

Automatic Vehicle License Plate Recognition Using Optimal Deep Learning Model

Thavavel Vaiyapuri¹, Sachi Nandan Mohanty², M. Sivaram³, Irina V. Pustokhina⁴,
Denis A. Pustokhin⁵ and K. Shankar^{6,*}

¹College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia

²Department of Computer Science & Engineering, IcfaiTech, ICFAI Foundation for Higher Education, Hyderabad, India

³Assistant professor Research, Research Center, Lebanese French University, Erbil, 44001, Iraq

⁴Department of Entrepreneurship and Logistics, Plekhanov Russian University of Economics, Moscow, 117997, Russia

⁵Department of Logistics, State University of Management, Moscow, 109542, Russia

⁶Department of Computer Applications, Alagappa University, Karaikudi, India

*Corresponding Author: K. Shankar. Email: drkshankar@ieee.org

Received: 27 October 2020; Accepted: 23 November 2020

Abstract: The latest advancements in highway research domain and increase in the number of vehicles everyday led to wider exposure and attention towards the development of efficient Intelligent Transportation System (ITS). One of the popular research areas i.e., Vehicle License Plate Recognition (VLPR) aims at determining the characters that exist in the license plate of the vehicles. The VLPR process is a difficult one due to the differences in viewpoint, shapes, colors, patterns, and non-uniform illumination at the time of capturing images. The current study develops a robust Deep Learning (DL)-based VLPR model using Squirrel Search Algorithm (SSA)-based Convolutional Neural Network (CNN), called the SSA-CNN model. The presented technique has a total of four major processes namely preprocessing, License Plate (LP) localization and detection, character segmentation, and recognition. Hough Transform (HT) is applied as a feature extractor and SSA-CNN algorithm is applied for character recognition in LP. The SSA-CNN method effectively recognizes the characters that exist in the segmented image by optimal tuning of CNN parameters. The HT-SSA-CNN model was experimentally validated using the Stanford Car, FZU Car, and HumAIn 2019 Challenge datasets. The experimentation outcome verified that the presented method was better under several aspects. The projected HT-SSA-CNN model implied the best performance with optimal overall accuracy of 0.983%.

Keywords: Deep learning; license plate recognition; intelligent transportation; segmentation

1 Introduction

Vehicle License Plate Recognition (VLPR) has been a major computer vision issue in recent decades. In this scenario, the prevalent systems of cameras are placed at road junctions to find



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

the vehicles' routing using urban platform [1]. The feasibility of leveraging the existing networks with transparent light cameras for VLPR is highly attractive due to its cost-efficiency, as ad-hoc architectures such as infrared illuminators and cameras are deployed. However, VLPR contains natural visible light which makes it a tedious operation, because it is deployed with a non-optimal camera that offers extensive difference in terms of plate and scale. Likewise, the surrounding illumination also makes VLPR a challenging task, because of the massive differences in brightness levels, perspectives, and minimum dimensions. Further, it is essential to monitor large scale vehicles with the help of videos clips captured during vehicle observation, then provide it to the administrator who applies essential bandwidth and fixes the data center. This results in increasing the capability of managing peak-time requests while the VLPR tasks can be operated locally on the cameras.

Deep learning and CNNs are considered to be the cornerstones of models which can resolve computer vision problems in an efficient manner [2]. Some of the issues faced are object prediction, character analysis and ImageNET Challenge with higher margins in comparison to classical image processing models. Moreover, CNN is a feed-forward and multi-layer NN, divided into feature extraction phase and inference stage. In the beginning, the feature extraction phase is comprised of numerous convolutional layers where it encompasses several learnable filters. These filters are enabled by the prediction of certain features present in the input data. The outcome of the feature extraction stage is determined with the help of Fully-Connected (FC) layers; the number of layers and the accessible parameters present in every layer rely upon some particular layers. Consequently, the final layer of the network yields the required results to which a class of object belongs (classification issues) or the location of the object in an image (regression issues).

In spite of the fact that CNNs have resulted in a remarkable performance in image processing operations, there are few complications associated with it when designing a practical VLPR system. Initially, it is vital to train the CNNs by testing an adequate number of images, along with a distribution of variables that include intra-class variance, i.e., the implication of real-world constraints. The sourcing and annotation of a massive training dataset with limited variable distribution, are considered overwhelming tasks even under the application of data augmentation methods. Thereafter, the current CNN structures would have attained a state-of-the-art function which would depend upon deep topologies with enormous amounts of readable parameters. The demand for moving VLPR computing, through smart cameras, calls for complex network structure that is applicable for smart cameras with minimum storage and processing resources.

In the last few decades, various methods were established to overcome the issues involved in VLPR, according to the type of structures [3–6]. Yu et al. [3] concentrated on the issue of predicting car plates under widely-varying brightness, background, and perspective conditions. This issue can be resolved when using a model that depends upon Wavelet Transform (WT) as well as empirical mode degradation, which exhibit optimal accuracy at plate deployment though does not provide a VLPR pipeline.

Zhou et al. [4] reported the issues in plate as well as character segmentation concerning conventional image processing models with a parameter-based method. Hence, these methodologies are infeasible to solve the problem of character analysis, a significant operation in VLPR system. Giannoukos et al. [5] presented a rapid technology to reduce the time of plate detection in high-resolution images. The technique presented was a context scanning model which can be applied to compute a Quarter Video Graphics Array (QVGA) image in a CPU. Regardless of the above, intelligent GPU-based CNNs are capable of processing massive images simultaneously.

Ghaili et al. [6] proposed an effective framework for LP under the identification of vertical edges by applying contrast, removal of unknown lines, and deployment of the plate in a binarized image. Though these methods exhibit minimum difficulty at plate identification, they could not offer a better solution in the case of a VLPR system. Hsu et al. [7] presented a complete 3-stage VLPR structure that was developed over edge clustering for plate prediction, and it is highly reliable for character segmentation. VLPR is known to have different applications like management, road patrol, among others. It showcases that an ad-hoc solution is responsible for the applied range of variables which performs better than the alternate solutions, irrespective of the specific application parameters. The major limitation in this method is that the best solution is required for all domains.

Li et al. [8] integrated various networks for handling VLPR problems from different perspectives. Firstly, a CNN was applied on already-trained datasets to predict the characters involved in the input image. The members of the images were categorized into plate and non-plate through a CNN trained using an Application Oriented License Plate (AOLP) dataset over Cross-Validation (CV) in order to eliminate false positives. As a result, a Long Short-Term Memory (LSTM) that was trained on identical character set, was applied to label the characters as a textual sequence instead of applying the character segmentation. In spite of applying three different networks, this model depicted similar results on the dataset applied.

Yuan et al. [9] proposed a plate prediction structure. In the first stage, a line density filter placed the candidate LP sites. Then, the LP classification method removed the false positive regions based on the color salience. The presented technique accomplished the best recall and precision results with limited resilience. Jiao et al. [10] reported the problem of placing plates with diverse appearances using a tunable algorithmic method. However, the synthetically-trained LPR system is applicable for processing several plate factors by template modification that produces training images.

Gou et al. [11] employed LPR technology based on Extremal Regions as well as Restricted Boltzmann Machines (RBM). Initially, a common examination of LP was carried out under the applications of edge detection as well as image filtering. Characters sites were filtered with the help of Extremal Regions which were then applied in the refining plate region. At last, the characters were analyzed by applying a hybrid discriminative RBM which was trained on character samples obtained from rotation and noise-augmented actual images.

Bulan et al. [12] presented a model to exploit weak and sparse classification methods and a strong CNN to isolate the readable LP. In the character analysis, the model eliminated the segmentation phase with the application of a sweeping SVM classifier and a hidden Markov approach to infer the positions. The character classifier was trained using a real sample that has been labelled by existing classifier. However, during the performance validation, a performance loss was observed when the network underwent training on synthetic data. Meyer et al. [13] identified the problem of making the best synthetic training data for DL in disparity as well as in optical flow determination. Finally, the simulation outcome was supported with the help of the attained results in the specific case of LPR. In Pustokhina et al. [14], a DL-centric VLPR approach by optimal k-means (OKM) clustering relied segmentation, as well as a CNN relied technique were introduced.

Some other DL models are available in the literatures [15–23]. In the literature [15], an improved VGG model was presented to recognize and classify the traffic signs. In the study [16], a new model called WI-Multi was proposed to identify the human activities using WiFi devices.

An event-driven plan recognition model using intuitionistic fuzzy theory was devised in the literature [17]. The authors [18] developed a CN-ELM model to recognize the Electrocardiograms. Besides, a deep local search method using internal spanning tree was devised for parameterized and approximation algorithms [19]. Another lightweight DL model to classify the traffic signs was developed in the literature [20]. A new grammatical model was also presented in the study conducted earlier [21]. An improved model for inspecting deep packets with the help of regular expression was proposed in research conducted earlier [22]. Another improved model to inspect deep packets in data stream detection was introduced in the study [23].

Though various LPR models are available in the literature, there still exists a need to develop a proficient VLPR model for the detection and analysis of the characters in an LP effectively. It is also required to consider the real time constraints in designing a VLPR model. From this view, this paper introduces a robust DL-based VLPR model using an SSA-based CNN, termed as an SSA-CNN model. The proposed model has a total of four major processes namely, preprocessing, LP localization and detection, HT-based character segmentation and SSA-CNN-based recognition. The SSA algorithm was applied in a CNN model to choose the hyper parameters properly and to effectually recognize the characters that exist in the segmented image. The HT-SSA-CNN model was experimentally validated using a benchmark dataset, and the simulation outcome confirmed that the presented model yielded better results.

The remaining portions of the study are formulated as follows. Section 2 details the proposed HT-SSA-CNN model with adequate explanations. Section 3 performs the experimental validation and Section 4 draws the conclusion for the study.

2 The Proposed HT-SSA-CNN Model

The entire working process, involved in the proposed HT-SSA-CNN model, is depicted in Fig. 1. As depicted in the figure, the input image is preprocessed to make it compatible for further processing. Then, the preprocessed image is fed into an LP localization process in order to detect and crop the LP effectively. Thereafter, HT is applied to segment the characters that exist in the LP. At last, an SSA-CNN model is applied to examine the characters in the classified image.

2.1 Preprocessing

During pre-processing, the RGB car image undergoes downscaling to 50% of its actual scale in order to confine the processing duration. Also, a reduction and a reforming of images is applied to minimize the candidate sites. The input image is composed of RGB channels, where each channel is restricted to within (0–255), while the gray scale image contains a single channel; thus an RGB image is converted into a gray scale template. Additionally, the contrast of the images is enhanced to compute the LP detection process. In line with this, previous models are used with English LP. So, there is no requirement for image cropping since the images are taken from nearby-placed vehicles.

2.2 License Plate Localization

In the presented model, the LP is filtered using a set of tasks, like (i) using Median Filter (MF) with (3×3) for image development and noise elimination. (ii) Exploiting sobel edge detector to detect proper edges. (iii) Employment of morphological tasks to isolate the plate from background. Here, dilation is utilized to enhance the boundary dimension so as to get rid of line issues. The dilation process creates bigger objects since every background pixel is transmitted to an object pixel. Erosion is applied to allocate the candidate plate regions under the application

of Squared Structuring component. Finally, the appropriate LP is placed. In this domain, it is applied with two fundamental checkers to ensure the plate region accurately, remove the unwanted sites. Few more steps are listed in the following section [24].

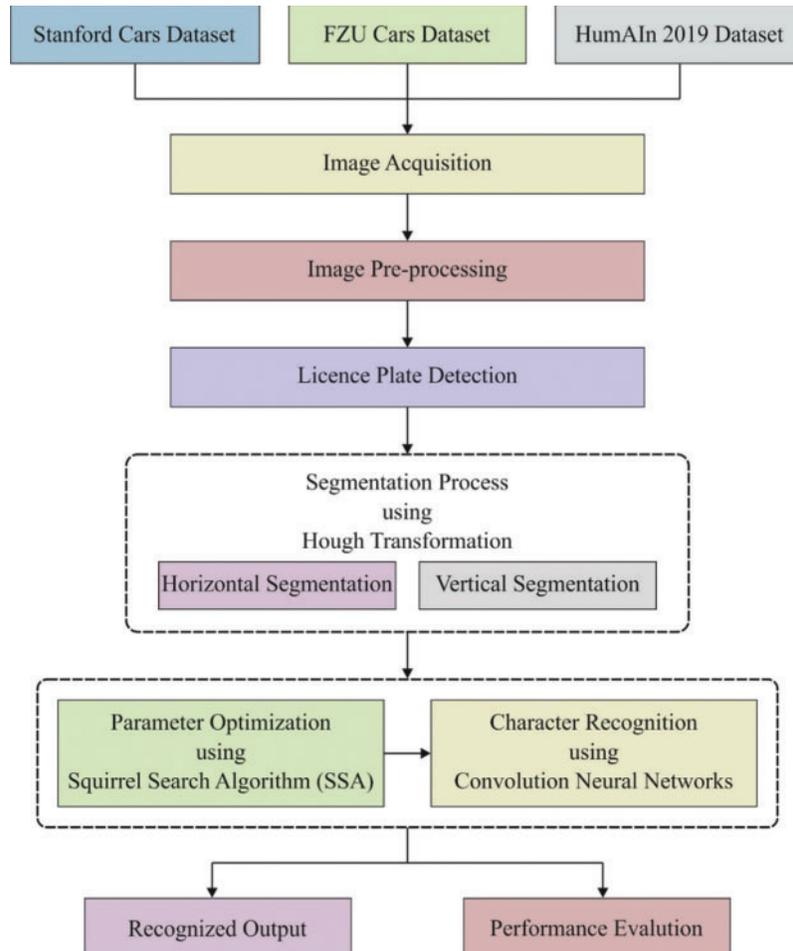


Figure 1: Block diagram of the HT-SSA-CNN model

2.2.1 Rectangle Shape Checker

The rectangle shape checker is executed to check the presence of rectangular-sized objects in the image. It verifies the sum of white pixels as $+5\%$ or -5% that is fixed as a threshold for the correct region of such areas.

2.2.2 Dimension of Plate Checker

Verify if $(a < \text{height/width of the succeed region} < b)$. Here the values of (a, b) parameters are based on the dimensions of LP. It is pointed out that when the predicted regions are not assumed as a plate, then the detection step is initialized. This green channel offers sufficient image contrast, blurs the image for LP edge smoothening and discard the artifacts.

2.3 Character Segmentation Using Hough Transform

Hough Transform (HT) is mainly employed to examine the lines in the images. The pixels in image space (x_0, y_0) can be represented by applying the transformation,

$$r = x \cdot \cos \theta + y \cdot \sin \theta \quad (1)$$

A curve $r = x_0 \cdot \cos \theta + y_0 \cdot \sin \theta$ is touted to have attained the parameter space (θ, r) . Once it is converted into parameter space, it gets 'n' curves from parameter space. While these curves exceed (θ_0, r_0) , then the 'n' points of the image space exist on a line that can be identified in image space, by exploring the cross points from parameter space.

A plate image with greater rotation cannot be used to accomplish horizontal segment lines. Thus, in case of a single character, the rotation shows a minimum impact on horizontal projection. Therefore, the horizontal segmentation model is provided below:

- Identify the valleys of vertical projection and classify the plate image as massive blocks in a vertical fashion. This classification would be unfit due to the frame and rivet.
- Identify horizontal segmentation lines for all blocks under the investigation of the horizontal projection of a block. This line is applied for a block of subsection line.
- Apply HT on all subsection lines which tends to eliminate unwanted lines and integrate the supreme subsection lines into a full line.

This model provides massive benefits. Initially, HT applies a voting strategy in which there are minimum numbers of incorrect subsection lines present and these lines can be discarded. In contrast, the linear fitting approach is highly sensitive for ineffective subsection lines. Later, it is assumed to be a local projection model that reduces the efficiency of background, brightness variation, and plate rotation. Obviously, the rotation correction process leads to image degradation and pose 'character analysis', a complex operation. Vertical segmentation method depends upon the projection investigation that is constrained by advanced knowledge. The size of LP is 440×140 (mm), where every character is 45×90 (mm), and the range among these characters is 12 (mm) while the big interval is (34 mm) from first 2 characters and last 5 characters. This kind of information is named as 'prior knowledge'. When prior knowledge is applied, segmentation becomes highly effective. There are four steps followed in vertical segmentation approach:

- Explore the candidates for vertical segmentation lines. A candidate is identified for all valleys of vertical implication.
- Evaluate the size of the plate and character using horizontal segmentation lines as well as the candidates.
- Determine both left and right borders by applying prior knowledge. The variance of gray level of the pixel with segmentation line is small. This has to be deployed in plate interval and pixels since it exceeds the background pixels with the same gray level. Thus, the vertical segmentation lines for big intervals can be reached by exploring the desired positions and finding better segmentation lines and lower variance from candidates.
- Alternate vertical segmentation lines could be placed similarly.

2.4 SSA-CNN Based Character Recognition in LP

CNN is a well-known DL approach applied for character analysis since it finds the segmented LPs. Fig. 2 demonstrates the CNN with conv, pooling and FC layers. These three layers are applied in CNN development along with various counts of blocks as well as presence or absence of blocks [25–28].

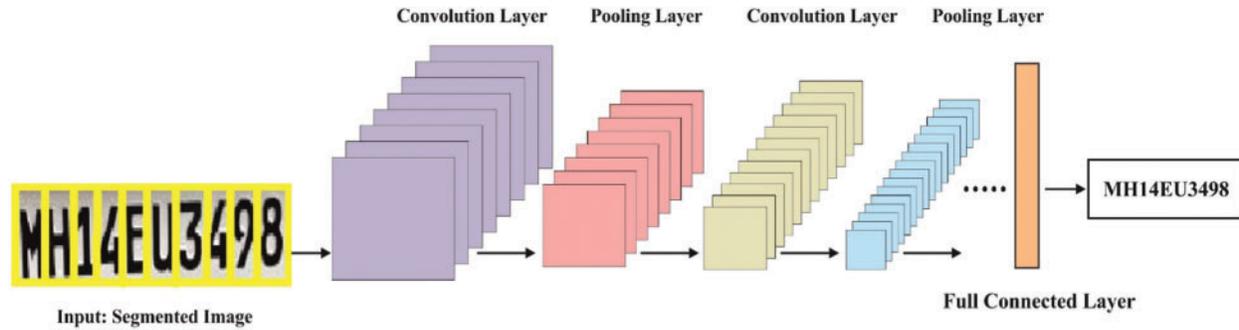


Figure 2: The working structure of CNN

Conv layer

It is different for each NN and not all the pixels are connected to weights and biases. Hence, the image is divided as minimum parts, whereas the weights and biases are used. These are named as filters or kernel that undergoes convolution with a smaller input image and it provides the feature map. The filters are assumed as elegant 'features' that are explored from the input image and the conv layer. The parameter has to execute the convolution process, which is a daunting operation as same filters are traversed. The local region size, filter value, padding, and stride are some of the hyper parameters in conv layer.

Pooling layer

In order to reduce the spatial size of an image, parameter value and computational cost, the pooling layer is employed. It is divided into average pooling, stochastic pooling and max pooling. The first type of pooling is mainly applied for $n \times n$ window that is slid by input with stride s . Moreover, the maximum value in $n \times n$ region is used and the size of input data is reduced. Finally, it offers conventional invariance, so that a small variation is also examined.

FC layer

The input of FC layer is assumed to be the result of last pooling layer. It is treated as a CNN where all the neurons of primary layer are connected to present layer. Hence, the parameter value in a layer is maximum, when compared to conv. layer. It is connected with the output layer named as classification method.

Activation function

Different activation functions are used with various structures of CNN. The non-linear activation functions, named as ReLU, LReLU, PReLU, and Swish, are accessible ones. It helps to increase the efficiency of the training process. Additionally, ReLUs function is highly efficient when compared to alternate models. Tuning hyper-parameters for CNN is highly sensitive and complex since it is slow in training a CNN and it has numerous parameters for configuration as given below:

- Learning rate
- Epoch count
- Batch size
- Activation function
- Count of hidden layers and units
- Weight initialization

2.5 Hyper Parameter Tuning Using SSA

SSA algorithm is applied to tune the parameters of CNN as discussed earlier. SSA model is evolved from the foraging behavior of flying squirrels. These tiny creatures use an effective method to travel a long distance. In warm weather, a squirrel changes the place by jumping from one tree to another in the forest and search for foods. It simply determines acorn nuts to meet every day's energy requirements. Then, it starts exploring hickory nuts (a better food source) which are saved for winter. In cold weather, it becomes weaker while it regains the power by consuming hickory nuts. When the warm weather turns up again, squirrels become active and energetic. The predefined procedures are duplicated and followed at the time of food searching. Depending upon the food foraging nature of squirrels [29], the optimized SSA is modeled with subsequent stages mathematically. Fig. 3 shows the flowchart for SSA model.

2.5.1 Initialization Phase

The important parameters of SSA are highest count of iteration $Iter_{max}$, population size NP, decision variable count n, the predator occurrence possibility P_{dp} , scaling factor sf, gliding constant G_c , and upper as well as lower bounds for decision variable, i.e., FS_U and FS_L . The above mentioned parameters are initialized from the beginning of SSA process.

2.5.2 Location Initialization Phase

The flying squirrels' positions are arbitrarily loaded in the searching space as following:

$$FS_{i,j} = FS_L + \text{rand}() * (FS_U - FS_L), \quad i = 1, 2, \dots, NP, \quad j = 1, 2, \dots, n \quad (2)$$

where $\text{rand}()$ shows uniformly-distributed arbitrary values from [0,1]. A fitness value $f = (f_1, f_2, \dots, f_{NP})$ of a separate flying squirrel's position is computed by replacing the decision variables' value into a fitness function:

$$f_i = f_i (FS_{i,1}, FS_{i,2}, \dots, FS_{i,n}), \quad i = 1, 2, \dots, NP \quad (3)$$

After that, the food sources' quality is determined with fitness value of flying squirrels' place as given below:

$$[\text{sorted}_f, \text{sorte_index}] = \text{sort} (f) \quad (4)$$

Next, the arrangement of food sources of all the flying squirrels takes place. There are three varieties of trees namely, oak tree (acorn nuts), hickory tree (hickory nuts) and normal tree. A place of the optimal food source (i.e., minimum fitness) is considered as hickory nut tree (FS_{hr}), the places of subsequent food sources are assumed to be acorn nut trees (FS_{ar}), and the remaining are named as normal trees (FS_{nt}):

$$FS_{ht} = FS(\text{sorte-index}(1)) \quad (5)$$

$$FS_{at}(1:3) = FS(\text{sorte-index}(2:4)) \quad (6)$$

$$FS_{nt}(1:NP - 4) = FS(\text{sorte-index}(5:NP)) \quad (7)$$

2.5.3 Location Creation Phase

The three situations which can show the dynamic gliding procedure of flying squirrels are discussed below [30].

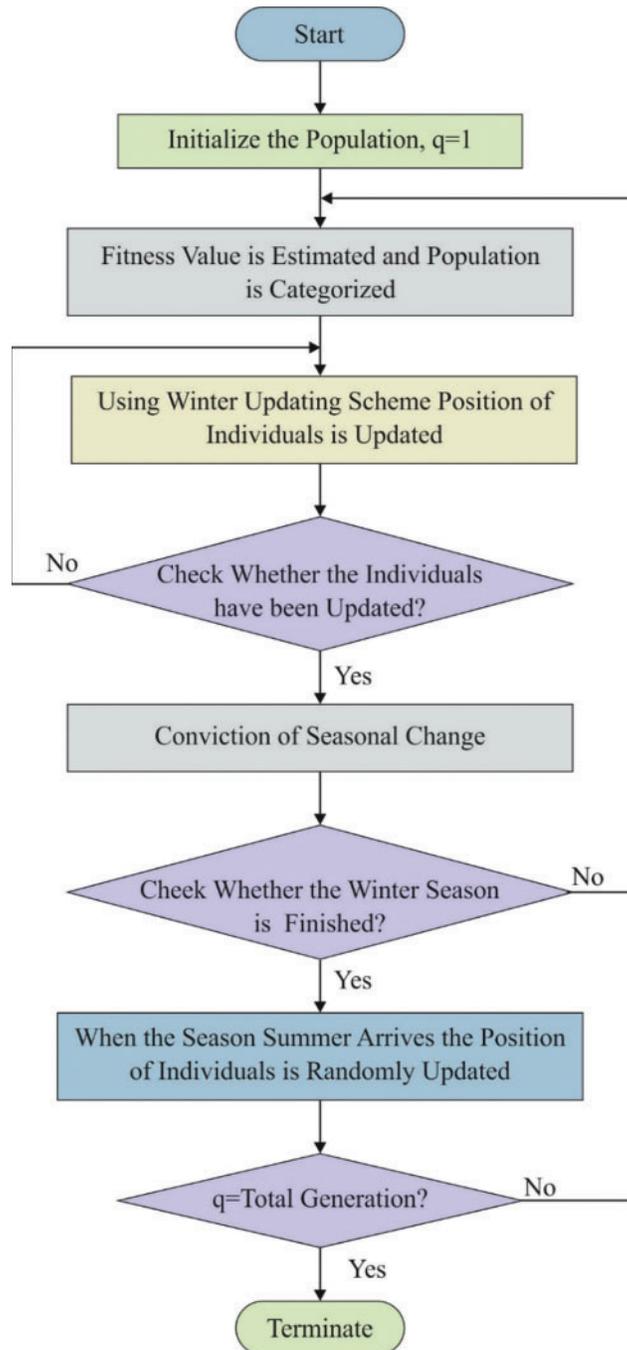


Figure 3: The flowchart of SSA

Scenario 1. The flying squirrels on acorn nut trees move to hickory nut tree. A novel location is created as following:

$$FS_{at}^{new} = \begin{cases} FS_{at}^{old} + d_g G_c (FS_{ht}^{old} - FS_{at}^{old}) & \text{if } R1 \geq P_{dp} \\ \text{randomlocation} & \text{otherwise} \end{cases} \quad (8)$$

where d_g is arbitrary gliding distance, R_1 is a function that provides the value of uniform distribution within $[0, 1]$, and G_c refers a gliding constant.

Scenario 2. Squirrels on normal trees go to acorn nut tree for collecting the essential food. A novel location is created as following:

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c (FS_{at}^{old} - FS_{nt}^{old}) & \text{if } R_2 \geq P_{dp} \\ \text{randomlocation,} & \text{otherwise} \end{cases} \tag{9}$$

where R_2 denotes a function that returns a value of uniform distribution from $[0,1]$.

Scenario 3. Few flying squirrels on normal trees move to hickory nut tree, while it satisfies the regular objectives. During this condition, a novel place of squirrels is created as following:

$$FS_{nt}^{new} = \begin{cases} FS_{nt}^{old} + d_g G_c (FS_{ht}^{old} - FS_{nt}^{old}) & \text{if } R_3 \geq P_{dp} \\ \text{randomlocation} & \text{otherwise} \end{cases} \tag{10}$$

where R_3 implies a function which gives the value of uniform distribution from $[0,1]$. In all the scenarios, gliding distance d_g should exist between 9–20 m. But, these values are somewhat huge and can initiate huge perturbations in (8)–(10) a. To obtain suitable action of the technique, a scaling factor (sf) is treated as a divisor of d_g with the value 18.

2.5.4 Seasonal Monitoring Criteria Validation

The foraging behaviour of flying squirrels is considerably concerned with varying seasons. So, seasonal monitoring is mainly employed to get rid of the trapping from better local results. The seasonal constant S_c^t and the lowest value are computed initially:

$$S_c^t = \sqrt{\sum_{k=1}^n (FS_{at,k}^t - FS_{ht,k}^t)^2}, \quad t = 1, 2, 3 \tag{11}$$

$$S_{cmin} = \frac{10E - 6}{365^{Iter/(Iter_{max})/2.5}} \tag{12}$$

For $S_c^t < S_{cmin}$, the winter is higher, and the flying squirrels lose the searching potential is random and transfer the exploring positions of food source:

$$FS_{nt}^{new} = FS_L + \text{Lévy}(n) \times (FS_U - FS_L) \tag{13}$$

where Lévy distribution is an effective numerical device to improve the global search for optimization techniques:

$$\text{Lévy}(x) = 0.01 \times \frac{r_a \times \sigma}{|r_b|^{1/\beta}} \tag{14}$$

$$\sigma = \left(\frac{\Gamma(1 + \beta) \times \sin(\pi\beta/2)}{\Gamma((1 + \beta)/2) \times \beta \times 2^{((\beta-1)/2)}} \right)^{1/\beta} \tag{15}$$

where $\Gamma(x) = (x - 1)!$, r_a and r_b are two functions that return the values of uniform distribution from $[0,1]$ and β signifies a constant ($\beta = 1.5$).

2.5.5 Stopping Criterion

This technique is terminated when the large number of iterations are fulfilled. Or else, the behaviour of creating novel locations as well as ensuring seasonal monitoring situation become repetitive.

3 Experimental Validation

The presented HT-SSA-CNN method was stimulated using PC configured with i5, 8th generation and 16GB RAM. Fig. 4 shows some of the test images. The proposed model was simulated using Python 3.6.5 tool with some packages namely tensorflow (GPU-CUDA Enabled), keras, numpy, pickle, matplotlib, sklearn, pillow and opencv-python. The performance analysis showed that a total of three datasets was applied. The initial Stanford Cars dataset contains 297 model cars with 43,615 images. Secondly, 196 model cars with 16,185 images were deployed in this study. Finally, the images from HumAIn 2019 Challenge dataset (<https://campuscommune.tcs.com/enin/intro/contests/tcs-humain-2019>) were utilized.

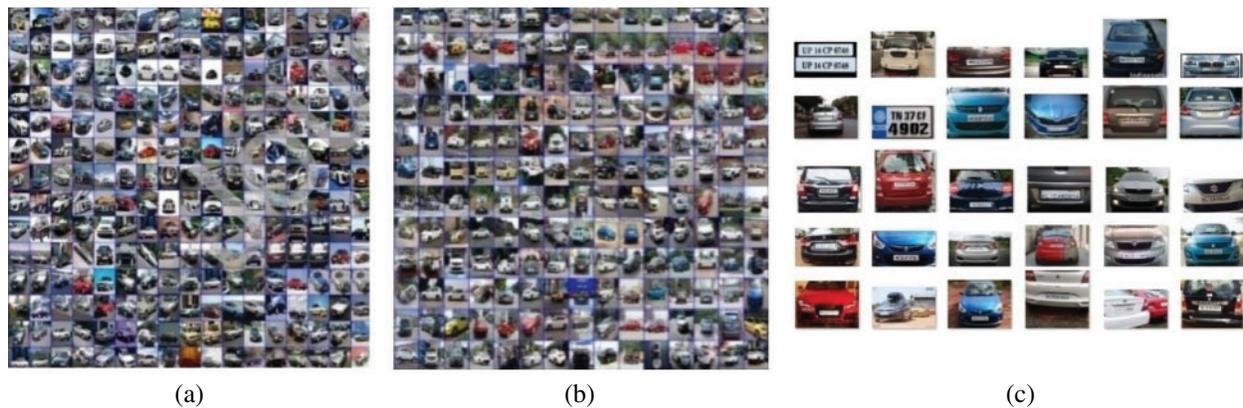


Figure 4: Sample test images. (a) Stanford cars dataset, (b) FZU cars dataset, (c) HumAIn 2019 challenge dataset

Fig. 5 shows the visualization of the results yielded by the proposed model for the given set of applied input images. The figure infers that the LPs were clearly detected, characters in LP were properly segmented and the characters were effectively recognized by the proposed model.

Figs. 6–8 offered comprehensive LP detection results of the HT-SSA-CNN model over other compared methods [14] on the applied datasets. Fig. 6 illustrates the results of the analysis for HT-SSA-CNN model in terms of different measures for the given FZU Cars dataset. The figure portrays that the ZF model was the ineffective performer since it attained the minimal precision value of 0.916 and the recall value of 0.948. It is noted that the VGG16 model achieved slightly better detection outcome with precision value of 0.925 and recall value of 0.955. Further the ResNet50 model achieved certain level of effectiveness by exhibiting a precision of 0.938 and recall of 0.951. Followed by, the ResNet 101 mode was found to be superior to earlier models with the precision of 0.945 and recall of 0.958. At the same time, the DA-Net136, DA-Net160, DA-Net168 and DA-Net200 frameworks computed an outstanding performance when compared to earlier models and attained closer precision values of 0.961, 0.965, 0.966 and 0.969; recall values of 0.964, 0.966, 0.968 and 0.971 respectively. Though these methods tried to produce better detection rate,

the proposed HT-SSA-CNN model showed proficient performance with the maximum precision of 0.981 and recall of 0.986.



Figure 5: (a) Original images (b) LP detection images (c) Segmented images (d) Recognized images

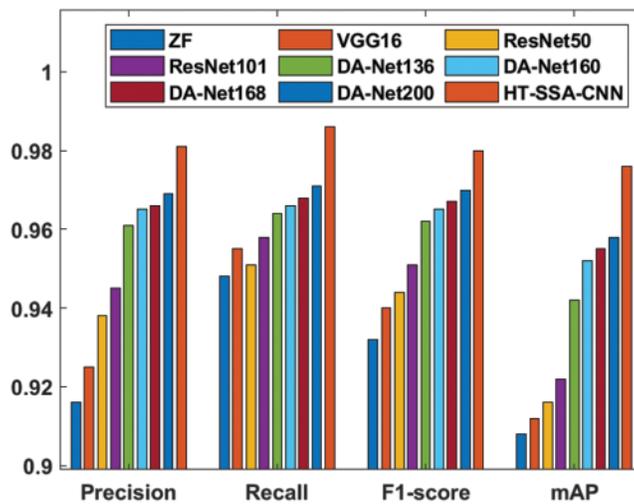


Figure 6: Results of the analysis of HT-SSA-CNN model on FZU Cars dataset

The figure also depicts the results of the analysis of HT-SSA-CNN model with respect to diverse measures on the considered FZU Cars dataset. The figure implies that ZF approach was the worst performer since it attained the least F1-score value (0.932) and mAP value (0.908). It is evident that the VGG16 model accomplished reasonable detection results with F1-score of 0.940 and mAP of 0.912. Further, the ResNet50 model resulted in specific efficiency by yielding a F1-score value of 0.944 as well as mAP value of 0.916. Then, the ResNet 101 model got qualified than previous approaches by attaining the F1-score of 0.951 and mAP value of 0.922. Simultaneously, the DA-Net136, DA-Net160, DA-Net168 and DA-Net200 methods performed well compared to previous models and achieved closer F1-score values of 0.962, 0.965, 0.967 and 0.970; mAP values of 0.942, 0.952, 0.955 and 0.958 correspondingly. Even though these approaches attempted to accomplish the best detection rate, the projected HT-SSA-CNN model showcased immense performance with a maximum F1-score of 0.980 as well as mAP value of 0.976.

Fig. 7 shows the results of HT-SSA-CNN framework by means of different measures for the considered Stanford Cars dataset. From the figure, it is clear that the ZF method was the inefficient performer since it obtained minimal precision of 0.856 and recall of 0.897. Clearly, VGG16 model accomplished gradual detection outcome with the precision value of 0.911 and recall score of 0.954. Additionally, the ResNet50 model yielded certain level of efficiency by yielding a precision of 0.912 and recall of 0.946. Besides, the ResNet 101 mode produced excellent value than the existing models with precision value of 0.941 and recall value of 0.952. Meanwhile DA-Net136, DA-Net160, DA-Net168 and DA-Net200 frameworks outperformed the previous models and reached closer precision values of 0.953, 0.949, 0.962 and 0.955; recall values of 0.962, 0.954, 0.967 and 0.959 respectively. Although these methods tried to exhibit better detection rate, the proposed HT-SSA-CNN model showcased effective performance with higher precision value of 0.979 and recall value of 0.989. The figure demonstrated the results of the analysis of HT-SSA-CNN model with respect to different measures for the given Stanford Cars dataset. The figure shows that the ZF technology was the ineffective performer since it accomplished a low F1-score of 0.876 and mAP of 0.872. It is observed that the VGG16 model accomplished considerable detection results with the F1-score of 0.932 and mAP of 0.907. Moreover, it is obvious that the ResNet50 model resulted in specific efficiency by showing an F1-score of 0.929 and mAP value of 0.918. Next, the ResNet 101 approach was found to be superior to existing models by achieving the F1-score of 0.946 and mAP of 0.909. Simultaneously, DA-Net136, DA-Net160, DA-Net168 and DA-Net200 methodologies performed quite well when compared to previous models and accomplished closer F1-score values of 0.957, 0.951, 0.964 and 0.957; mAP values of 0.932, 0.938, 0.945 and 0.941 correspondingly. In addition to better detection rate, the proposed HT-SSA-CNN model also implied remarkable performance with high F1-score of 0.969 and mAP value of 0.953.

Fig. 8 shows the results of the analysis of HT-SSA-CNN model in terms of various measures on the given HumAIn 2019 dataset. From the figure, it is clear that ZF model was the worst performer with minimum precision of 0.863 and recall of 0.873. It is pointed that the VGG16 model further accomplished moderate prediction outcome with the precision of 0.869 and recall of 0.889. It is also noted that the ResNet50 model resulted in certain level of efficiency by implying a precision value of 0.871 and recall value of 0.892. Besides, the ResNet 101 mode was found to be optimal than earlier models by attaining the precision value of 0.913 and recall value of 0.923. Meanwhile, DA-Net136, DA-Net160, DA-Net168 and DA-Net200 approaches outperformed the traditional models and reached closer precision values of 0.923, 0.936, 0.932 and 0.945; recall values of 0.931, 0.942, 0.948 and 0.957 correspondingly. Though these models attempted to yield

the best prediction rate, the presented HT-SSA-CNN model showcased effective performance with high precision value of 0.961 and recall value of 0.968.

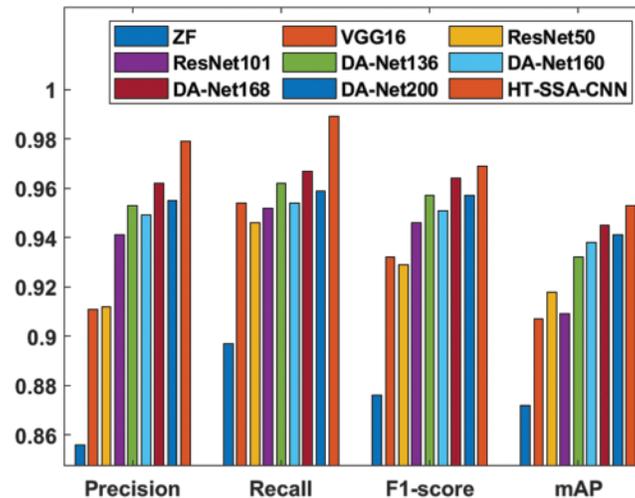


Figure 7: Results of the analysis of HT-SSA-CNN model on Stanford cars dataset

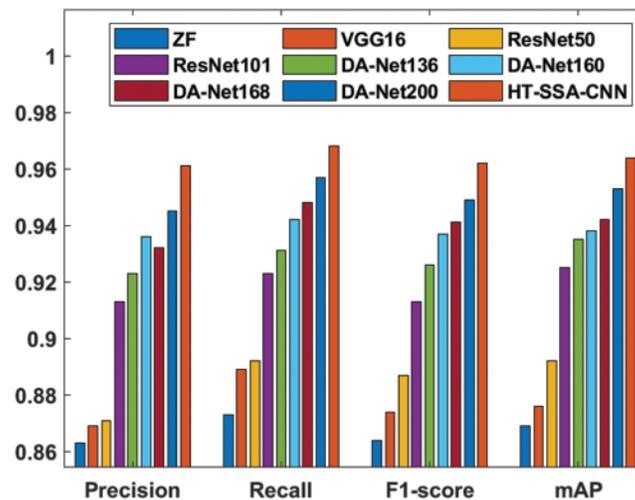


Figure 8: Results of the analysis of HT-SSA-CNN model on HumAIn 2019 dataset

The figure also shows the results of the analysis of HT-SSA-CNN model by means of different measures on applied HumAIn 2019 Cars dataset. From the figure, it is clear that the ZF method was the poor performer since it achieved a low F1-score of 0.864 and mAP value of 0.869. It is pointed out that the VGG16 scheme accomplished better detection results with F1-score of 0.874 and mAP of 0.876. Additionally, it is notified that the ResNet50 model yielded a certain level of efficiency by achieving F1-score of 0.887 and mAP of 0.892. Next, the ResNet 101 mode was found to be supreme than the earlier models by attaining the F1-score of 0.913 and mAP of 0.925. Concurrently, DA-Net136, DA-Net160, DA-Net168 and DA-Net200 methodologies

performed quite-well than conventional models and reached closer F1-score values of 0.926, 0.937, 0.941 and 0.949; mAP values of 0.935, 0.938, 0.942 and 0.953 correspondingly. Though these technologies attempted to accomplish the optimal detection rate, the presented HT-SSA-CNN model implied effective performance with high F1-score of 0.962 and mAP of 0.964.

Fig. 9 depicts the competing results of the analysis of HT-SSA-CNN method by means of overall accuracy. The figure indicates that the ZF model was the worst performer as it attained the least overall accuracy of 0.942%. The ResNet 101 model was found to be superior to previous models by acquiring the overall accuracy of 0.943%. Simultaneously, the VGG_CNN_M_1024 model outperformed the existing models and achieved overall accuracy value of 0.967%. Further the VGG16 model achieved manageable detection results with an overall accuracy of 0.971%. The ResNet50 model led to a certain level of efficiency by showing an overall accuracy of 0.976%. Though these methods attempted to achieve a better detection rate, the projected HT-SSA-CNN model implied the best performance with optimal overall accuracy of 0.983%. The experimentation outcome verified that the presented method yielded better results under several aspects. The projected HT-SSA-CNN model provided the best performance with high precision of 0.981, 0.979 and 0.961 on Stanford Cars, FZU Cars as well as HumAIn 2019 Challenge datasets. Therefore, it can be employed as an effective tool to recognize the LPs in real-time environment.

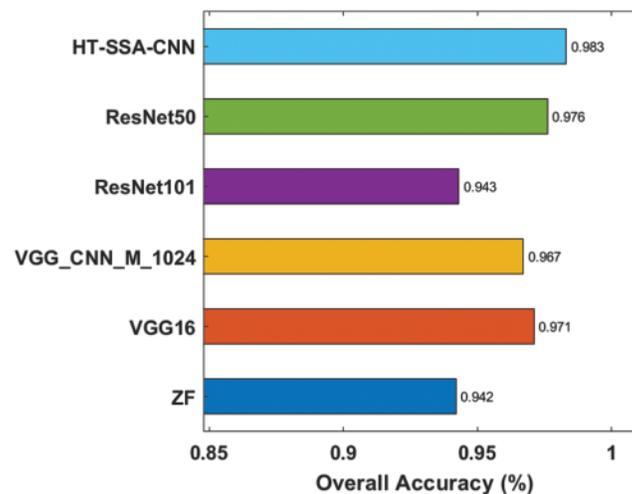


Figure 9: The overall accuracy analysis of HT-SSA-CNN model with the existing model

4 Conclusion

The current study introduced a robust DL-based VLPR model using SSA-CNN model. The proposed model had a total of four major processes namely preprocessing, LP localization and detection, HT-based character segmentation, and SSA-CNN based recognition. The input image was preprocessed to make it compatible with further processing. The preprocessed image was then fed into LP localization process to detect and crop the LP effectively. Followed by, HT was applied for character segmentation in the LP. At last, SSA-CNN model was applied to examine the characters in the classified image. The SSA algorithm was applied to CNN model to choose the hyper parameters properly and effectually recognize the characters that exist in the segmented image. The proposed HT-SSA-CNN approach achieved higher precision of 0.981, 0.979 and 0.961

on Stanford Cars, FZU Cars as well as HumAIn 2019 Challenge datasets. In future, the projected HT-SSA-CNN technique can be extended using DL models other than CNN.

Acknowledgement: The author Irina Pustokhina is thankful to the Department of Entrepreneurship and Logistics, Plekhanov Russian University of Economics, Moscow, Russia. The author K. Shankar would like to thank RUSA PHASE 2.0, Department of Edn., Government of India.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] C. N. E. Anagnostopoulos, "License plate recognition: A brief tutorial," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 1, pp. 59–67, 2014.
- [2] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [3] S. Yu, B. Li, Q. Zhang, C. Liu and M. Q. H. Meng, "A novel license plate location method based on wavelet transform and EMD analysis," *Pattern Recognition*, vol. 48, no. 1, pp. 114–125, 2015.
- [4] S. R. Zhou, W. L. Liang, J. G. Li and J. U. Kim, "Improved VGG model for road traffic sign recognition," *Computers, Materials & Continua*, vol. 57, no. 1, pp. 11–24, 2018.
- [5] I. Giannoukos, C. N. Anagnostopoulos, V. Loumos and E. Kayafas, "Operator context scanning to support high segmentation rates for real time license plate recognition," *Pattern Recognition*, vol. 43, no. 11, pp. 3866–3878, 2010.
- [6] A. M. A. Ghaili, S. Mashohor, A. R. Ramli and A. Ismail, "Vertical-edge-based car-license-plate detection method," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 26–38, 2013.
- [7] G. S. Hsu, J. C. Chen and Y. Z. Chung, "Application-oriented license plate recognition," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 552–561, 2013.
- [8] H. Li, P. Wang, M. You and C. Shen, "Reading car license plates using deep neural networks," *Image and Vision Computing*, vol. 72, pp. 14–23, 2018.
- [9] Y. Yuan, W. Zou, Y. Zhao, X. Wang, X. Hu *et al.*, "A robust and efficient approach to license plate detection," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1102–1114, 2017.
- [10] J. Jiao, Q. Ye and Q. Huang, "A configurable method for multi-style license plate recognition," *Pattern Recognition*, vol. 42, no. 3, pp. 358–369, 2009.
- [11] C. Gou, K. Wang, Y. Yao and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted boltzmann machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1096–1107, 2016.
- [12] O. Bulan, V. Kozitsky, P. Ramesh and M. Shreve, "Segmentation-and annotation-free license plate recognition with deep localization and failure identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2351–2363, 2017.
- [13] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers *et al.*, "What makes good synthetic training data for learning disparity and optical flow estimation?," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 942–960, 2018.
- [14] I. V. Pustokhina, D. A. Pustokhin, J. J. Rodrigues, D. Gupta, A. Khanna *et al.*, "Automatic vehicle license plate recognition using optimal k-means with convolutional neural network for intelligent transportation systems," *IEEE Access*, vol. 8, pp. 92907–92917, 2020.
- [15] S. R. Zhou, W. L. Liang, J. G. Li and J. U. Kim, "Improved VGG model for road traffic sign recognition," *Computers, Materials & Continua*, vol. 57, no. 1, pp. 11–24, 2018.

- [16] C. Feng, S. Arshad, S. Zhou, D. Cao and Y. Liu, "Wi-multi: A three-phase system for multiple human activity recognition with commercial WIFI devices," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7293–7304, 2019.
- [17] X. Wang, L. Wang, S. Li and J. Wang, "An event-driven plan recognition algorithm based on intuitionistic fuzzy theory," *Journal of Supercomputing*, vol. 74, no. 12, pp. 6923–6938, 2018.
- [18] S. R. Zhou and B. Tan, "Electrocardiogram soft computing using hybrid deep learning CNN-ELM," *Applied Soft Computing*, vol. 86, pp. 105778–105789, 2020.
- [19] W. Li, Y. Cao, J. Chen and J. Wang, "Deeper local search for parameterized and approximation algorithms for maximum internal spanning tree," *Information and Computation*, vol. 252, pp. 187–200, 2017.
- [20] J. Zhang, W. Wang, C. Lu, J. Wang and A. K. Sangaiah, "Lightweight deep network for traffic sign classification," *Annals of Telecommunications*, vol. 75, no. 7–8, pp. 369–379, 2020.
- [21] P. He, Z. Deng, C. Gao, X. Wang and J. Li, "Model approach to grammatical evolution: Deep-structured analyzing of model and representation," *Soft Computing*, vol. 21, no. 18, pp. 5413–5423, 2017.
- [22] R. Sun, L. Shi, C. Yin and J. Wang, "An improved method in deep packet inspection based on regular expression," *Journal of Supercomputing*, vol. 75, no. 6, pp. 3317–3333, 2019.
- [23] C. Yin, H. Wang, X. Yin, R. Sun and J. Wang, "Improved deep packet inspection in data stream detection," *Journal of Supercomputing*, vol. 75, no. 8, pp. 4295–4308, 2019.
- [24] B. B. Yousif, M. M. Ata, N. Fawzy and M. Obaya, "Toward an optimized neutrosophic k-means with genetic algorithm for automatic vehicle license plate recognition (ONKM-AVLPR)," *IEEE Access*, vol. 8, pp. 49285–49312, 2020.
- [25] S. K, L. S. K., A. Khanna, S. Tanwar, J. J. P. C. Rodrigues *et al.*, "Alzheimer detection using group grey wolf optimization based features with convolutional classifier," *Computers & Electrical Engineering*, vol. 77, pp. 230–243, 2019.
- [26] M. Elhoseny and K. Shankar, "Optimal bilateral filter and convolutional neural network based denoising method of medical image measurements," *Measurement*, vol. 143, pp. 125–135, 2019.
- [27] N. Krishnaraj, M. Elhoseny, M. Thenmozhi, M. M. Selim and K. Shankar, "Deep learning model for real-time image compression in Internet of Underwater Things (IoUT)," *Journal of Real-Time Image Processing*, vol. 17, no. 6, pp. 1–15, 2020.
- [28] B. S. Murugan, M. Elhoseny, K. Shankar and J. Uthayakumar, "Region-based scalable smart system for anomaly detection in pedestrian walkways," *Computers & Electrical Engineering*, vol. 75, pp. 146–160, 2019.
- [29] T. Zheng and W. Luo, "An improved squirrel search algorithm for optimization," *Complexity*, vol. 2019, pp. 1–31, 2019.
- [30] M. Jain, V. Singh and A. Rani, "A novel nature-inspired algorithm for optimization: Squirrel search algorithm," *Swarm and Evolutionary Computation*, vol. 44, pp. 148–175, 2019.