white is most often used [11]. However, with the development of neural models in computer vision, the process of segmenting individual characters has become part of the OCR model [13].

Neighboring character information has proven to be important in sequence processing as character order plays an important role in language processing.

The use of neural models has simplified the task for researchers. The need for manual feature extraction, and artificial image quality improvement as in work [2] has disappeared. This allowed to speed up the development of models and make them universal, suitable for most real-life scenarios.

Convolutional neural networks have proven their good performance. Moreover, in many modern ALPR systems [20, 24], these models are part of the system. However, in recent years the development of transformers has shown its superiority for many tasks, including computer vision and language processing. For the task of recognizing characters on a license plate, the use of a full-fledged model, as the authors know, has not been studied.

3. Method

3.1. Detection model

The YOLO model is an effective neural net that balances speed and accuracy. In recent years many upgrades [5, 23, 10] have come to life with the development of this model. Originally it was designed for object detection, but now it is also leveraged for classification, semantic segmentation, object tracking, and pose estimation. In this paper, a detection model is used to detect and extract the location of license plates. YOLO models are related to one-stage detectors, which include object localization and its classification in one neural step. At the same time, this model achieves SOTA results on many well-known datasets and benchmarks [15].

The original YOLO object detector was trained on MS COCO dataset, in which the license plate class is absent. Thus, an approach is proposed for fine-tuning the model on the relevant dataset using the transfer learning method [21]. Since the 5th version of YOLO its domain adaptability increased, which allows to apply this approach successfully.

For the most part, the YOLOv5 model inherits the neural network basis from the 4th version [5]. The enhancements are mainly related to improving representativeness and adaptability. In particular, this can be observed when using this model in the fine-tuning scenario. An adaptive anchor box selection process, which is called "autoanchor", was introduced. Instead of predefined and customized on COCO dataset anchors, they try to fit to new data by using a genetic algorithm. This behavior made it possible to neutralize the difference in the distribution of aspect ratio between the source and target datasets. Another enhancement was adding different augmentation techniques, such as mosaic augmentation. It consisted in combining the images from the 4th version, including all the objects contained in them. The resulting image was treated as a new input image and used for training. This improvement increased the representativeness of the model and improved the ability to recognize small objects.

Currently, YOLOv8 is one of the latest releases. It shows SOTA results for several tasks and suits well for domain adaptation. The great impact of the 8th version is performance improvement. The structural convolution block has changed, in which the sizes of the cores and elements order have been changed, thereby reducing the number of parameters and increasing efficiency. The anchor-free detection system was developed allowing to predict directly the center of an object instead of the offset from a known anchor box. In addition, mosaic augmentation fixes with closing it for the last 10 epochs on training to correct the distribution that the model learns.

3.2. OCR model

The main method for optical character recognition from detected license plates is TrOCR [14]. Authors proposed to use pre-trained transformer-based CV and NLP models for character recognition (BEiT and RoBERTa respectively) from images. Its training took place in unsupervised mode on a large synthetically generated Wikipedia corpus of raw text images. The model consists of encoder and decoder models. The encoder model takes an image as input and returns its hidden representation. Decoder takes obtained representation and tries to restore written characters. TROcr has different pretrained versions. In the end, the model was retrained for one of two subtasks: recognition of printed or handwritten text. Stage 1 is mostly suited for adapting to different tasks.

The task of license plate recognition refers to optical character recognition. One of the main features is that the sequence of characters on the plate does not represent words as for language models. The TrOCR was trained in unsupervised learning mode and the outputs of this model are not tied to any dictionary. It allows applying this model to many downstream tasks. Our proposal is to leverage TrOCR transformer model pretrained on large-scale data and fine-tune it for license plate recognition.

4. Experiments

4.1. Data and Preprocessing

Two independent data sets were used for license plate detection and recognition. Data was taken from [3] and named as Nomeroff Net dataset. Datasets were split into training, validation, and testing sets. A validation set was used for controlling target metrics during training and saving the best weights. Values on test sets are provided as a result. All splits were taken as it is from original source for fair comparison.

Fig. 1. Vehicles examples with license plates from Nomeroff Net dataset. Photos taken under different environments. In many samples, the camera viewpoint transforms the perspective projection of license plates in the image.

The detection dataset (see Fig. 1) has 11384 images. Images have different vehicles and different camera views. Data were split for training, validation, and testing as 7968 (70%), 1708 (15%), and 1708 (15%) instances respectively. For YOLO each instance was resized to 640x640 size. For augmenting setup scaling, color space adjustments, and mosaic [5] transforms. Labels had only one class as license plate appearance, represented by a non-normalized bounding box.

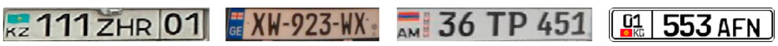

Fig. 2. License plates examples from Kazakhstan, Georgia, Armenia, and Kyrgyzstan. Each country has an original design, where the font, position and the order of characters differ.

OCR datasets (see Fig. 2) are represented by normalized license plate images from different countries. The work is focused on the countries earlier belonged to CIS: Russia (Ru), Kazakhstan (Kz), Armenia (Am), Georgia (Ge), Kyrgyzstan (Kg), Moldova (Md), Ukraine (Ua), and old license plates of ex-USSR (Su). Also, there was Europe (Eu). Datasets consisted of 9k to 125k images and were also split into training, validation, and testing (see Table 1).

For TrOCR-based experiments, license plates were resized to 384x384 size. For training from scratch, images were used in 224x224 resolution as it took much longer time for training. To enhance data utilization, various augmentations

Table 1. Number of samples according to the split sets for each available license plate OCR dataset from Nomeroff Net.

Country	Short name	Train	Valid	Test
Kazakhstan	Kz	8642	1001	279
Moldova	Md	10531	789	849
Armenia	Am	11524	565	620
Georgia	Ge	24986	738	2777
Kyrgyzstan	Kg	29204	1645	1068
ex USSR	Su	35310	1874	1618
Europe	Eu	42738	1359	1531
Russia	Ru	49382	4893	2845
Ukraine	Ua	121721	2146	2304

were performed. One of six kinds of image transformations (random rotation (-10 to 10 degrees), Gaussian blurring, image dilation, image erosion, downscaling, and underlining) or keeping the original image are taken for each dataset. We randomly decided which image transformation to take with equal possibilities for each sample.

For some experiments, "square" preprocessing was applied. The idea was to negotiate the effect of vertical distortions of license plates when it resized to model appropriate input. Most license plate images were of the width greater than height and after resizing they would have a vertical elongation of letters. The squaring operation involved adding equal white indentations at the bottom and top of the number plate image and adding the size of the image to square one. Thus, the length and height ratios of letters remained equal after resizing.

4.2. Evaluation metrics

The evaluation on license plate detection experiments was performed using mAP@50 metric, which is popular in many object detection applications [15, 9]. Mark @50 means detection predictions are considered as TP if $IoU \geq 0.5$. With this restriction, the precision-recall curve is built, where precision is interpolated (Eq. 2) by each point. Area under this curve represents AP (Eq. 1). Mean average precision is calculated by averaging AP over all classes.

$$AP = \sum (r_{n+1} - r_n) p_{interp}(r_{n+1}), \tag{1}$$

where

$$p_{interp}(r_{n+1}) = \max_{\hat{r} \geq r_{n+1}} p(\hat{r}). \tag{2}$$

For optical character recognition on license plates, the character error rate (CER) metric is used since it focuses on character-level errors. Consider reference as a ground true string and prediction as a predicted string. Any reference can be restored for a certain number of operations of insertions, deletions, and substitutions. CER metric (Eq. 3 aggregates the number of necessary transformations, normalizing by the number of characters in the prediction.

$$CER = \frac{I + D + S}{N}, \tag{3}$$

where I is a number of insertions, D is a number of deletions, S is a number of substitutions, N is a number of characters in the prediction.

4.3. Settings

All experiments were performed on NVIDIA TITAN RTX (24 GB of RAM). License plate detection models were built on Ultralytics [10]. OCR models were built on HuggingFace Transformers library [25].

As a localization model, YOLO family detection algorithms were reviewed. For a fair comparison, YOLOv5 from baseline was retrained with default hyperparameters. Initial weights for both YOLOv5 and YOLOv8 models were taken as pretrained on MS COCO dataset [15]. The size of the model is 's' small, which corresponds to 7.2M parameters. For both models batch size 32 was set. Input image size was taken 640x640. YOLO models were trained for 100 epochs after which the best weights were saved by minimum loss value on the validation set. Other hyperparameters were taken as default.

For optical character license plate recognition two setups TrOCR fine-tuning and encoder-decoder transformers training were used. Original TrOCR was trained on large-scale synthetic data generated from English Wikipedia texts. Only the variant which incorporates the computer vision transformer BEiT [4] model and language transformer RoBERTa [16] model was considered. Different sizes of TrOCR are considered as 'base' and 'large'. TrOCR fine-tuning processed for 1500 or 4000 steps. For each OCR model, 4 batch size is set, except for the large TrOCR model where the batch size is 2 (because of GPU memory utilization).

Encoder-decoder transformer was similar to TrOCR model structure, where the vision transformer model was used as the encoder and the language transformer model as the decoder. Scratch training means that model was not previously trained for any OCR task, but encoder and decoder weights were taken as pretrained. Training for encoder-decoder models was done for 9000 steps.

For all the experiments, after the end of the training process the best model weights for the loss function values on the validation set were saved for further use for testing on the corresponding data subset.

4.4. License plate localization experiments

Different setups for training were investigated. Non-pretrained model variants showed poor results in early experiments, which allows to conclude about the effectiveness of the transfer learning technique for the task of detecting license plates. For both YOLOv5 and YOLOv8 training was done with and without augmentation. However, the results during training and at the end of the test were almost the same. Therefore, report experiments are presented in a non-augmented mode.

Fig. 3. Examples of successful license plates detected on various vehicles. Recognized license plate highlighted with a green bounding box. The trained YOLOv8 model copes with license plates of motorcycles, perspective distortions, as well as very remote objects.

Table 2. Results by mAP@50 metric of YOLO model fine-tuning for license plate localization task. The best result achieved on the test set after 100 epoch training is shown in bold

method	mAP@50	Inference time
YOLOv5	0.951	8.7ms
YOLOv8	**0.983**	**3.3ms**

For the license plate detection on Nomeroff Net dataset, YOLOv8 model exceeded (see Table 2) the baseline model result and amounted to 0.983 by $mAP@50$ metric. Visual tests which can be seen in Fig. 3 have shown that the trained YOLO model can cope with difficult cases of license plate detection. In terms of speed, this model also proved to be more efficient.

4.5. Optical character recognition experiments

The first experiments were done using only the Russian dataset (see Tables 3,4) from Nomeroff Net since using all data required a significant amount of time. Different pretrained weights were used for TrOCR model. TrOCR model has two versions of fine-tuning for printed and handwritten texts. As a rule, the font for license plates is “printed”. Therefore, it was proposed to use the same size TrOCR model fine-tuned for printed texts. However, as shown in Table 3, “stage1” performs better than the printed version, which demonstrates that pretraining in an unsupervised manner leads to better domain adaptation and the printed fonts were far from matching the license plates ones.

Table 3. Testing results of TrOCR fine-tuning on license plate recognition task by CER metric for ”Ru” dataset OCR experiments. The best result is shown in bold

Method	Step num	Model size	Pretraining	Aug	Tune	CER
Nomeroff app						**0.00050**
TrOCR	1500 steps	base	stage1			0.00075
TrOCR	1500 steps	base	printed			0.00084
TrOCR	1500 steps	base	stage1	augmented		0.00063
TrOCR	4000 steps	base	stage1	augmented		0.00485
TrOCR	1500 steps	base	stage1		tuned	0.00121
TrOCR	1500 steps	large	stage1			0.00134

Table 4. Vision transformer encoder-decoder scratch testing results by CER metric for ”Ru” dataset OCR experiments.

Method	Encoder	Decoder	Num steps	Extra params	CER
Scratch	encoder(ViT224)	decoder(en-BERTc)	9000 steps	square, lower	0.00472
Scratch	encoder(ViT224)	decoder(deepavlov-BERTc)	9000 steps		0.00430
Scratch	encoder(ViT224)	decoder(geo-BERTc)	9000 steps		0.04516

The large size of TrOCR model was trained with a batch size of 2, which in our opinion led to less accurate results than the base size model. That is batch size hyperparameter is important, however, with the available resources, this study did not seem feasible. Hyperparameter tuning involved “weight decay”(0 to 0.1), “warmup ratio”(0 to 0.5), “learning rate”(1e-6 to 1e-4), “max steps”(500 to 1500). The search for this process is performed by the population-based training algorithm from the ray [17] framework. Hyperparameters tuning for TrOCR base model showed on validation data almost ideal result but on testing it still could not obtain the best result. Large step numbers also degenerated results in a test setting. This seems with steps noticed like this model sensible to overfitting.

As TrOCR represents an encoder-decoder model we tried (see Table 4) to replace its parts with other pretrained modules. All experiments with decoder replacement from ViT [8] to BEiT or DEiT [22] on training from scratch [22] failed as it did not converge during training. As encoder replacement was taken BERT cased [7] with square preprocessing and lowering predicted and labeled symbols.

In addition, as a decoder for the Russian dataset, BERT-based models trained to recognize characters in Russian texts were considered. The first model is BERT from DeepPavlov laboratory (deeppavlov-BERTc) [12] which was trained on the Russian part of Wikipedia and news data. Next is BERT from geotrend (geo-BERTc) [1] which is trained for a small set of languages on the XNLI dataset [6]. One of the 15 languages is Russian. These models were combined with the ViT encoder. CER error was minimal for deeppavlov-BERTc decoder model. However, the results of methods (see in Table 4) couldn’t exceed baseline or TrOCR fine-tuning. This indicates that the available license plate data is not enough for full-fledged training from scratch.

The following experiments (see Table 5) were carried out with all the remaining datasets (Am, Eu, Kg, Kz, Su, Ge, Ua, Md). Models that performed well on the Russian dataset were tested on all other datasets. Lowering characters and squaring techniques did not perform well and lost for the rest training setups. It indicates that TrOCR was already pre-trained with vertical distortions and is case-sensitive. For most datasets (Kazakhstan, Armenia, ex-USSR, Ukraine, Europe), base size TrOCR taken with pretrained weights from stage 1 trained with augmentations for 1500 steps

Table 5. Results by CER metric on test data for other (Am, Eu, Kg, Kz, Su, Ge, Ua, Md) datasets OCR experiments. The best results are shown in bold.

Country	Nomeroff app	TrOCR base- stage1, 1500 steps	TrOCR base-stage1, augmented, 1500 steps	TrOCR base-stage1, augmented, 1500 steps, square, lower
Kz	0.00454	0.00499	**0.00136**	0.00181
Md	0.00195	**0.00019**	0.00058	0.00039
Am	0.00069	0.00069	**0.00023**	0.00069
Ge	**0.00119**	0.00549	0.00281	0.00233
Kg	0.00267	**0.00069**	0.00214	0.00253
Su	0.00406	0.01094	**0.00273**	0.00361
Eu	0.00576	0.01359	**0.00490**	0.00726
Ua	0.00152	0.00211	**0.00092**	0.00211

proved to be the best model. TrOCR without augmentation outperformed on Moldova and Kyrgyzstan, however, augmented took second place for this data with a small gap. Thereby proposed TrOCR model fine-tuning showed the best result on 7 out of 9 datasets.

5. Conclusion and future work

The proposed model can be employed as the major element of intelligent infrastructure like toll fee collection, parking management, and traffic surveillance. An extensive experimental investigation was conducted using Nomeroff Net datasets for license plate localization and optical character recognition. The powerful detection model shows a result of 0.983 for mAP@50 metric in the fine-tuning scenario while surpassing the baseline in terms of speed and performance. The optical character recognition model reaches the best values by the CER metric for Armenia, Kazakhstan, Ukraine, Moldova, ex-USSR, Kyrgyzstan, and Europe. Thus, confirming the effectiveness of fine-tuning a full-fledged transformer for the task of license plate recognition.

In the future, we are planning to develop a full-fledged automatic license plate recognition (ALPR) system for the CIS countries. For this purpose, it requires automatic OCR model selection for a specific country. In addition, to unify the template of objects, it is required to develop an algorithm for normalizing perspective distortions of license plates. Another possibility is to expand the database with additional data. Since there are no CIS external benchmarks for license plate recognition, with the exception of Nomeroff Net, it is possible to collect data via Internet requests - for example, from [18] containing more than 40k samples for each country in question.

6. Code and data avaliability

Code to reproduce the experiments described in the paper is available at provided repository: `https://github.com/CaptainFest/LP_CIS`. The Nomeroff Net dataset [3] was used as data.

References

[1] Abdaoui, A., Pradel, C., Sigel, G., 2020. Load what you need: Smaller versions of multilingual bert, in: SustaiNLP / EMNLP.

[2] Ashtari, A.H., Nordin, M.J., Fathy, M., 2014. An iranian license plate recognition system based on color features. IEEE transactions on intelligent transportation systems 15, 1690–1705.

[3] AUTO-RIA, . Nomeroff net license plate recognition framework. URL: `https://github.com/ria-com/nomeroff-net`.

[4] Bao, H., Dong, L., Piao, S., Wei, F., 2021. Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 .

[5] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 .

[6] Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S.R., Schwenk, H., Stoyanov, V., 2018. Xnli: Evaluating cross-lingual sentence representations. arXiv preprint arXiv:1809.05053 .

[7] Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 .

[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 .
[9] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., . The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.
[10] Jocher, G., Chaurasia, A., Qiu, J., 2023. YOLO by Ultralytics. URL: https://github.com/ultralytics/ultralytics.
[11] Kakani, B.V., Gandhi, D., Jani, S., 2017. Improved ocr based automatic vehicle number plate recognition using features trained neural network, in: 2017 8th international conference on computing, communication and networking technologies (ICCCNT), IEEE. pp. 1–6.
[12] Kuratov, Y., Arkhipov, M., 2019. Adaptation of deep bidirectional multilingual transformers for russian language. arXiv preprint arXiv:1905.07213 .
[13] Laroca, R., Severo, E., Zanlorensi, L.A., Oliveira, L.S., Gonçalves, G.R., Schwartz, W.R., Menotti, D., 2018. A robust real-time automatic license plate recognition based on the yolo detector, in: 2018 international joint conference on neural networks (ijcnn), IEEE. pp. 1–10.
[14] Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F., 2021. Trocr: Transformer-based optical character recognition with pre-trained models. arXiv preprint arXiv:2109.10282 .
[15] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, Springer. pp. 740–755.
[16] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 .
[17] Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M.I., et al., 2018. Ray: A distributed framework for emerging {AI} applications, in: 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18), pp. 561–577.
[18] PlatesMania, . International database for vehicles license plates. URL: https://platesmania.com.
[19] Puranic, A., Deepak, K., Umadevi, V., 2016. Vehicle number plate recognition system: a literature review and implementation using template matching. International Journal of Computer Applications 134, 12–16.
[20] Pustokhina, I.V., Pustokhin, D.A., Rodrigues, J.J., Gupta, D., Khanna, A., Shankar, K., Seo, C., 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.
[21] Torrey, L., Shavlik, J., 2009. Transfer learning. handbook of research on machine learning applications. IGI Global 3, 17–35.
[22] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H., 2021. Training data-efficient image transformers & distillation through attention, in: International conference on machine learning, PMLR. pp. 10347–10357.
[23] Ultralytics, . Yolov5. URL: https://doi.org/10.5281/zenodo.3908559.
[24] Wang, Y., Bian, Z.P., Zhou, Y., Chau, L.P., 2021. Rethinking and designing a high-performing automatic license plate recognition approach. IEEE Transactions on Intelligent Transportation Systems 23, 8868–8880.
[25] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A.M., 2020. Transformers: State-of-the-art natural language processing, Association for Computational Linguistics. pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.