SRTTU

Journal of Electrical and Computer Engineering Innovations (JECEI)

Journal homepage: http://www.jecei.sru.ac.ir

JECEI

Research paper

DPRSMR: Deep learning-based Persian Road Surface Marking Recognition

S. H. Safavi[1,*], M. Sadeghi[2], M. Ebadpour[2]

[1] *Faculty of Advanced Technologies, University of Mohaghegh Ardabili, Namin, Iran.*

[2] *Faculty of Engineering, University of Mohaghegh Ardabili, Ardabil, Iran.*

Article Info

Article History:

Received 16 Dec 2022
Reviewed 28 Jan 2023
Revised 01 March 2023
Accepted 11 March 2023

Keywords:

Self-driving technology
Scene understanding
Persian Road Surface Marking Recognition
Deep Learning
Alex-Net
VGG

*Corresponding Author's Email Address: *h.safavi@uma.ac.ir*

Abstract

Background and Objectives: Persian Road Surface Markings (PRSMs) recognition is a prerequisite for future intelligent vehicles in Iran. First, the existence of Persian texts on the Road Surface Markings (RSMs) makes it challenging. Second, the RSM could appear on the road with different qualities, such as poor, fair, and excellent quality. Since the type of poor-quality RSM is variable from one province to another (i.e., varying road structure and scene complexity), it is a very essential and challenging task to recognize unforeseen poor-quality RSMs. Third, almost all existed datasets have imbalanced classes that affect the accuracy of the recognition problem.

Methods: To address the first challenge, the proposed Persian Road Surface Recognizer (PRSR) approach hierarchically separates the texts and symbols before recognition. To this end, the Symbol Text Separator Network (STS-Net) is proposed. Consequently, the proposed Text Recognizer Network (TR-Net) and Symbol Recognizer Network (SR-Net) respectively recognize the text and symbol. To investigate the second challenge, we introduce two different scenarios. Scenario A: Conventional random splitting training and testing data. Scenario B: Since the PRSM dataset include few images of different distance from each scene of RSM, it is highly probable that at least one of these images appear in the training set, making the recognition process easy. Since in any province of Iran, we may see a new type of poor quality RSM, which is unforeseen before (in training set), we design a realistic and challengeable scenario B in which the network is trained using excellent and fair quality RSMs and tested on poor quality ones. Besides, we propose to use the data augmentation technique to overcome the class imbalanced data challenge.

Results: The proposed approach achieves reliable performance (precision of 73.37% for scenario B) on the PRSM dataset. It significantly improves the recognition accuracy up to 15% in different scenarios.

Conclusion: Since the PRSMs include both Persian texts (with different styles) and symbols, prior to recognition process, separating the text and symbol by a proposed STS-Net could increase the recognition rate. Deploying new powerful networks and investigating new techniques to deal with class imbalanced data in the recognition problem of the PRSM dataset as well as data augmentation would be an interesting future work.

Introduction

The world is advancing toward a driverless future. However, self-driving car technology is still in its infancy stage, especially in Iran, and it cannot be deployed on urban traffic-filled roads yet. Although the fully autonomous car is not released yet in Iran, but the Advanced Driver Assistance Systems (ADAS) can help to achieve some levels of automation. Road intelligence in ADAS can be achieved by computer vision techniques. It enables self-driving vehicles to identify obstacles, traffic signs, and road markings, which avoid collisions and accidents. Road Surface Markings (RSMs) refer to the symbols or texts painted on the road surface with the aim of traffic guidance for drivers and pedestrians. Standard RSMs include lane indication arrows, crosswalks, caution words, speed limits, etc. These markings are as important as traffic signs at the side or on top of the roads, as they enable a better understanding of autonomous vehicles about their surrounding environments.

On the other hand, if one takes, for example, Google's self-driving car (developed for the U.S.) and tries to drive it in other (Europe or Asian) countries such as Iran, it will end up in an accident since there are lots of unforeseen scenes that are needed to be learned by the recognition algorithm. Road markings mainly include texts and symbols. Although the symbol markings in different countries are similar, text markings vary between countries and depend on the country's language. In Iran, road markings include some texts in the Persian language, and there is a need to address the recognition problem of them as well as symbol markings. To this end, and to facilitate the advent of self-driving car technology on the streets of all countries, this paper takes a small step toward it.

A. Related Datasets

Recently, various datasets have been released, which include RSMs. Most of them provide only RGB camera images such as ROMA [1], Road Marking Dataset [2], Reading the road dataset [3], Tsinghua Road Marking (TRoM) [4], BDD100K [5], and PRSM [6]. ROMA (ROad MArkings) image database [1] was collected in 2008. It comprises more than 100 original images of various road scenes. Moreover, the authors in [2] gathered a new dataset for road marking detection and classification. It consists of over 1200 labeled images of road markings with bounding boxes showing the location of the markings.

Furthermore, the authors in [3] created a benchmark ground truth class annotated dataset containing 2068 images spanning the city, residential, and motorway roads and over 13099 unique annotations. This dataset contains seven symbol-type categories and does not include texts. Tsinghua Road Markings (TRoM) dataset [4] is proposed for the recognition of road marking. This dataset has collected in Beijing municipality, China. It covers a diversity of traffic and weather conditions. In the current version of TRoM, the authors annotated 19 categories of road markings for recognition use. The BDD100K dataset [5] provides a large-scale, diverse driving video dataset with rich labels that reflects the challenges of street scene understanding. In addition to frames, it consists of GPS/IMU information to record the trajectories. Persian road surface marking (PRSM) dataset consists of 18 popular classes (6 text markings and 12 symbol markings) with the option of labeling different qualities such as excellent, fair, and poor. The whole dataset includes more than 68 thousand labeled images of RSMs. Moreover, the authors consider the rotation above 30 degrees of each road surface marking.

On the other hand, a few multimodal datasets use different sensors like the KITTI vision benchmark [7], Malaga Urban Dataset [8], and Oxford RobotCar dataset [9]-[10]. The KITTI Vision Benchmark Suite is the Karlsruhe Institute of Technology and Toyota technological institute (KITTI) dataset [7]. Also, they provide a benchmark for various autonomous vehicle applications. The KITTI suite includes images and other information for different tasks such as stereo, optical flow, visual odometry, 3D object detection, and 3D tracking. Malaga urban dataset [8] was gathered entirely in urban scenarios with a car equipped with several sensors, including one stereo camera and five laser scanners. Furthermore, the Oxford RobotCar dataset [9]-[10] contains over 100 repetitions of a consistent route through Oxford, UK, captured over a year. The dataset captures different combinations of weather, traffic, and pedestrians, along with longer-term changes such as construction and roadworks.

B. Related Works

Recently, vision-based techniques for road scene understanding such as lane detection [11]-[14], road surface marking detection and recognition [11], [15]-[17], road type classification [18], pedestrian action recognition [19], etc. have achieved great interest. Lane detection is an initial and important task to guide the car to be between lines. In [11], the authors propose a real-time integrated framework to perform lane-detection and tracking, road surface marking detection, and recognition on various datasets. In [13], the authors implement a real-time lane detection based on conventional edge features and Hough transform. Noise removal is applied using Gaussian filter and then the binary image is extracted using the Otsu algorithm. Then the Canny edge detection algorithm is followed by the Hough transform to perform lane detection. In [14], the authors propose an approach for lane detection called fully convolutional neural network (FCNN) that consists of nine convolutional layers. The runtime of implemented FCNN on Raspberry Pi reported as 3.75 seconds that is not suitable for real-time applications. Hence, the authors suggest accelerating the processing using FPGA or neural processing units. The authors in [15] investigate the effect of illumination on road surface marking recognition, and they present a real-time method that tries to find an illumination-free representation of road surfaces. The authors in [16] benefited from the YOLOv3 object detector [20] to detect 25 classes of road surface marking over 25 thousand images collected from Google Images.

C. Limitations

However, the vision-based techniques provide the details of the scene, but the reliability of them are affected by many challenges such as different weather condition (fog, haze, rain), different lighting condition (sunny, sunset, nighttime), sudden change of lighting (in and out of the tunnel), occlusion, etc., [21]. Besides cameras, other sensors like Radar and Light Detection and Ranging (LiDAR) could enhance the reliability of detection and recognition. Furthermore, the advent of Mobile Laser Scanning (MLS) technology assists the detection task [22]. Currently, not only the available multimodal datasets are not large enough to achieve higher accuracy, but also they do not have accurate ground-truth labels. Therefore, a weakly supervised learning system for real-time lane and road marking detection using multimodal data was proposed in [12].

D. Key Contributions

In this paper, the recognition of road surface marking on the PRSM dataset [6] presented. Fig. 1, shows the different classes of the PRSM dataset. Also, the first row of Table 1, shows the basic class distribution of the PRSM dataset. The contributions of the paper can be summarized as follows:

- We propose a network architecture for recognizing PRSMs inspired by VGG16 [15] and Alex-Net [16]. We call it Persian Road Surface Recognizer-Net (PRSR-Net).
- We investigate different challenging scenarios on the PRSM dataset. We design a realistic and interesting scenario to recognize unforeseen poor-quality road surface markings.
- To deal with the class imbalanced challenge, we propose to use the data augmentation technique.
- To achieve higher recognition accuracy, we propose to separate the text and symbol markings. The whole framework called Persian Road surface Recognizer (PRSR).

The rest of this paper is organized as follows: The second section introduces the recognition framework and describes the proposed approach. The third section describes the different scenarios considered in this paper. The fourth section gives simulation results. Finally, the fifth section concludes the paper.

Proposed Approach: Persian Road Surface Recognizer

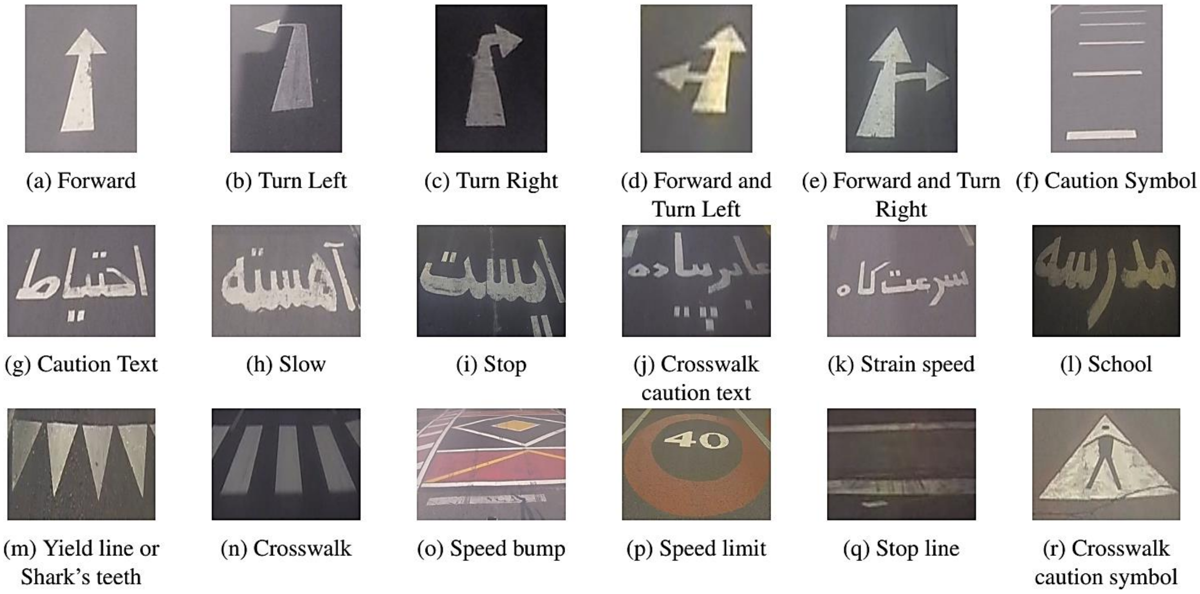
A. Network Architecture

Training a deep neural network often takes a substantial amount of time and needs powerful hardware. Regarding speed of the training procedure and overall accuracy, we propose to use the advantages of both Alex-Net [23] and VGG [24], respectively. Although the VGG and Alex-Net are not currently state-of-the-art methods, they are still used in the core of most recent neural networks [17], [25]-[30]. Therefore, these two architectures inspired us to use the advantages of each to get acceptable accuracy in Persian road surface marking recognition. Fig. 2 illustrates the proposed network architecture for road surface marking recognition.

The proposed architecture is composed of four essential stages. The max-pooling layers are used at the end of each stage. Similar to VGG16, we apply more than one convolutional layer before each max-pooling layer. Hence, the network captures more details. Accordingly, there are two convolutional layers to get enough features in each stage. Besides, the depth of the model was essential for its high performance.

Table 1: Class distribution of the used PRSM dataset with data augmentation

	Caution Symbol	Caution Text	Crosswalk	Crosswalk Caution Symbol	Crosswalk Caution Text	Forward	Forward and Turn Left	Forward and Turn Right	School	Slow	Speed Bump	Speed Limit	Stop	Stop line	Strain Speed	Turn Left	Turn Right	Yield Line
Basic	1824	4915	27893	1180	233	5747	963	2089	1085	3893	5368	193	1773	5615	938	203	625	3519
Augmented	3897	-	-	2888	1568	-	2988	3252	2472	-	-	1600	2721	-	2272	1793	1995	-
Used	3000	2844	3000	2888	1568	3000	2988	3000	2472	2213	3000	1500	2500	3000	2200	1600	1700	2291



(a) Forward (b) Turn Left (c) Turn Right (d) Forward and Turn Left (e) Forward and Turn Right (f) Caution Symbol

(g) Caution Text (h) Slow (i) Stop (j) Crosswalk caution text (k) Strain speed (l) School

(m) Yield line or Shark's teeth (n) Crosswalk (o) Speed bump (p) Speed limit (q) Stop line (r) Crosswalk caution symbol

Fig. 1: Classes of PRSM dataset [6].

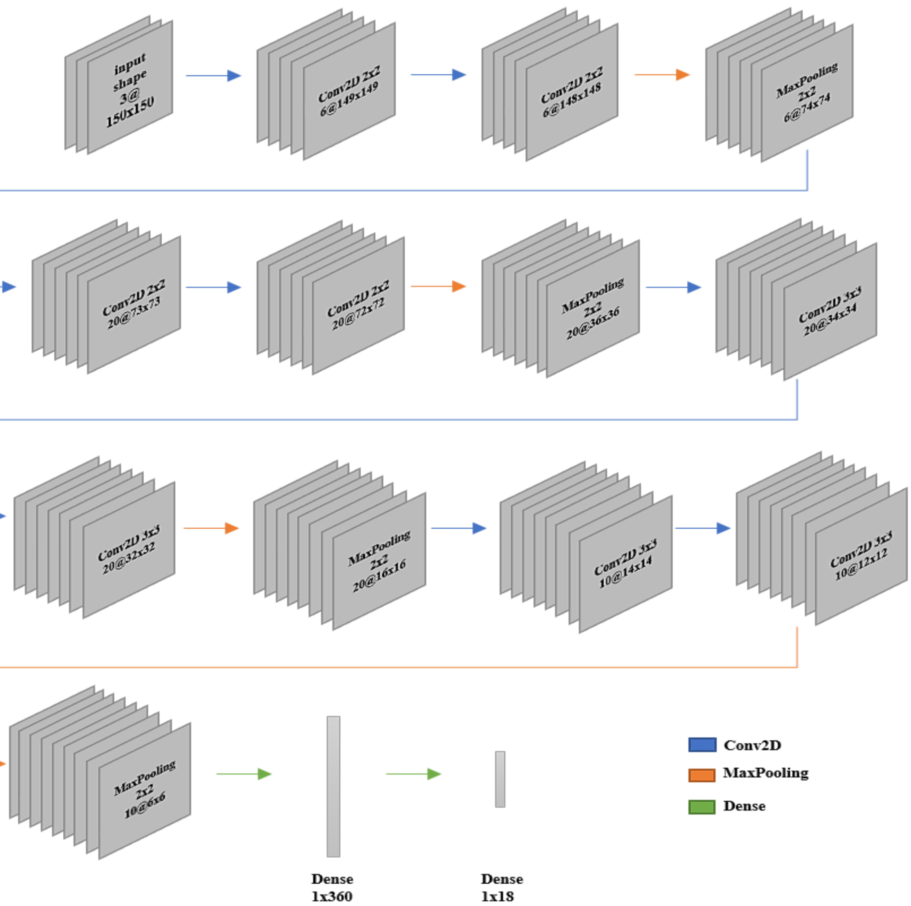


Fig. 2: Proposed network architecture for road surface marking recognition.

The Alex-Net architecture uses less depth number at the beginning and the end of the convolutional layers compared to the middle ones. Therefore, unlike VGG16, the depth number of each CNN layer is chosen based on the Alex-Net. Hence, as can be seen from Fig. 2, we use two CNN layers with a depth of six at the beginning and two CNN layers with a depth of ten at the end. The choice of these numbers is due to making the number of nodes in the next flattened layer less to make it faster; otherwise, we could use more than ten layers as depth. The depth of the middle four CNN layers is 20 to get more details. The numbers are not exactly like the numbers in

Alex-net or the VGG16. We just used the patterns of both architectures.

B. Different Scenarios

The road markings could be partially visible, occluded, or even faded. In addition, diverse lighting conditions (sunny, shadows, and bright) can affect RSMs recognition. The PRSM dataset includes road markings with three kinds of labeled quality: Poor, Fair, and Excellent. Fig. 3 and Fig. 4 show some road markings with different qualities in this dataset. It can be seen that it is hard to recognize the poor-quality markings as an unforeseen image in the test set. Besides, environmental factors (i.e., varying road structure and scene complexity) are variable from one province to another. Moreover, the available datasets are not large enough to capture comprehensive structural variations of RSMs. Therefore, it can lead to unforeseen scenes for self-driving cars. Hence, these challenges motivated us to design a different interesting scenario in which the model should recognize the unforeseen road markings with acceptable accuracy.

Fig. 3: “Forward” Symbol class with different qualities: First row includes “Excellent” quality markings. The second row includes “Fair” quality markings. The third row includes “Poor” quality markings. All of them are selected from the PRSM dataset [6].

The following subsections include two main scenarios regarding how we choose the train and test set.

Scenario A: We train our proposed model using images with different qualities in the first scenario. We randomly choose 70% of the images as training data and the rest for the test part in each class. This splitting scenario is conventional in different machine-learning tasks. The PRSM dataset contains various quality images (excellent, fair, and poor) in the learning phase. Therefore, the highest accuracy is expected to achieve in this scenario compared with scenarios that are not learned using images with all kinds of quality.

Fig. 4: “Caution Text” class with different qualities: First row includes “Excellent” quality markings. The second row includes “Fair” quality markings. The third row includes “Poor” quality markings. All of them are selected from the PRSM dataset [6].

Scenario B: In this scenario, we train our model with excellent and fair-quality images, and then the model is validated with poor-quality images. Compared to Scenario A, it is expected to have a low accuracy due to the difficulty of the scenario. Training a model with high-quality images and testing them on poor images that the model has never seen would have been less accurate. However, we tried several approaches in this scenario, containing practical and new techniques, aiming to get better results. In this regard, we propose the following methods:

1. Training the model with a balanced dataset (by eliminating extra images).
2. Data Augmentation.
3. Separating text and symbols.

Training the model with a balanced dataset: In real life, the RSMs do not appear equally. We expect to see the “CrossWalk” symbol more than, e.g., the “School” symbol. Table 1 shows the class distribution of road markings in the PRSM dataset. We observe that the “CrossWalk” class contains more than 27000 images, however, the “Caution Symbol” includes only about 1800 images. We believe these differences could lead to a model which learned more in the “CrossWalk” class rather than the “Caution Symbol” class. Therefore, first, we prefer to choose a limited number of images from each to create a more balanced dataset. Hence, we could create a model which learned equally over classes. It is also worth noting that [6] used only 10000 images of the “CrossWalk” class.

Data Augmentation: Balanced dataset is preferred as long as it provides enough information for recognition.

Although the class imbalanced problem can be avoided by elimination of extra images, the overall recognition accuracy would be decreased. Therefore, Data Augmentation technique could be used to produce additional new images for classes that have less than 2100 images. To augment these classes, we used an image generator in Keras. The following parameters were applied to create new images: Height shift of 0.1, width shift of 0.1, rotation range of 7, shear range of 0.15, zoom range of 0.1, and some brightness alters. After generating new images, we fit them into PRSR-Net. Table 1 represents the number of produced images for the mentioned classes. In addition, we still use limited numbers of images from some classes to prevent the model from over-learning in some classes.

Separating text and symbols: Similar to other datasets, the PRSM dataset includes two main categories: symbolic classes and text classes. For example, "Stopline" is symbolic, while "slow" is text-based. Fig. 1 illustrates an example of them. All images of the second row in this figure are from text classes. Keep in mind that text classes in the PRSM dataset are Persian. Inspecting the experimental results of previous techniques on the PRSM dataset, we observe some misclassified symbolic classes with text classes and vice versa. Therefore, we propose a novel hierarchical approach. If the recognition process is decomposed into two deep learning models, one for text recognition and one for symbol recognition, we expect an improvement in accuracy. In summary, we propose three deep models:

1. Symbol Text Separator (STS-Net)
2. Text Recognizer (TR-Net)
3. Symbol Recognizer (SR-Net)

Fig. 5 demonstrates the block diagram of the proposed approach. Because we need to train three different models with different outputs, it is clear that we will need different architectures, varying from the basic one represented in previous techniques. However, they are still taken from the basic model. STS-Net has only two outputs. So training this can be easier than the other ones. Because of output reductions, we eliminate two last Conv layers (20*20*3) from our basic model. Hence, it leads to less time and resources used for training.

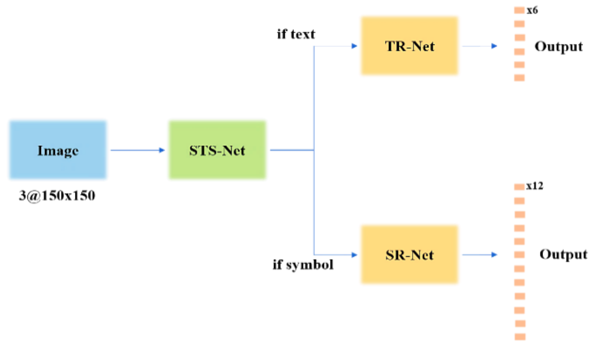


Fig. 5: Block Diagram of the proposed approach.

Fig. 6 represents the STS-Net architecture. About other models, based on the expected number of their outputs, we just altered the last dense layer. In TR-Net, the number of outputs should be six (the number of classes that are text classes) rather than eighteen in basic architecture. Similarly, in SR-Net we changed the last dense layer of the basic model and reduced it from 18 to 12 (the number of classes that are symbolic).

Results and Discussion

In this section, we present the results of the proposed methods. We also compare the results with earlier work [6]. For all of our four presented models, we apply the Adam optimizer. The back-propagation process uses the cross-entropy loss function. Training is done using different epochs, which we describe in the following sections. To make the procedure fast enough, we used the GPU version of Tensorflow, using Keras-GPU as API in python 3.5. GPU used in this article was NVIDIA GeForce 930MX with its 2G RAM. CPU was Intel corei7 and a DDR4 RAM with 8G of capacity.

In the following, different simulation scenarios are investigated and Table 2 summarizes them.

A. Scenario A

As it was explained before, this scenario was more convenient for the model to learn. Although all images were used, random images choosing and learning from poor-quality images helped us get better output, classifying and predicting unforeseen poor-quality images. In 35 epochs, we could gain almost 99.15% learning accuracy and 97% validation accuracy.

Table 2: Different Scenarios defined in this paper

	Train Set	Test Set	Network Architecture	Num. of epochs	Data augmentation	Overall Accuracy
Scenario A	Poor, Fair, Excellent	Poor, Fair, Excellent	Fig. 2	35	No	99.15
Scenario B	Fair, Excellent	Poor	Fig. 5, Fig. 6	35	Yes	65
Scenario C	Fair, Excellent	Poor	Fig. 5, Fig. 6	35	No	73.37

Fig. 7 represents the accuracy changes in different epochs and loss function variation. The confusion matrix is also shown in Fig. 8. As it shows, the most challenging classes for the model were Turn Left and Crosswalk Caution Text. PRSR-Net misclassified almost 20% of their images. Training more epochs or using data augmentation on the most misclassified classes would be a better solution to improve the model.

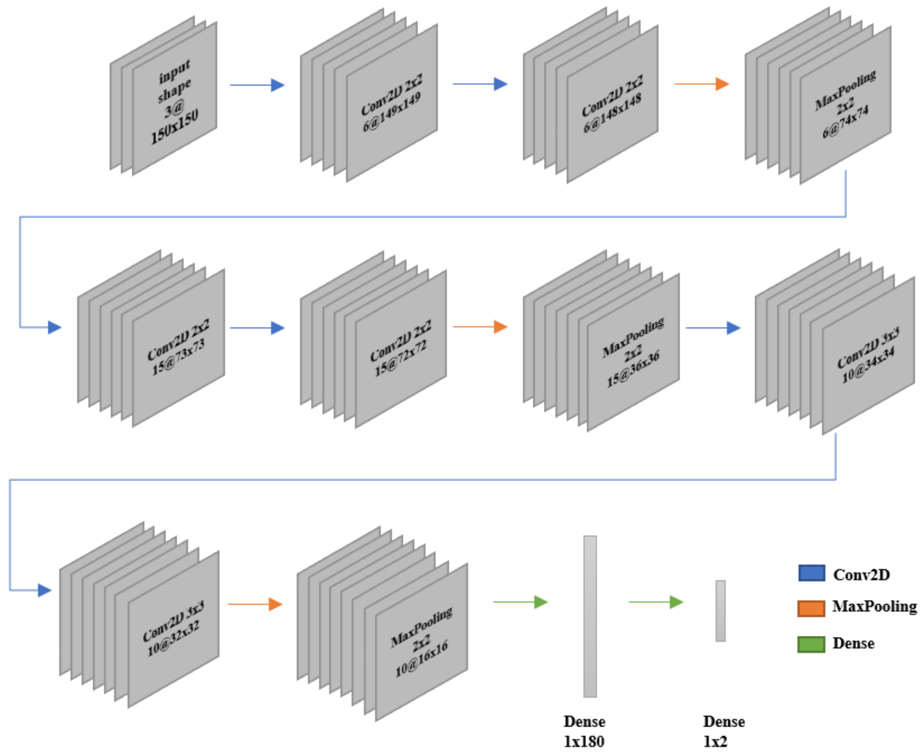


Fig. 6: Proposed Network Architecture for Symbol Text Separator (STS-Net).

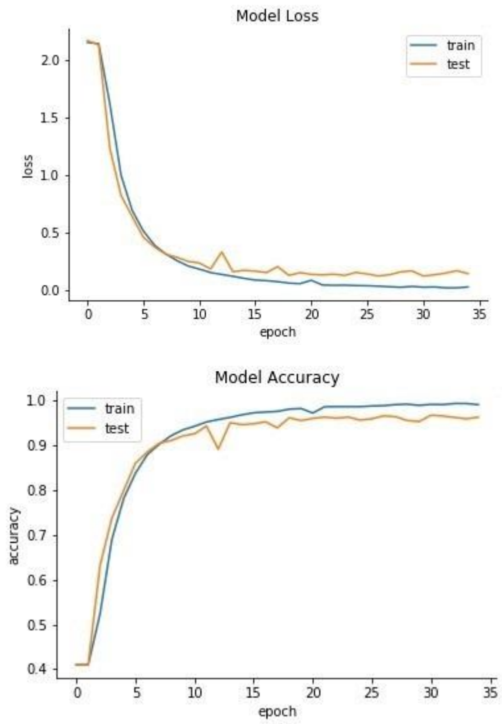


Fig. 7: Model Loss and Accuracy for Scenario A.

Fig. 9 represents some of the misclassified images. Obviously, they are hardly classified even by a human. Compared to the previous works, there are some advantages here. Since we see the "Crosswalk" class more in real life, rather than, e.g., the "School" class, it is supposed to have a large number of images in the dataset. Our model's validation accuracy for this class was about 98%, while earlier work classified only 88% of this class correctly. "Speed Limit" is another example. The presented model reached 97% accuracy in this class, while previous work did only 91%. For those with less accuracy, we should take this point into account how deep neural networks work. As we all know, deep learning models need more data than other classifiers. Having said this, a lack of enough data could lead our model to misclassify some classes more than others. As we suggested, data augmentation on these classes can solve the problem and raise accuracy.

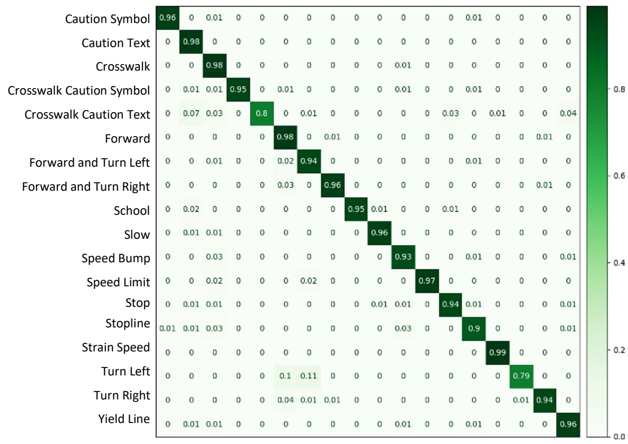


Fig. 8: Confusion matrix for the scenario A.

B. Scenario B

Training a balanced dataset: As we mentioned earlier, we created a balanced training dataset, and we restricted the number of images in classes that the proportion of them was unacceptable, intending to reduce consumed time and fewer computations required.

Fig. 9: misclassified examples.

Pointing out again, we trained PRSR-Net (our basic model) with excellent and fair-quality images and tested them with poor-quality images. With these in mind, we got almost 64% validation accuracy in 35 epochs. Although it is still not the desirable accuracy, it is 8% more than the best result of previous work [6], which was 58%. In addition, note that we achieved this accuracy using less number of images compared to [6]. Moreover, we used data augmentation to add more images to classes that hadn't as many as other classes. The general points of this part are described earlier.

During 35 epochs, we got almost 65% accuracy. This accuracy is noteworthy due to the less-used number of images compared to the last technique. In other words, using all images, we would have achieved a more accurate model. However, it increased our accuracy by about 1%. Fig. 10 shows the confusion matrix of this method.

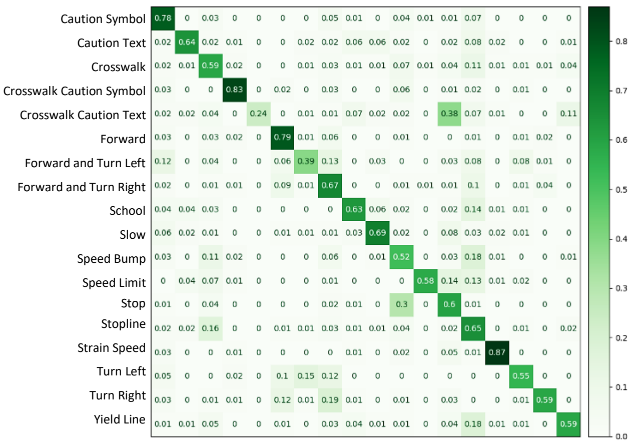


Fig. 10: Confusion matrix for the proposed method: scenario B.

C. *Scenario C*

Separating text and symbols: In this approach, we trained three different models. Starting from STS-Net, we trained it in 20 epochs, getting almost 99% validation accuracy. SR-Net and TR-Net were both trained in 35 epochs, achieving validation accuracy of almost 73.5% and 75.2%, respectively. Overall accuracy was about 74%, being higher than every presented technique in scenario B and also 16% more accurate than [6]. The block diagram of this approach is already shown in the earlier section. Fig. 11 shows the confusion matrix of this method.

"Stop Line" and "Forward and Turn Right" with 21% and 28% accuracies, respectively, were misclassified the most and had not satisfying results. Regardless of these classes, the others gained better outputs. E.g., Yield Line, Crosswalk, and Forward are classes that have almost perfect results. Take "Turn Left" as an example. The validation accuracy of this class in [6] is about 28%, while the proposed method could achieve a significant performance which is 80% validation accuracy.

Finally, Table 3, summarizes the number of parameters used in the proposed PRSR Network. Moreover, each frame is processed at about 15 msec.

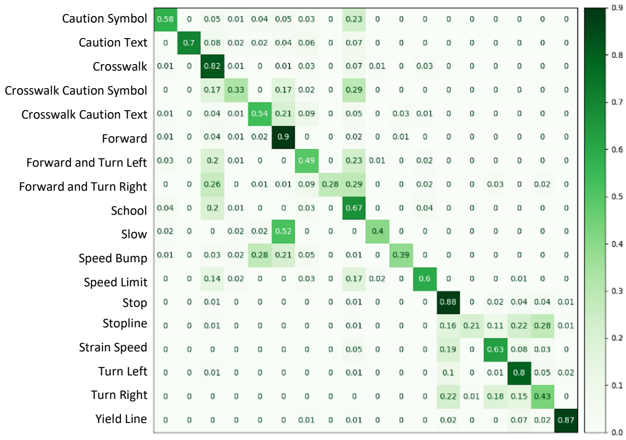


Fig. 11: Confusion matrix for the proposed method: scenario C.

Table 3: Number of parameters in the PRSR Network of Fig. 2

Layer	Output Shape	Number of Parameters
Conv_2D	(149,149,6)	78
Conv_2D	(148,148,6)	150
Max_Pooling_2D	(74,74,6)	0
Conv_2D	(73,73,20)	500
Conv_2D	(72,72,20)	1620
Max_Pooling_2D	(36,36,20)	0
Conv_2D	(34,34,20)	3620
Conv_2D	(32,32,20)	3620
Max_Pooling_2D	(16,16,20)	0
Conv_2D	(14,14,10)	1810
Conv_2D	(12,12,10)	910
Max_Pooling_2D	(6,6,10)	0
Flatten	360	0
Dense	360	129960
Dense	18	6498
Total Trainable Parameters		148766

Conclusion

In this paper, inspired by Alex-Net and VGG, a deep-learning approach was developed to overcome the recognition challenge of the PRSM dataset. Since the PRSMs include both Persian texts (with different styles) and symbols, prior to recognition process, separating the text and symbol by a proposed STS-Net could increase the recognition rate. Moreover, we design a realistic and challengeable scenario in which the network is trained using excellent and fair quality RSMs and tested on poor quality ones. The proposed approach achieves reliable performance (precision of 73.37%) on this dataset. It significantly improves the recognition accuracy by up to 15% in different scenarios compared to [6]. Deploying new powerful networks and investigating new techniques to deal with class imbalanced data in the recognition problem of the PRSM dataset as well as data augmentation would be an interesting future work.

Author Contributions

S. H. Safavi raise the idea. All authors equally contribute in designing the experiments. M. Sadeghi and M. Ebadpour run the simulations. All authors involve in data analysis and interpreting the results and S. H. Safavi wrote the initial draft of the manuscript and revised the manuscript. This paper was a research collaboration of M. Sadeghi and M. Ebadpour as overplus university activities of them during their B.Sc.

Acknowledgment

The authors would like to thank both the Editor and anonymous Reviewers for their insightful comments and suggestion to improve the quality of the paper.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

DPRSMR	Deep learning-based Persian Road Surface Marking Recognition
CNN	Convolutional Neural Networks
PRSM	Persian road surface marking
PRSR	Persian Road Surface Recognizer
STS-Net	Symbol Text Separator Network
TR-Net	Text Recognizer Network
SR-Net	Symbol Recognizer Network

References

[1] T. Veit, J. P. Tarel, P. Nicolle, and P. Charbonnier, "Evaluation of road marking feature extraction," in Proc. 11th International IEEE Conference on Intelligent Transportation Systems: 174–181, 2008.

[2] T. Wu, A. Ranganathan, "A practical system for road marking detection and recognition," in Proc. IEEE Intelligent Vehicles Symposium: 25–30, 2012.

[3] B. Mathibela, P. Newman, I. Posner, "Reading the road: Road marking classification and interpretation," IEEE Trans. Intell. Transp. Syst., 16(4): 2072–2081, 2015.

[4] X. Liu, Z. Deng, H. Lu, L. Cao, "Benchmark for road marking detection: Dataset specification and performance baseline," in Proc. IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017.

[5] F. Yu *et al.*, "BDD100K: A diverse driving video database with scalable annotation tooling," arXiv preprint arXiv:1805.04687, 2018.

[6] S. H. Safavi *et al.*, "Image dataset for persian road surface markings," in Proc. IEEE 10th Iranian Conference on Machine Vision and Image Processing (MVIP): 258–264, 2017.

[7] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, "Vision meets Robotics: The KITTI Dataset," Int. J. Rob. Res., 32(11): 1231–1237, 2013.

[8] J. L. Blanco, F. A. Moreno, J. Gonzalez-Jimenez, "The málaga urban dataset: High-rate stereo and lidars in a realistic urban scenario," Int. J. Rob. Res., 33(2): 207–214, 2014.

[9] W. Maddern, G. Pascoe, C. Linegar, P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," Int. J. Rob. Res. (IJRR), 36(1): 3–15, 2017.

[10] W. Maddern, G. Pascoe, M. Gadd, D. Barnes, B. Yeomans, P. Newman, "Real-time kinematic ground truth for the oxford robotcar dataset", in arXiv preprint arXiv: 2002.10152, 2020.

[11] A. Gupta, A. Choudhary, "A framework for camera-based real-time lane and road surface marking detection and recognition," IEEE Trans. Intell. Veh., 3(4): 476–485, 2018.

[12] T. Bruls, W. Maddern, A. A. Morye, P. Newman, "Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data," in Proc. IEEE International Conference on Robotics and Automation (ICRA): 1863-1870, 2018.

[13] A. Fallah, A. Soliemani, H. Khosravi, "Real-time lane detection based on image edge feature and hough transform," J. Electr. Comput. Eng. Innovations (JECEI), 9(2): 193-202, 2021.

[14] N. S. Danishevskiy, I. A. Ershov, D. O. Budanov, "Computer vision system for road surface marking recognition," in Proc. IEEE International Conference on Electrical Engineering and Photonics (EExPolytech): 130-133, 2022.

[15] B. A. Maxwell *et al.*, "Real-time physics-based removal of shadows and shading from road surfaces", in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW): 1277–1285, 2019.

[16] E. S. Dawam, X. Feng, "Smart city lane detection for autonomous vehicle" in Proc. IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech): 334-338, 2020.

[17] M. R. Bachute, J. M. Subhedar, "Autonomous driving architectures: insights of machine learning and deep learning algorithms," Elsevier, Machine Learning with Applications, 6(100164): 1-25 2021.

[18] D. K. Dewangan, S. P. Sahu, "RCNet: road classification convolutional neural networks for intelligent vehicle system," Intell. Serv. Rob., 14(2): 199-214, 2021.

[19] R. D. Brehar, M. P. Muresan, T. Mariţa, C. C. Vancea, M. Negru, S. Nedevschi, “Pedestrian street-cross action recognition in monocular far infrared sequences,” IEEE Access, (9): 74302-74324, 2021.

[20] J. Redmon, A. Farhadi, “Yolov3: An incremental improvement”, arXiv preprint arXiv:1804.02767, 2018.

[21] Z. Feng, M. Li, M. Stolz, M. Kunert, W. Wiesbeck, “Lane detection with a high-resolution automotive radar by introducing a new type of road marking,” IEEE Trans. Intell. Trans. Syst., 20(7): 2430-2447, 2018.

[22] S. Chen, Z. Zhang, H. Ma, L. Zhang, R. Zhong, “A content-adaptive hierarchical deep learning model for detecting arbitrary-oriented road surface elements using MLS point clouds,” IEEE Trans. Geosc. Remote Sens., 61(5700516): 1-16, 2023.

[23] A. Krizhevsky, I. Sutskever, G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” Advances in neural information processing systems (NIPS 2012): 1097–1105, 2012.

[24] K. Simonyan. A. Zisserman, “Very deep convolutional networks for largescale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[25] R. Girshick, “Fast r-cnn,” in Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR): 1440-1448, 2015.

[26] S. Ren, K. He, R. Girshick, J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” Advances in neural information processing systems (NIPS), (28):1-9, 2015.

[27] S. Ren, K. He, R. Girshick, J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI), 39(6):1137-1149, 2017.

[28] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, J. Sun, “RepVGG: Making VGG-style ConvNets great again,” in Proc. IEEE/CVF conference on computer vision and pattern recognition (CVPR): 13733-13742, 2021.

[29] L. Chen *et. al.,* “Deep integration: A multi-label architecture for road scene recognition,” IEEE Trans. Image Proc., 28(10): 4883–4898, 2019.

[30] K. Lis *et. al.*, “Detecting the unexpected via image resynthesis” in Proc. IEEE/CVF International Conference on Computer Vision (ICCV): 2152-2161, 2019.

Biographies

Seyed Hamid Safavi is an Assistant Professor at University of Mohaghegh Ardabili (UMA). He received his PhD in electrical engineering from Shahid Beheshti University (SBU), Tehran, Iran in 2017. He also received his B.Sc. and M.Sc. degrees both in electrical engineering from K. N. Toosi University of Technology (KNTU), Tehran, Iran in 2010 and 2012, respectively. He was also a visiting fellow at Singapore University of Technology and Design (SUTD) from 2015 to 2016. He received the best Ph.D. thesis award from IEEE SBU Student Branch. He also received the IEEE travel grant to participate at the 1st IEEE ComSoc Summer School in 2015. His research interests include Signal and Image Processing, Machine Learning, Compressive Sensing, and Convex Optimization.

- Email: h.safavi@uma.ac.ir
- ORCID: 0000-0001-7833-7381
- Web of Science Researcher ID: B-7187-2014
- Scopus Author ID: 53864204500
- Homepage: https://uma.ac.ir/cv.php?slc_lang=fa&sid=1&mod=scv&cv=2513

Mahyar Sadeghi received his B.Sc. from the University of Mohaghegh Ardabili (UMA) in computer engineering. He is currently a M.Sc. student in computer science -Data Engineering and Artificial Intelligence in university of Genova, Italy. He is also a research fellow in this university, working on “AI agent-based model oriented to traffic control using CAVs”. His research interests include Reinforcement Learning, Machine Learning, Multi Agent Systems and brain inspired Artificial Intelligence.

- Email: mahyarsadeghi.ga@gmail.com
- ORCID: 0000-0003-0388-8669
- Web of Science Researcher ID: HDM-4187-2022
- Scopus Author ID: NA
- Homepage: https://sites.google.com/view/mahyarsadeghi-ga/home?pli=1

Mohsen Ebadpour received his B.Sc. from the University of Mohaghegh Ardabili (UMA) in computer engineering. He is currently a M.Sc. student in artificial intelligence at Amirkabir University of Technology (AUT) and a research fellow at the Image Processing and Pattern Recognition (IPPR) laboratory. Currently, he is working on "Few-Shot Learning in Multi-Label Classification." His research interests include Image Processing, Computer Vision, Deep Learning and Generative models, Graph Theory, and Complex Networks.

- Email: m.ebadpour@aut.ac.ir
- ORCID: 0000-0003-2745-7329
- Web of Science Researcher ID: HDN-2945-2022
- Scopus Author ID: NA
- Homepage: https://ce.aut.ac.ir/~ebadpour/

How to cite this paper:
S. H. Safavi, M. Sadeghi, M. Ebadpour, “DPRSMR: Deep learning-based persian road surface marking recognition,” J. Electr. Comput. Eng. Innovations, 11(2): 409-418, 2023.

DOI: 10.22061/jecei.2023.9496.627

URL: https://jecei.sru.ac.ir/article_1851.html