

Multi-national and Multi-language License Plate Detection using Convolutional Neural Networks

Mohammed Salemdeeb

Electronics & Telecommunications Engineering Department
Kocaeli University
Kocaeli, Turkey
en_mis@hotmail.com

Sarp Erturk

Electronics & Telecommunications Engineering Department
Kocaeli University
Kocaeli, Turkey
sertur@kocaeli.edu.tr

Abstract—Many real-life machine and computer vision applications are focusing on object detection and recognition. In recent years, deep learning-based approaches gained increasing interest due to their high accuracy levels. License Plate (LP) detection and classification have been studied extensively over the last decades. However, more accurate and language-independent approaches are still required. This paper presents a new approach to detect LPs and recognize their country, language, and layout. Furthermore, a new LP dataset for both multi-national and multi-language detection, with either one-line or two-line layouts is presented. The YOLOv2 detector with ResNet feature extraction core was utilized for LP detection, and a new low complexity convolutional neural network architecture was proposed to classify LPs. Results show that the proposed approach achieves an average detection precision of 99.57%, whereas the country, language, and layout classification accuracy is 99.33%.

Keywords—license plate detection; license plate classification; LPD; Yolo detector; convolutional neural network; deep learning

I. INTRODUCTION

Object detection and classification has attracted a lot of research the recent years, with the advancements in vision technology, computer technology, and deep learning algorithms [1]. Object detection aims to estimate the location of objects of interest contained in an image, while object classification aims to categorize an object within a certain number of categories [2]. Traditional object detection and classification approaches have three steps, namely informative region selection, feature extraction and classification. In region selection, it is possible to scan the entire image using a multi-scale sliding window, as numerous objects may appear in different locations with various sizes and aspect ratios [1]. Feature extraction aims to obtain visual features providing a semantic and robust representation. Some popular feature extraction methods used in the literature are Haar-like features [3], Scale-Invariant Feature Transform (SIFT) [4], Histogram of Oriented Gradients (HOG) [5], and hybrid feature selection techniques [6]. Classification aims to assign a target object in one of many categories. Traditional classification approaches include Supported Vector Machine (SVM) [7], AdaBoost [8], and Deformable Part-based Models (DPM) [9]. Recent breakthroughs in Convolutional Neural Network (CNN)-based

approaches [10] attracted researchers to use Regions with CNN (R-CNN) features for object detection [11]. CNN-based methods have the capacity to learn complex features with deeper architectures and utilize training algorithms to learn informative object representations without the need to design the features manually [12]. Furthermore, researchers studied extensively various CNN models such as AlexNet [10], VGG [13], GoogLeNet [14], ResNet [15], and FDRNet [16] to improve the accuracy of classification and regression problems in machine learning. Generic object detection refers to the detection of objects from predefined classes obtaining the spatial location (e.g. bounding box) inside an image. It can typically be categorized into two types, namely regression/classification and region-based methods [17]. Region-based methods include R-CNN [11], Fast R-CNN [18], Faster R-CNN [19] and Mask R-CNN [20]. On the other hand, regression/classification-based methods include YOLO (You Only Look Once) [21], SSD [22], YOLOv2 [23] and YOLOv3 [24].

Automatic License Plate Recognition (ALPR) is a group of techniques that use License Plate Detection (LPD), character segmentation, and character recognition on images to identify vehicle LP numbers. ALPR is also referred as License Plate Detection and Recognition (LPDR). ALPR is used in various real-life applications such as parking systems, electronic toll collection, and traffic security and control [25]. State-of-the-art object detection algorithms based on deep learning have provided promising results for LP country and layout classification. However, the multi-orientation and multi-scale nature of LPs in addition to distortion and illumination issues, make LPD a challenging task to perform [26]. LPD using deep learning has been extensively studied over the last decade. Authors in [27] proposed the use of a CNN-based Multi-Directional (MD)-YOLO framework for LPD, but their method does not successfully detect small LPs. In [28] a faster R-CNN approach was presented, detecting at first vehicle regions and then locating the LP in each vehicle region. Its performance evaluation results showed 98.39% precision and 96.83% recall. A new approach was proposed in [29], referred as YOLO-L, where the prospective number and size of LP candidate boxes are selected using “k-means++” clustering with a modified YOLOv2 model and pre-identification to distinguish LPs from similar objects. This method achieved a precision of 98.86%

Corresponding author: Mohammed Salemdeeb

and a recall of 98.86%. Researchers in [30] introduced the largest Brazilian LP dataset, referred as UFPR dataset, and proposed a four stage LPDR system comprising of vehicle detection, LPD, character segmentation, and character recognition. The LPD stage used CR-CNN core fast-YOLO, obtained a recall of 98.33%. Furthermore, researchers in [31] introduced a large and comprehensive Chinese LP dataset called CCPD, and proposed an end-to-end LPDR system using RPNet in the LPD phase, comparing the detection of Average Precision (AP) results to SSD, YOLOv2, and Faster R-CNN detection techniques by using 250k unique car LPs.

On the other hand, little research has been performed on multi-language and multi-national LP detection, mostly due to the lack of international LP datasets. Nevertheless, a few recent studies focused on developing a global end-to-end ALPR system, as reported in [32]. Authors in [32] proposed an approach for multi-national license plate detection for images with complex backgrounds, in which the YUV color space was initially used for detecting the rear vehicle lights, and the LP area was detected using a histogram-based approach on the edge energy map. The utilized dataset comprised of LPs from America, China, Serbia, Italy, Pakistan, United Arab Emirates (UAE), and Hungary. The dataset had only single-line LPs and obtained a detection accuracy of 90%. Researchers in [33] used VGG with LSTM to classify the registration country of LPs from Latvia, Lithuania, Estonia, Russia, Sweden, Poland, Germany, Finland, and Belarus. A recent research used tiny YOLOv3 to detect LPs from South Korea, Taiwan, Greece, USA, and Croatia [34]. Several approaches expressed interest in multi-national LPs, but they tested their detectors on each country's dataset separately, rather than accumulating them into one dataset [35-38]. Moreover, multi-language LPs were addressed in a few approaches. Authors in [38] proposed a mask R-CNN detector for LPs with English and Arabic characters from USA and Tunisia. In [39] Korean and English LPs were targeted, using the term multi-style detection to refer to different country, language and one or two-line LP styles. Most of the reported researches studied the LP Classification (LPC) problem inside the LPD stage. In these cases, the detector determines the bounding box, and at the same time gives the class label of an LP. However, in [32, 37] multi-national LPD was presented by just detecting LPs, without providing any other information for nation, language or layout. In [33] the classification of detected LPs by the issuing country was studied, reporting a classification accuracy of 92.8%. On the other hand, authors in [39] proposed a module to classify the detected LPs to single and double-line, without reporting its accuracy but only the entire system results.

In this paper, multi-national LPs from USA, Europe (EU), Turkey (TR), UAE and Kingdom of Saudi Arabia (KSA) are targeted, using YOLOv2 detector with ResNet50 feature extraction for LPD. For this purpose, a new dataset, named as LPDC2020, was constructed and presented. After the segmentation of the detected LPs, a CNN was used to detect the country, language and the one or two-line layout of the LP. The proposed detector and classifier were also tested on several benchmark datasets from those countries, in addition to LPDC2020. The proposed approach aims to close the gap in multi-national, multi-language and multi-layout LP detection

problem, by utilizing a single unified system, and to the best of our knowledge it is the first and only study incorporating LPs from North and South America, Europe, and Middle East (TR, UAE and KSA).

II. DATASETS

A. LP Datasets Available in the Literature

Most of the frequently used LP datasets utilized in previous researches are available online, and their details are summarized in Table I. Any private datasets, not publicly accessible, are disregarded.

TABLE I. A SUMMARY OF PUBLICLY AVAILABLE LP DATASETS

Dataset	Year	# of images	Accuracy %	Country
Caltech [40]	1999	126	-	USA
Zemris [41]	2002	510	86.2	Croatia
UCSD [42]	2005	405	89.5	USA
Snapshots [43]	2007	97	85	Croatia
Medialab [44]	2018	730	-	Greece
Reld [45]	2017	77k	96.5	Czech
UFPR [30]	2018	4500	78.33	Brazil

B. LPDC2020 Dataset

This paper introduces a new LP dataset, which was collected manually using mobile cameras in Turkey, named LPDC2020. It has two image sets: vehicular images to train the LPD module, and cropped LP images to train the LPC module. In addition, due the lack of publicly available Arabic LP datasets, images for KSA and UAE LPs available in the internet were used. All images were processed and annotated manually in a labor-intensive process. Table II shows the number of LPD images collected for each country. Some sample LPs from different countries with one and two-line layouts included in the dataset are shown in Figure 1. Table III shows the structure of LPDC2020 classification dataset. It is noted that, taking one and two-line layouts into account, the LPC dataset incorporates 11 different characteristics. The total number of cropped LP images is 29030, containing LP images from the previously mentioned countries.

TABLE II. A SUMMARY OF LPDC2020 LPD DATASET

Country	TR	EU	USA	KSA	UAE	Total
# of images	4182	2636	715	1000	488	9021



Fig. 1. Some sample LPs from different countries with various layouts.

TABLE III. STRUCTURE OF THE LPDC2020 LPC DATASET

Country	Language of characters	Layout	Number of instances
BR	Latin	One-Line	3714
		Two-Line	900
UAE	Arabic	One-Line	500
		Two-Line	276
EU	Latin	One-Line	5296
		Two-Line	4350
KSA	Arabic	One-Line	290
		Two-Line	792
TR	Latin	One-Line	7771
		Two-Line	3560
USA	Latin	One-Line	1401

III. FUNDAMENTALS OF CNN

The fundamentals of any CNN are the convolutional layers consisting of learnable filters having small spatial size and specific depth. For an input image I and kernel K the general equation of 2D convolution [46] used in computer vision and machine learning is defined as:

$$(I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (1)$$

with i and m being the row indexes, while j and n are the column indexes. The activation layer produces an output value of the neuron using certain activation functions for a given input value. An example is the Rectified Linear Unit (ReLU) [10], where the output will be zero for negative input values and same as the input in any other case. The second important part of CNN is the pooling layer, which is responsible for reducing the input's spatial size by keeping the most important activations. This reduces the amount of computations and the number of learnable parameters. A dropout layer is used to combat overfitting, omitting randomly some neurons in each training step by setting their activation values to zero. As a result, the network can learn using a random combination of neurons. The Fully Connected (FC) layer, also called as dense layer [47], is the third important part of CNNs. Each neuron in the input layer is connected to all output neurons of this layer. The purpose of the FC layer is to learn for non-linear combinations of features. For x neurons input, learnable weight matrix W , and learnable bias vector b , the output of the fully connected layer y can be expressed as:

$$y = Wx + b \quad (2)$$

At the end of the architecture, i.e. after the last fully connected layer, a Softmax layer is used. This layer is used for classification problems, providing a probabilistic interpretation of the input with respect to the sum of all input exponentials, declared as:

$$\text{softmax}(x)_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (3)$$

This layer is also called the loss function layer, since during training a loss function is applied at the end of the CNN. In general, for N samples, the Mean Square Error (MSE) can be used in object detection as in (4) and cross-entropy function is used for classification problems as in (5) [47]:

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4)$$

$$L_{\text{cross-entropy}} = - \sum_{i=1}^N \{y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)\} \quad (5)$$

where, y_i is the i -th actual output, and \hat{y}_i is the i -th predicted output.

IV. PROPOSED APPROACH

This research addresses two problems; The detection of an LP in an image, and the classification of the detected LP's country, language, and layout.

A. License Plate Detection

The proposed approach is based on using the YOLOv2 detector with the ResNet50 [15] network as the core CNN for the LP detector. The utilized ResNet50 architecture is displayed in Table IV.

TABLE IV. RESNET50 ARCHITECTURE

Layer	Size	Filters
Input	224 × 224 × 3	-----
Conv1	112 × 112 × 64	7 × 7 × 64, stride 2
Max pooling	56 × 56 × 64	3 × 3 max pool, stride 2
Conv2	56 × 56 × 256	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
Conv3	28 × 28 × 512	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
Conv4	14 × 14 × 1024	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
Conv5	7 × 7 × 2048	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
Average pooling	1 × 1 × 2048	7 × 7
Fully connected	1 × 1 × 1000	1000
Softmax	1 × 1 × 1000	-----

The input layer size of ResNet50 was redesigned to be 672×672 instead of the original 224×224 pixels. The original size did not provide adequate features for LPD. For an original vehicular image with small size it will be difficult to detect the LP region after reducing its resolution. Naturally, there is a restriction on the minimum LP size required inside the detector's input image, due to the network forward propagation size of ResNet50, which is 224/7=32. Hence, LPs sized 32×32 pixels will correspond to a single point in the output feature map and consequently, any smaller regions will vanish. The proposed detector core network was designed to have a forward propagation size of 672/42=16. The first 40 layers of ResNet50 were used in the proposed YOLOv2 core CNN. The input size was set to 672×672 pixels, and the output feature map was 42×42 pixels. The minimum LP size was set to 16×16 pixels. It should be noted that smaller LPs can still be detected but with lower precision. In addition, the proposed approach can detect LPs sized up to 670×670 pixels. Figure 2 shows the block diagram of the proposed approach. The proposed detector had 27992604≈28M total learnable parameters.

The YOLOv2 detector divides the input image to an $S \times S$ grid, where S is the output feature map size of the YOLOv2 core Resnet40 (i.e. the output of Conv4 layer), and S was set to 42. Anchor boxes were downsized by forward propagation size. YOLOv2 uses A anchor boxes to predict objects. The

detection results are the bounding boxes and the confidence scores, so that for C class probabilities [23] the number of filters is given by:

$$\text{number of filters} = (C + 5) \times A \quad (6)$$

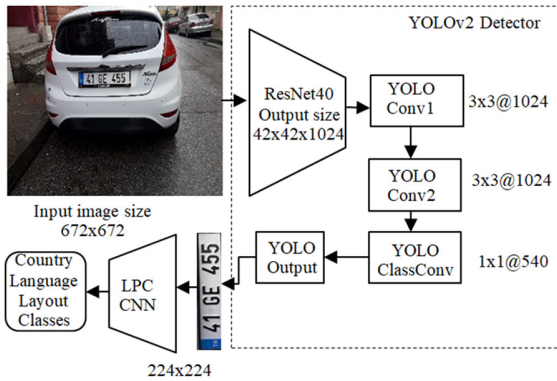


Fig. 2. Block diagram of the proposed approach.

The LP sizes in LPDC2020 were analyzed to select their anchor boxes, using the pyramid of anchors method of Faster R-CNN [19]. As shown in Figure 3, LP sizes span on a range of 10 to 670 pixels. Hence, in order to select anchor boxes of high intersection of union (IOU), six minimum LP sizes were used. These sizes were defined as 10×10 , 10×20 , 10×30 , 10×40 , 10×50 , and 30×14 pixels, with a pyramid level of 15 and anchor box pyramid scale of 1.3. As a result, 90 anchor boxes with a minimum of 0.625 and mean IOU of 0.85 were obtained. According to (6), the proposed last YOLOv2 layer had 540 filters.

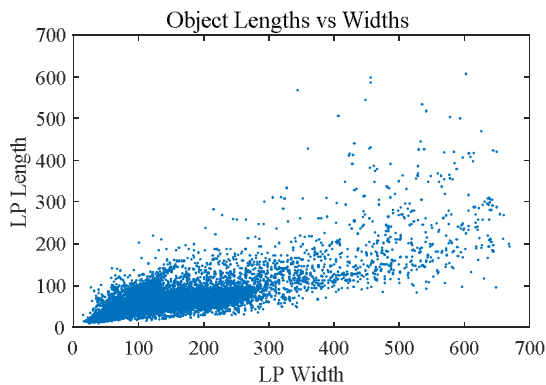


Fig. 3. LP sizes in LPDC2020 dataset.

B. License Plate Classification

A simple CNN was designed for LP classification, and its accuracy is compared to VGG [13]. The input image size is set to 224×224 pixels, being the same as the input size of VGG network for fair comparison. The classification CNN construction is shown in Table V. The proposed classifier design has a total number of $2635773 \approx 2.64\text{M}$ learnable parameters, being much less than the VGG learnable parameter amount of 138M. Both a Batch Normalization (BN) [48] and a ReLU non-linear activation layer [10] follow each

convolutional layer. BN normalizes the input batch mean and standard deviation, and then performs scaling and shifting based on learnable scale and shift parameters [48]. All convolution kernels have a size of 5×5 with stride 1 without padding. Hence, each convolutional layer results in a dimension shrinkage of 4 rows/columns. The dimension of the output feature map was computed according to (7):

$$W_{out} = \frac{W_{in} - W_k + 2P}{W_s} + 1 \quad (7)$$

where W_{out} is the output feature map width, W_{in} is the input feature map width, W_k is the kernel width, P is the padding, and W_s is the kernel stride in the horizontal direction. For input/output height relation, (7) can be applied using H instead of W . The input size is $224 \times 224 \times 3$. After 4 pooling and 8 convolutional layers, the output size is reduced to $6 \times 6 \times 128$. After that, Conv9 and Conv10 layers shrink the output to $1 \times 1 \times 512$ neurons. Using this design, the input image is convolved to a single neuron with 512 channels. Afterwards, these neurons are fitted to 11 classes in the FC layer by applying (2). This layer weights all input neurons and forwards them to the Softmax layer, which provides a score for the 11 classes and performs the classification task as described in (3). It is safe to note that the proposed design is a simple stacked CNN with a low number of learnable parameters.

TABLE V. PROPOSED CNN DESIGN FOR CLASSIFICATION

Layer	Filters & size	Output size	Learnable parameters
Input	-	$224 \times 224 \times 3$	-
Conv1	$5 \times 5 \times 32$	$220 \times 220 \times 32$	2496
Conv2	$5 \times 5 \times 32$	$216 \times 216 \times 32$	25696
Maxpooling	2×2	$108 \times 108 \times 32$	-
Conv3	$5 \times 5 \times 64$	$104 \times 104 \times 64$	51392
Conv4	$5 \times 5 \times 64$	$100 \times 100 \times 64$	102592
Maxpooling	2×2	$50 \times 50 \times 64$	-
Conv5	$5 \times 5 \times 96$	$46 \times 46 \times 96$	153888
Conv6	$5 \times 5 \times 96$	$42 \times 42 \times 96$	230688
Maxpooling	2×2	$21 \times 21 \times 96$	-
Conv7	$5 \times 5 \times 128$	$17 \times 17 \times 128$	307584
Conv8	$5 \times 5 \times 128$	$13 \times 13 \times 128$	409984
Maxpooling	2×2	$6 \times 6 \times 128$	-
Conv9	$5 \times 5 \times 256$	$2 \times 2 \times 256$	819968
Conv10	$2 \times 2 \times 512$	$1 \times 1 \times 512$	525824
Fully connected	11	$1 \times 1 \times 11$	5643
Softmax	-	$1 \times 1 \times 11$	-

C. Practical Aspects

The training process used Stochastic Gradient Descent with Momentum (SGDM) [46]. The SGDM training was carried out for 10 epochs, with an initial Learning Rate (LR) drop factor of 0.5 for every 2 epochs. The training set was shuffled for every epoch. In YOLOv2 training, the mini-batch size was only six images, due to memory constraints, and LR was set to 1×10^{-5} . Also, LP classification CNN mini-batch size was 120 images and LR was set to 2.5×10^{-2} . After the first results, model parameter tuning was applied to continue training, using ADAM adaptive learning rate optimization [46]. In ADAM, the batch size was doubled and LR was halved every 10 epochs, as long as the final error shows improvement.

V. RESULTS AND DISCUSSION

A MATLAB environment was used to evaluate the proposed approach. A GeForce 1060 6GB RAM GPU with computational capability of 6.1 was used for training and testing. The next subsections describe the evaluation criteria for both LPD and LPC.

A. LPD

The LP detection performance evaluation was performed using Precision (P), Recall (R) and Average Precision (AP) values. Any detected LP bounding box having an overlap greater than IOU=0.5 with the ground truth bounding box is considered as a correct detection. Precision is the percentage of the number of correctly detected LPs over the total number of detected LPs. R is the percentage of the number of correctly detected LPs over the total number of ground truth LPs. AP is the area under the precision recall curve. P and R are calculated by (8) and (9), where TP is true positive, FP is false positive, and FN is false negative detection.

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

Table VI shows the proposed detector’s AP performance compared to previous approaches presented in [32, 33, 37]. The proposed detector outperforms the previous approaches in terms of AP performance. It should be noted that in [32] only the accuracy for detected over all LPs in a private dataset is evaluated. Authors in [37] evaluated only the LPD precision, without presenting any AP values. It is evident that the proposed approach provides better detection score.

TABLE VI. MULTI-SET LPD COMPARISON RESULTS.

Approach	Detector	Score	Processing time (s)
[32]	Image processing	90.4% AP	0.25
[33]	VGG+ LSTM	98.07% AP	Not reported
[37]	Image processing + Alexnet + SVM	99.03% P	0.16
Proposed	ResNet40+ YOLOv2	99.57% AP	0.09

Those approaches were selected because they evaluated performance using images from all the countries of interest together in one dataset. Hence, these approaches can be considered as multi-national and multi-language LPD methods. Furthermore, some researches trained and tested detectors for different datasets separately, in order to evaluate the performance on each dataset. Table VII provides a comparison in terms of P, R and AP performance for these methods. In order to conduct a fair comparison, there was a need to train the proposed detector on every dataset separately. However, the proposed detector had higher R rate and AP on all datasets. This is partly due to the large number of different LPs used in LPDC2020 and to its superior architecture. It is noted that one and two-line LP layout classification was studied in [34] with classification results combined in the character recognition stage for multi-national Korean, Taiwanese, Chinese, and Latin LPs. Table VIII shows the proposed method’s AP results per country. It is apparent that the performance is similar, with slightly lower results for KSA LPs.

TABLE VII. SINGLE-SET LPD COMPARISON RESULTS

Approach	Caltech dataset	Zemris dataset	Medialab dataset	Various datasets
[34] tiny YOLOv3	P=100% R=100%	98% 99%	98.8% 99.7%	Taiwan: 100/100% Korea: 98.3/99%
[35] VGG+ Faster RCNN	AP = 98.03%	----	----	China: 98.33% Taiwan: 98.80%
[36] VGG +SSD	AP = 98.4%	97.83%	99.8%	----
[38] mask RCNN	P=98.9% R=98.6%	----	----	Taiwan: 99.1% China: 99.4% Tunisia: 97.9%
Proposed ResNet40+ YOLOv2	P=98.43% R=100% AP=99.96%	97.88% 100% 99.99%	98.4% 99.75% 99.74%	Snapshots: 98/100/99.99% UCSD: 99/100/99.93%

TABLE VIII. LPD RESULTS FOR LPDC2020 DATASET PER COUNTRY

Dataset	TR	EU	USA	UAE	KSA
Proposed ResNet40+ YOLOv2	99.48%	99.91%	99.95%	99.55%	98.67%

B. LPC

The proposed CNN for classifying the LP’s issuing country, language and layout was evaluated in terms of overall accuracy. Table IX shows the classification accuracy of the proposed CNN. The proposed CNN classification is only 0.38% less accurate than VGG16, which is regarded to be state-of-the-art, but with significantly fewer learnable parameters. The number of learnable parameters of the proposed approach is only 1.9% of the parameters used in VGG16. As a result, the proposed CNN is faster and less complex with a small penalty in classification accuracy.

TABLE IX. PROPOSED CNN LP CLASSIFICATION ACCURACY

CNN Architecture	Accuracy	Learnable parameters
VGG16	99.71%	136 M
Proposed CNN	99.33%	2.635 M

Table X shows the misclassification rates of the proposed approach. It is noted that Turkish and European Union’s LPs have a higher classification error, as they share the same LP style standard. In contrast, BR and UAE have a unique style, and USA LPs can include object shapes varying from standard LP characters, making it easy to classify them with a small error.

TABLE X. MISCLASSIFICATION OF LPDC2020-LPC DATASETS

Country	Language of characters	Layout	Number of instances	Misclassified LPs
BR	Latin	One-Line	3714	0
		Two-Line	900	0
UAE	Arabic	One-Line	500	0
		Two-Line	276	0
EU	Latin	One-Line	5296	14
		Two-Line	4350	0
KSA	Arabic	One-Line	290	0
		Two-Line	792	4
TR	Latin	One-Line	7771	18
		Two-Line	3560	0
USA	Latin	One-Line	1401	3

VI. CONCLUSION

Detecting country and language is important to build a global ALPR system, while correct layout classification is essential in order to read the detected characters in the right order. This paper focused on LP detection and classification of multi-national and multi-language LPs with different layouts from BR, USA, EU, TR, KSA and UAE, proposing a method that can detect LPs regardless of their country of origin, language or layout. Furthermore, a second classification stage was used to recognize LPs' issuing country, language and layout. In addition, a new multi-national, multi-language and multi-layout LP dataset was introduced in order to enable benchmarking and to close the gap in this field. The developed detection and classification approach was based on deep learning. The results were promising and the LP detection average precision was 99.57%, while the LP classification accuracy was 99.33%. The current study paves the way to designing a global ALPR system. In the future, an end-to-end training process could be developed to test the whole system as a unified ALPR model.

REFERENCES

- [1] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.
- [2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010, doi: 10.1109/TPAMI.2009.167.
- [3] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," presented at *International Conference on Image Processing*, Rochester, NY, USA, Sep. 22-25, 2002, doi: 10.1109/ICIP.2002.1038171.
- [4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.
- [5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, Jun. 2005, vol. 1, pp. 886–893, doi: 10.1109/CVPR.2005.177.
- [6] P. Matlani and M. Shrivastava, "An Efficient Algorithm Proposed For Smoke Detection in Video Using Hybrid Feature Selection Techniques," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 3939–3944, Apr. 2019.
- [7] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning*, Vol. 20, No. 3, pp. 273–297, Sep. 1995.
- [8] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *European Conference on Computational Learning Theory*, Mar. 1995, pp. 23–37, doi: 10.1007/3-540-59119-2_166.
- [9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010, doi: 10.1109/TPAMI.2009.167.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, Lake Tahoe, NV, USA, Dec. 2012, pp. 1097–1105.
- [11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, Jun. 2014, pp. 580–587.
- [12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," presented at the *International Conference on Learning Representations*, May 2015, arXiv: abs/1409.1556.
- [14] C. Szegedy *et al.*, "Going Deeper With Convolutions," presented at the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 7-12, 2015.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [16] D. Virmani, P. Girdhar, P. Jain, and P. Bamdev, "FDREnet: Face Detection and Recognition Pipeline," *Engineering, Technology & Applied Science Research*, vol. 9, no. 2, pp. 3933–3938, Apr. 2019.
- [17] L. Liu *et al.*, "Deep Learning for Generic Object Detection: A Survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, Feb. 2020, doi: 10.1007/s11263-019-01247-4.
- [18] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.
- [19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2980–2988, doi: 10.1109/ICCV.2017.322.
- [21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [22] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *European Conference on Computer Vision – ECCV 2016*, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.
- [23] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Jul. 2017, pp. 7263–7271.
- [24] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 6517–6525, doi: 10.1109/CVPR.2017.690.
- [25] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311–325, Feb. 2013, doi: 10.1109/TCSVT.2012.2203741.
- [26] J. Han, J. Yao, J. Zhao, J. Tu, and Y. Liu, "Multi-Oriented and Scale-Invariant License Plate Detection Based on Convolutional Neural Networks," *Sensors*, vol. 19, no. 5, p. 1175, Jan. 2019, doi: 10.3390/s19051175.
- [27] L. Xie, T. Ahmad, L. Jin, Y. Liu, and S. Zhang, "A New CNN-Based Method for Multi-Directional Car License Plate Detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 507–517, Feb. 2018, doi: 10.1109/ITITS.2017.2784093.
- [28] S. G. Kim, H. G. Jeon, and H. I. Koo, "Deep-learning-based license plate detection method using vehicle region extraction," *Electronics Letters*, vol. 53, no. 15, pp. 1034–1036, Jun. 2017, doi: 10.1049/el.2017.1373.
- [29] W. Min, X. Li, Q. Wang, Q. Zeng, and Y. Liao, "New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification," *IET Image Processing*, vol. 13, no. 7, pp. 1041–1049, Mar. 2019, doi: 10.1049/iet-ipr.2018.6449.
- [30] R. Laroca *et al.*, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018, pp. 1–10, doi: 10.1109/IJCNN.2018.8489629.
- [31] Z. Xu *et al.*, "Towards End-to-End License Plate Detection and Recognition: A Large Dataset and Baseline," in *European Conference*

- on *Computer Vision – ECCV 2018*, 2018, pp. 261–277, doi: 10.1007/978-3-030-01261-8_16.
- [32] M. R. Asif, Q. Chun, S. Hussain, M. S. Fareed, and S. Khan, “Multinational vehicle license plate detection in complex backgrounds,” *Journal of Visual Communication and Image Representation*, vol. 46, pp. 176–186, Jul. 2017, doi: 10.1016/j.jvcir.2017.03.020.
- [33] N. Dorbe, A. Jaundalders, R. Kadikis, and K. Nesenbergs, “FCN and LSTM Based Computer Vision System for Recognition of Vehicle Type, License Plate Number, and Registration Country,” *Automatic Control and Computer Sciences*, vol. 52, no. 2, pp. 146–154, Mar. 2018, doi: 10.3103/S0146411618020104.
- [34] C. Henry, S. Y. Ahn, and S.-W. Lee, “Multinational License Plate Recognition Using Generalized Character Sequence Detection,” *IEEE Access*, vol. 8, pp. 35185–35199, 2020, doi: 10.1109/ACCESS.2020.2974973.
- [35] H. Li, P. Wang, and C. Shen, “Toward End-to-End Car License Plate Detection and Recognition With Deep Neural Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1126–1136, Mar. 2019, doi: 10.1109/TITS.2018.2847291.
- [36] J. Yépez, R. D. Castro-Zunti, and S. B. Ko, “Deep learning-based embedded license plate localisation system,” *IET Intelligent Transport Systems*, vol. 13, no. 10, pp. 1569–1578, Jul. 2019, doi: 10.1049/iet-its.2019.0082.
- [37] M. R. Asif, C. Qi, T. Wang, M. S. Fareed, and S. A. Raza, “License plate detection for multi-national vehicles: An illumination invariant approach in multi-lane environment,” *Computers & Electrical Engineering*, vol. 78, pp. 132–147, Sep. 2019, doi: 10.1016/j.compeleceng.2019.07.012.
- [38] Z. Selmi, M. B. Halima, U. Pal, and M. A. Alimi, “DELP-DAR system for license plate detection and recognition,” *Pattern Recognition Letters*, vol. 129, pp. 213–223, Jan. 2020, doi: 10.1016/j.patrec.2019.11.007.
- [39] S. Park, H. Yoon, and S. Park, “Multi-Style License Plate Recognition System using K-Nearest Neighbors,” *KSI Transactions on Internet and Information Systems*, vol. 13, no. 5, pp. 2509–2528, May 2019, doi: 10.3837/tiis.2019.05.015.
- [40] *Caltech Computational Vision: Archive*, California Institute of Technology, 1999. [Online]. Available: <http://www.vision.caltech.edu/html-files/archive.html>.
- [41] K. Kraupner, “Using Multilayered Perceptron for Recognition of Alphanumeric Characters on License Plates”, Ph.D. dissertation, University of Zagreb, Croatia, 2003.
- [42] L. Dlagnekov, “Video-Based Car Surveillance: License Plate, Make, and Model Recognition”, M. S. thesis, University of California, San Diego, 2005.
- [43] O. Martinsky, “Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems”, B.Sc. thesis, Brno University of Technology, Croatia, 2007.
- [44] *Medialab LPR Database*, Multimedia Technology Laboratory, National Technical University of Athens, Greece. [Online]. Available: <http://www.medialab.ntua.gr/research/LPRdatabase.html>.
- [45] J. Spanhel, J. Sochor, R. Juranek, A. Herout, L. Marsik, and P. Zencik, “Holistic recognition of low quality license plates by CNN using track annotated data,” in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug. 2017, pp. 1–6, doi: 10.1109/AVSS.2017.8078501.
- [46] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [47] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed., New York, NY, USA: Springer, 2006.
- [48] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, Jun. 2015, pp. 448–456.