**International Journal of Electrical and Computer Engineering (IJECE)**
Vol. 11, No. 6, December 2021, pp. 5266~5276
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i6.pp5266-5276 

# Multiclassification of license plate based on deep convolution neural networks

**Masar Abed Uthaib[1], Muayad Sadik Croock[2]**
[1]Iraqi Commission for Computers and Informatics (ICCI), Informatics Institute for Postgraduate Studies, Baghdad, Iraq
[2]Control and Systems Engineering Department, University of Technology, Baghdad, Iraq

**Article Info**

***Article history:***

Received Jul 11, 2020
Revised May 12, 2021
Accepted Jun 6, 2021

***Keywords:***

Adam optimizer
Convolution neural networks
Data augmentation
Dropout
License plate classification

**ABSTRACT**

In the classification of license plate there are some challenges such that the different sizes of plate numbers, the plates' background, and the number of the dataset of the plates. In this paper, a multiclass classification model established using deep convolutional neural network (CNN) to classify the license plate for three countries (Armenia, Belarus, Hungary) with the dataset of 600 images as 200 images for each class (160 for training and 40 for validation sets). Because of the small numbers of datasets, a preprocessing on the dataset is performed using pixel normalization and image data augmentation techniques (rotation, horizontal flip, zoom range) to increase the number of datasets. After that, we feed the augmented images into the convolution layer model, which consists of four blocks of convolution layer. For calculating and optimizing the efficiency of the classification model, a categorical cross-entropy and Adam optimizer used with a learning rate was 0.0001. The model's performance showed 99.17% and 97.50% of the training and validation sets accuracies sequentially, with total accuracy of classification is 96.66%. The time of training is lasting for 12 minutes. An anaconda python 3.7 and Keras Tensor flow backend are used.



***Corresponding Author:***

Masar Abed Uthaib
Department of Computer Science
Iraqi Commission for Computers and Informatics (ICCI), Informatics Institute for Postgraduate Studies
Baghdad, Iraq
Email: Masar.uthaib2018@gmail.com

Muayad Sadik Croock
Control and Systems Engineering Department
University of Technoogy-Iraq
Baghdad, iraq
Email: muayad.s.croock@uotechnology.edu.iq

## 1. INTRODUCTION

In all manufacturing areas and in our everyday life, vehicles are commonly used. An effective way of distinguishing vehicles and make them unique by the license plate (LP). With the fast-growing number of cars, traffic violations occur more often in public transportation, such as highway or parking fraud tolls, speeding and car theft. Therefore the vehicle LPs need to be identified for protection. The information derived from an LP can be used for various purposes, like looking for the vehicles that may be stolen or might be fighting crime, monitoring crossing borders, or accessing the highway toll station. Thus, there is a need for a license plate identification system. Recently, deep learning has shown an excellent performance in

most complex tasks such as medical imaging and cyber security [1]-[4]. Convolutional neural networks (CNNs) are one of the deep learning algorithms, which is why deep learning is popular [5], [6]. The layers of CNNs layers primarily three layers [7]:

a. Convolutional-layer
b. Pooling-layer
c. Fully-connected layer

The challenges we faced through the classification of license plates: i) Different sizes of plates, ii) Small datasets that make overfitting, iii) The time for the processing CNN's model during training.

On the other hand, several studies have discussed license plate classification in recent years. Jose *et al.* [8] suggested an algorithm for the classification of Vietnamese multi-standard licensing plates based on a convolution neural network, in which transfer learning (Residual Net) was used. The final layer is deleted and changed with a new layer of classification. Categorization of the model into three groups (the Rizal monument series sticker for new vehicles, Rizal monument series, 2014 series). Cross-entropy as an optimization method as well as 0.001 represented the learning rate. The validation accuracy was 82.61%. In [9], a HAAR Feature-based Classifier was used to detect and segment the plates into characters for recognizing it by CNN. Their training and validation accuracies were 93.54% and 91.38%, respectively. The recognition accuracy was 90.90%.

Wang *et al.* [10] documented a way of identifying plates' characters by using a template matching technique and artificial neural networks. In the beginning, the secondary positioning technique was used for plate localization. The precise position of the plate depended on the vertical edge and HSV color of the plate. Their precisions approximation for the localization and recognition for the number of the plate were 75.8% and 72.5% sequentially. In artificial neural network (ANN), the accuracy of recognition was 75%.

A K-nearest neighbour classification was employed for classifying characters in the license plate [11]. Initially, Otsu method to extract plate image was used and then converting to a binary image. The test dataset was 100 images. They reached 93,75%, which is the accuracy of identifying numbers, while 91.92% accuracy of letter recognition. Wang *et al.* [12] designed a system for detection and recognizing plate numbers for the Indian license plate, where they used (You Only Look Once) Yolo v.3 for training the dataset that consists of 37 class of characters images. They used augmentation techniques to increase the number of character samples through training. The training process was accomplished by using Nvidia Giga Texel Shader eXtreme (GTX) 1080. The accuracy of recognition is 91%. Patel *et al.* [13] introduced the number plate recognition (NPR) approach based on morphological operation and the Sobel edge detection methods. The bounding box method was used to segment the numbers and letters of the plate. Template matching was used to recognize numbers and characters after segmentation. For segmenting letters and numbers in the plate bounding box technique used. The matching of templates was used to identify after segmentation numbers and characters.

Sharma in [14] have reported plate number identification using phase correlation and cross-correlation structured approaches. The regular cross-correlation approach was found to be better than the phase-correlation method to identify the vehicle number plate. The normalized cross-correlation recognition accuracy for the plate was 67.98%, while the phase correlation was 63.46%. Öztürk and Özen [15] proposed a procedure for identifying plate characters. For identifying plates, Otsu's thresholding was used. The plate segmentation was evaluated by a Horizontal and vertical histogram. Eventually, for character recognition, probabilistic neural networks was used. The recognition accuracy for the characters was 96.5%. In [16], CNN was used to recognize 16 Bangala license plate characters, and the model tested over various image samples. They used 1750 training samples and 350 for testing samples. The recognition accuracy depends on the number of epochs from (100-1000) and the samples' number from (1300-1750) so that the recognition accuracy range (70-88)%. Babu *et al.* [17] proposed artificial neural networks for recognizing the characters of the plate. In the beginning, the plate segmented by using the connected components analysis and the vertical projection into blocks of characters and numbers where the blocks resized 32×32. The segmentation and recognition accuracies were 85.4% and 78% sequentially.

In this paper, an efficient system suggested for classifying multinational license plate (Armenia, Belarus, Hungary) countries based on deep CNN than previously literature studies where the training and validation sets accuracies were 99.17% and 97.50%, respectively. The overall plate classification accuracy is 96.66% with 60 tested plate images. In computer vision, the classification of images has an essential role. CNN's are frequently used in deep learning models where feature extraction without expert human intervention. Among various methods, the proposed CNN model achieved high performance in image classification.

## 2. RESEARCH METHOD

For implementing the classification process, deep learning CNN's are applied, as illustrated in Figure 1. We considered a plate image for three countries (Armenia, Belarus, Hungary) as the input to be feeded into the CNN model. Afterward, the proposed method classifies it into (Armenia), (Belarus), and (Hungary) plate.

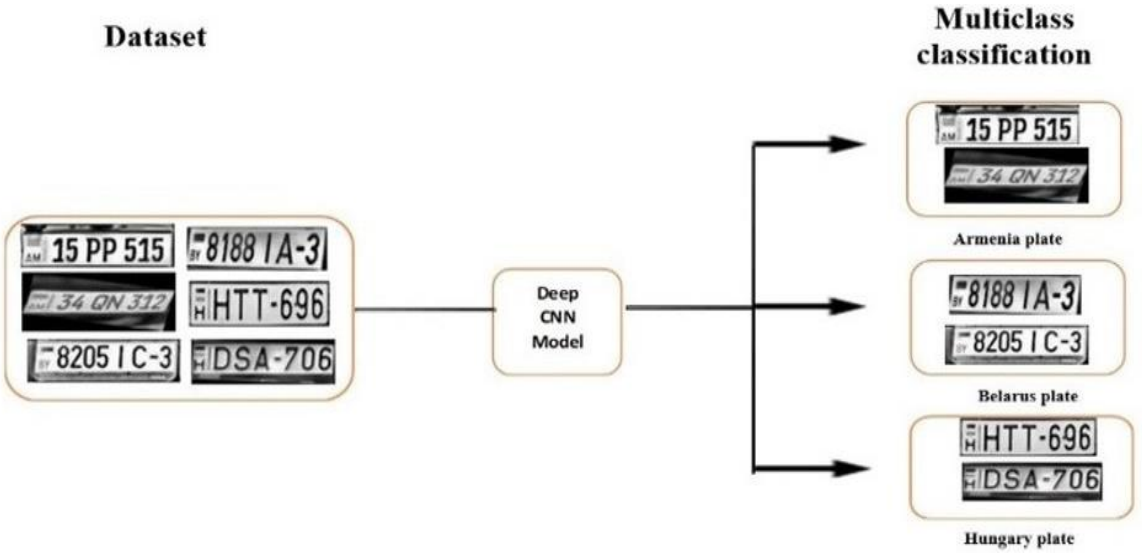



Figure 1. The proposed multinational license plate classification workflow

### 2.1. Dataset gathering

The experiment is conducted on 600 images for different license plates (Armenia, Belarus, Hungary) that are collected from the localization stage under different weather conditions. Datasets are distributed for each class of the plates. Each class has 200 images (160 plate images for the training set and 40 plate images for the validation set, as reported in Figure 2. So that training and validation sets are 80% and 20% of each class, respectively.

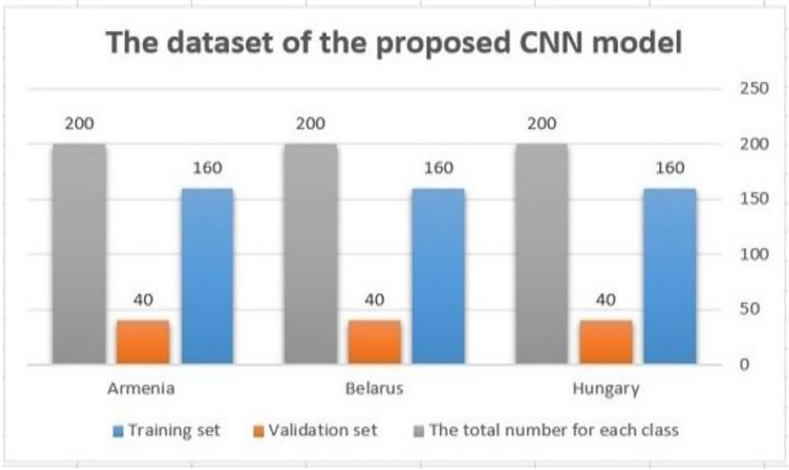



Figure 2. Datasets allocation for each class of license plates in the proposed CNN classification model

### 2.2. Pre-processing stage

A pre-processing step is taken place before images given to the CNN model. This is to minimize dimensions, computation and to show better performance. The original image is a grayscale plate with

different sizes; we uniform them to 200×100×1. The data is shuffled before splitting so that the training not only focuses on narrow data but also unsorted data in the dataset. The pixels of the image normalized, which is the changing process for intensity pixel values range. As mentioned earlier, the greyscale channel has one channel in the image (0-255) that makes the calculation complex so that it normalized image pixel to be in the range (0-1). Data augmentation techniques are used with assist Keras augmentations to increase the size and also improves the quality of the training samples, and to get rid of overfitting, where overfitting is a gap between the accuracy of training and validation sets. They also improve model efficiency and make model generality [18], [19]. Figure 3 shows the presented augmented images. Augmentations are set before training the models. We used three augmentation techniques (rotation, horizontal flip, zoom) to get new three training sets that expand the training samples dataset. The newly generated set of training is essentially the initial training images with the augmented images generated by the choices of augmentation techniques. Each augmentation is as follows [20]-[22]:

a. Rotation

The image is rotated 20° for getting better classification.

b. Horizontal Flipping

It is one of the geometric augmentation, its flips image towards the horizontal axis. Its more common than vertical flip and proven its benefit in CIFAR-10 and ImageNet datasets.

c. Zoom

We zoomed images to make features clear.


(a) (b) (c) (d)

Figure 3. Data augmentation techniques, (a) original image, (b) rotated image, (c) horizontal flip image, (d) zoom image

### 2.3. The proposed CNN classifier architecture

The suggested proposed CNN model is reported in Figure 4 with Table 1, which summarizes the CNN layers. The system contains 30 layers that started from the layer of input, which carry images from the augmentation methods in the preceding preprocessing phase. Four blocks of convolution layers that consist of (convolution as well as a Rectified linear unit (Relu) which is the function of activation), batch normalization, max-pooling, and Dropout ranges (20-25)% layers. After four blocks of convolution layers, three fully connected layers are implemented, and then the last dropout with (30%) probability before the final layer (softmax-layer) with three-classes of plates [23]. The description of each layer is illustrated below [24].

#### 2.3.1. Convolution layer

The central component in the CNN-model is the convolution layer, which has local ties and common weights. Its objectives are learning the representing of entered features. It contains various feature-maps. The similarity of the neuron features in diverse locations is used to extract the local proprieties in the previous layer's different positions. According to individual neurons, characteristics extraction is performed in the same feature map area in the preceding layer. We used a convolution filter (kernel) with different sizes (3×3, 5×5, 7×7) overlapped horizontally and vertically along with the input image to get features. The padding and stride one pixel and two steps, respectively, as shown in Figure 5, which reported convolution layer operations. After that, the result leads to the non-linear activation function that is Relu. Figure 6 clarifies the work of Relu [25].

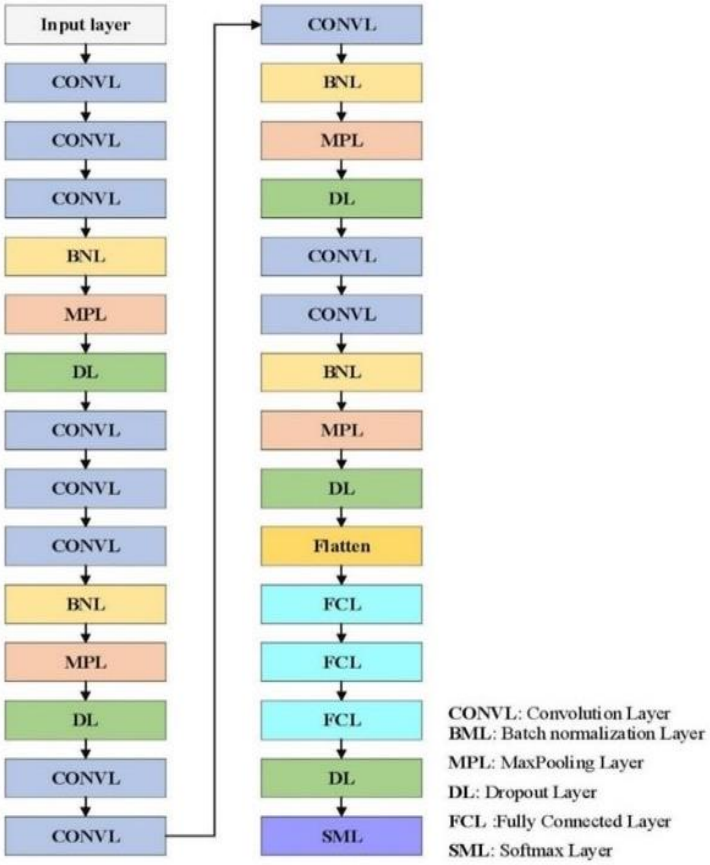



Figure 4. The proposed CNN model

Table 1. Summary of CNN's layers

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_1 (InputLayer) | (None, 100, 200, 1) | 0 |
| conv2d_1 (Conv2D) | (None, 100, 200, 32) | 320 |
| conv2d_2 (Conv2D) | (None, 100, 200, 32) | 9248 |
| conv2d_3 (Conv2D) | (None, 100, 200, 32) | 9248 |
| batch_normalization_1 (Batch | (None, 100, 200, 32) | 128 |
| max_pooling2d_1 (MaxPooling2 | (None, 50, 100, 32) | 0 |
| dropout_1 (Dropout) | (None, 50, 100, 32) | 0 |
| conv2d_4 (Conv2D) | (None, 50, 100, 64) | 51264 |
| conv2d_5 (Conv2D) | (None, 50, 100, 64) | 102464 |
| conv2d_6 (Conv2D) | (None, 50, 100, 64) | 102464 |
| batch_normalization_2 (Batch | (None, 50, 100, 64) | 256 |
| max_pooling2d_2 (MaxPooling2 | (None, 25, 50, 64) | 0 |
| dropout_2 (Dropout) | (None, 25, 50, 64) | 0 |
| conv2d_7 (Conv2D) | (None, 25, 50, 128) | 401536 |
| conv2d_8 (Conv2D) | (None, 25, 50, 128) | 802944 |
| conv2d_9 (Conv2D) | (None, 25, 50, 128) | 802944 |
| batch_normalization_3 (Batch | (None, 25, 50, 128) | 512 |
| max_pooling2d_3 (MaxPooling2 | (None, 12, 25, 128) | 0 |
| dropout_3 (Dropout) | (None, 12, 25, 128) | 0 |
| conv2d_10 (Conv2D) | (None, 12, 25, 32) | 102432 |
| conv2d_11 (Conv2D) | (None, 12, 25, 32) | 25632 |
| batch_normalization_4 (Batch | (None, 12, 25, 32) | 128 |
| max_pooling2d_4 (MaxPooling2 | (None, 6, 12, 32) | 0 |
| dropout_4 (Dropout) | (None, 6, 12, 32) | 0 |
| flatten_1 (Flatten) | (None, 2304) | 0 |
| dense_1 (Dense) | (None, 2048) | 4720640 |
| dense_2 (Dense) | (None, 1024) | 2098176 |
| dense_3 (Dense) | (None, 512) | 524800 |
| dropout_5 (Dropout) | (None, 512) | 0 |
| dense_4 (Dense) | (None, 4) | 1539 |

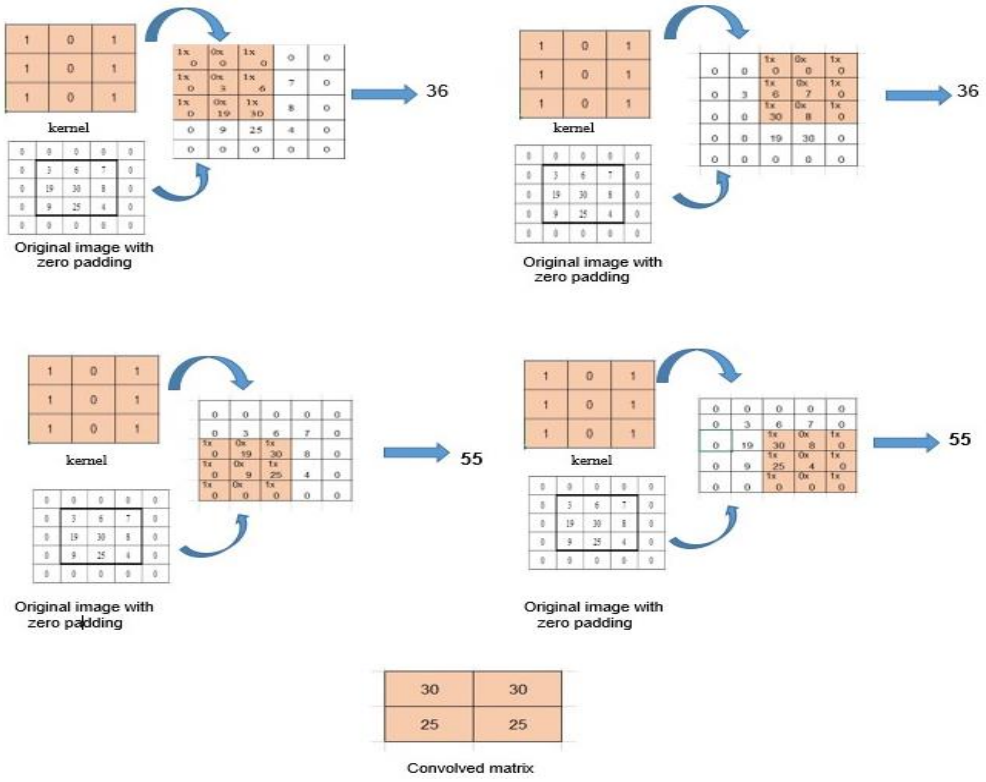



Figure 5. The example for operations of convolution layer (where the input_shape: 3×3, padding=1, kernel_ size= 3×3, Stride = 2, output_shape: 2×2)

The equation of Relu is [26]:

$$F(X) = MAX(0, X) \quad (1)$$

where
$F(X) = X$ if $x$:positive
$F(X) = 0$ if $x$:negative

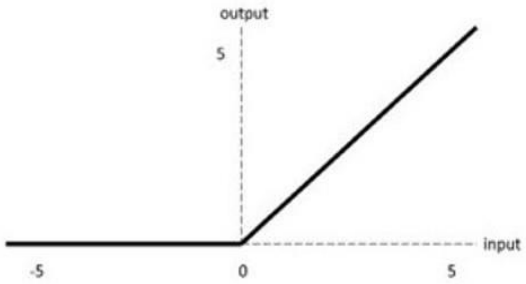



Figure 6. Relu activation function [26]

### 2.3.2. Batch normalization

Batch normalization (BN) is a strategy for normalizing activations in deep neural network intermediate layers. BN is a favorite technique in detail due to its ability to boost accuracy and accelerate preparation. It enables users to use higher levels of learning rates, preventing minor parameter changes from boosting into larger and suboptimal changes in gradient activation. Also, it prevents the training from being stuck in the saturated non-linearity regimes. It acts as a regularizer. When Batch Normalization is found during network training, there is communication among a training example, and the remaining examples in

Mini-batch and the generation of the deterministic values do not take time for the training given example so that its beneficial for network generalization [27]-[29].

### 2.3.3. Max pooling layer

The information amount in every gathered featured in the convolution layer is decreased while retaining the crucial details (commonly Multiple convolutions and pool layers rounds). Often large images are in the CNN model; therefore, we need to decrease images that minimize the number of the parameters. All the utilized max-pooling layers are (2,2), besides the stride is (2,2) to move horizontally and vertically. It is accomplished by dividing the entire image into smaller squares (2×2 the suggested method), which pass over the image with a specified string (2×2). After that, the largest value in the matrix of four numbers chosen [30], [31]. Figure 7 shows the operation Max-pooling layer.

In (2)-(4) are used for Max-pooling [32]:

$$W2 = \frac{W1-F}{S+1} \quad (2)$$

$$H2 = \frac{H1-F}{S+1} \quad (3)$$

$$D2 = D1 \quad (4)$$

where: F: spatial extend of the filter, S: stride, D1: the depth of convolution layer, D2: depth-convolution layer, W1: convolution layer width, H1: convolution layer height, W2: Max-pooling layer-width, and H2: Max pooling layer-height.

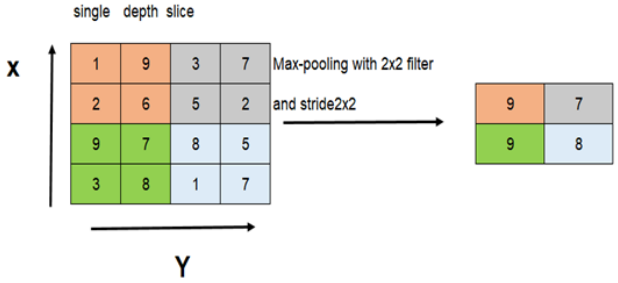



Figure 7. The operation of max-pooling layer to one block

### 2.3.4. Dropout layer

Many activations (nodes) are randomly discarded in this layer, which often dramatically accelerates the training- stage and decreasing overfitting. The probabilities for dropout ranges in our proposed-method were (20-25)% for the four Dropout-layers that follow max-pooling layers. The last dropout layer ratio after the fully connected layer is 30% [33]-[35]. Figure 8 presented Dropout layers.

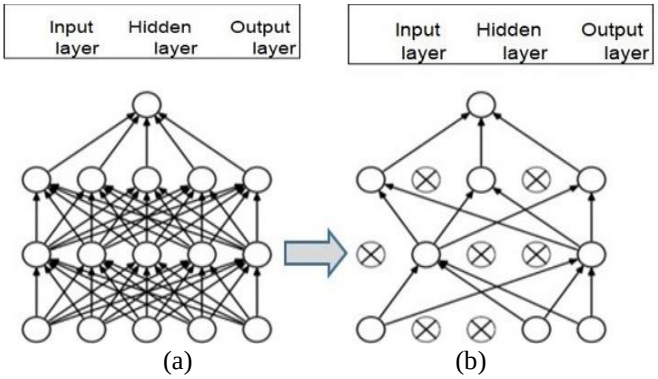



Figure 8. Example of the dropout layer: (a) standart neural net, (b) after applying dropout [36]

### 2.3.5. Fully-connected-layers

The arrangement of layers like neurons in a typical neural network is organized. A node in a fully connected-layer is attached directly to each node in the previous and next layers, as cleared in Figure 9. The connection between nodes in the previous frame of the pooling-layer and the layer of fully-connected is a vector, which represents the first layer. It contains parameters that take longer to be trained. Therefore, the number of nodes and links are eliminated using the Dropout technique with probability (30%) before the softmax layer. Softmax is a mixture of many sigmoid activation functions. Sigmoid function returns values in the range (0-1). These can be viewed as the likelihood of data points of a given class. Softmax can be used for multiclass classification problems, unlike sigmoid functions that are used for binary classification. The function returns the likelihood for every data point of all groups [37] as explained in the (5) [38]:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for j= 1,...,k .} \tag{5}$$

K: class number and the function of Z output equal to one. The output layer of the network can contain similar amounts of neurons as the number of classes within the target if we create a network or a model for multiple classifications. According to our work, the output of softmax is three-classes (Armenia, Belarus, Hungary). Figure 9 shows the softmax activation function operation. In the end, we have used the loss function, which is cross-entropy to estimate the loss of classification and to predict the label of the input image. The cross-entropy equation can be written as in (6) [39]:

$$L(y, \hat{y}) = -\sum_{k=0}^{n} yk * \log(\hat{y}k) \tag{6}$$

where $\hat{y}$ : the ratio of the predicted output layer, y the binary indicator whether the classification is correct (1) or not (0), and n: the class numbers/labels (the output nodes numbers).

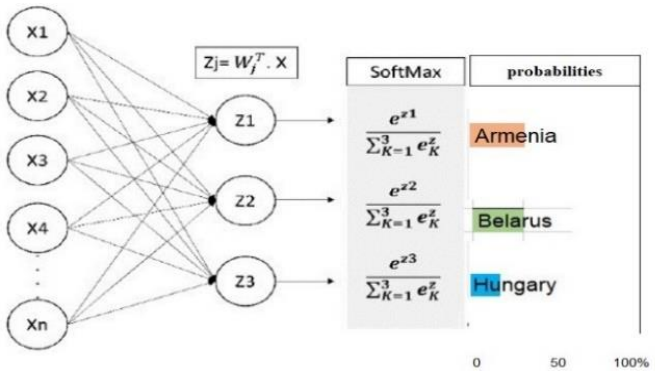



Figure 9. Softmax activation function example

### 2.4. Optimization algorithm

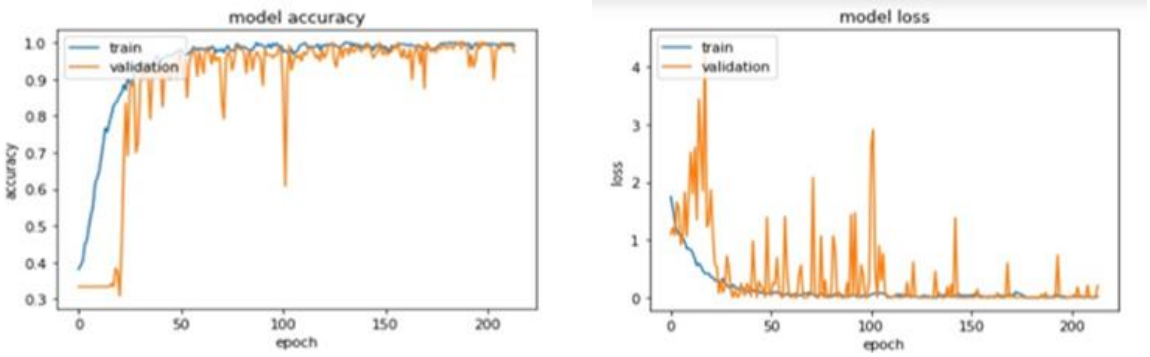
The training of CNN model by changing on the iterative basis the layer's parameters in the network. The optimizer performs a significant role in the context of learning deep CNN models. The loss function is minimized by the optimizer. We used Adam optimizers. It proves its efficiency in the learning process in a short time. It takes a small size of memory. The computation with this optimizer is done efficiently [40]-[44].

## 3. RESULTS AND DISCUSSION

The specification for hardware and software used in this study as follows: Processors: Intel R Core (TM) i7-9750 H CPU, the logical processor numbers: 12, and GPU: NVIDIA GeForce GTX 1660 TI. Moreover, we employees the following: Loss function: categorical cross-entropy, Optimization: Adam optimizer, Learning rate =0.0001, No. of Epochs: 214 Epochs, Time of training: 12 minutes, Programming by anaconda python 3.7, Keras (Tensorflow backend), and number of parameters: 9,756,675. Figure 10(a) shows both the accuracy progress during the training and validation phases for our proposed model. Figure 10(b) shows that loss has been achieved right after 214 epochs. The loss has risen and fallen. After 194 epochs, it reaches stability, where the curve begins to drop. Despite a small number of datasets

(600 image samples), the proposed model proves efficiency where the accuracy for the training and validation sets is 99.17% and 97.50 %, sequentially with efficient time (12 minutes) for training. We tested 60 plate images where the overall classification accuracy is 96.66% is computed according to (7). The comparison between the proposed classification method and other classification methods, as illustrated in Table 2. Table 3 reported the performance of the proposed CNN model in classifying multinational license plates. In order to test the accuracy of the classification of multinational license plate the (7) is adopted [11]:

$$Accuracy\ of\ classification = \frac{Number\ of\ corrected\ classified\ license\ plate}{Total\ number\ of\ samples} \times 100\% \quad (7)$$




(a) (b)

Figure 10. Training and validation accuracy and loss, (a) training and validation accuracy in various epochs, (b) training and validation loss in various epochs

Table 2. Comparison among different recognition of license plate methods

| Reference | Approach | Recognition |
|---|---|---|
| [6] | Convolution neural networks with transfer learning | 82.61% |
| [7] | Convolution neural networks | 90.90% |
| [8] | Template matching and Artificial neural networks | 72.5% and 75.8% respectively |
| [9] | K-nearest neighbors | 93.75 |
| [10] | Yolo v.3 | 91% |
| [11] | Template matching | |
| [12] | Cross and phase correlation | 67.98% and 63.46% sequentially |
| [13] | Probabilistic neural networks | 96.5% |
| [14] | Convolution neural networks | (70-88)% |
| [15] | Artificial neural networks | 78% |
| The proposed multiclassification by CNN | | 96.66% |

Table 3. The performance of the proposed CNN model for classification of multinational license plate

| The type of license plate | Total test images | No. of corrected classified license plate | No. of not corrected classified license plate | License plate classification percentage |
|---|---|---|---|---|
| Armenia | 20 | 20 | None | (20/20) ×100%=100% |
| Belarus | 20 | 19 | 1 | (19/20) ×100%=95% |
| Hungary | 20 | 19 | 1 | (19/20) ×100%=95% |
| Total | 80 | 78 | 2 | 96.66% |

## 4. CONCLUSION

An effective approach for multi-nationality vehicle plate classification based on CNNs was proposed. It focused on the preprocessing for the plate (data augmentation that increased the dataset) and regularization using (batch normalization and dropout layers) to eliminate overfitting and make generality to the model. The proposed method consisted of three stages: preprocessing, the CNN model architecture, and prediction of the class of license. The proposed method had been proved as an efficient method in performing the plate classification of the multinationally-licensed plate for the three countries (Armenia, Belarus, Hungary) where the accuracies of training and validation sets are 99.17% and 97.50%, respectively, with the overall classification accuracy is 96.66%.

## REFERENCES


[1] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, and Y. Duan, "Deep Learning Models for Classification of Red Blood Cells in Microscopy Images to Aid in Sickle Cell Anemia Diagnosis," *Electronics*, vol. 9, no. 3, 2020, Art. no. 427, doi. 10.3390/electronics9030427.

[2] L. Alzubaidi *et al.*, "Novel Transfer Learning Approach for Medical Imaging with Limited Labeled Data," *Cancers*, vol. 13, no. 7, 2021, Art. no. 1590 doi: 10.3390/cancers13071590.

[3] M. N. Yasir and M. S. Croock, "Software engineering based self-checking process for cyber security system in VANET," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 5844–5852, 2020, doi: 10.11591/ijece.v10i6.pp5844-5852.

[4] M. N. Yasir and M. S. Croock, "Cyber DoS attack-based security simulator for VANET," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, pp. 5832–5843, 2020, doi: 10.11591/ijece.v10i6.pp5832-5843.

[5] L. Alzubaidi, O. Al-Shamma, M. A. Fadhel, L. Farhan, J. Zhang, and Y. Duan, "Optimizing the Performance of Breast Cancer Classification by Employing the Same Domain Transfer Learning from Hybrid Deep Convolutional Neural Network Model," *Electronics*, vol. 9, no. 3, 2020, Art. no. 445, doi: 10.3390/electronics9030445.

[6] L. Alzubaidi *et al*., "Towards a Better Understanding of Transfer Learning for Medical Imaging: A Case Study," *Appl. Sci*, vol. 10, no. 13, 2020, Art. no. 4523, doi: 10.3390/app10134523.

[7] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, pp. 1–74, 2021, doi: 10.1186/s40537-021-00444-8.

[8] J. A. C. Jose *et al.*, "Categorizing License Plates Using Convolutional Neural Network with Residual Learning," in *2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, 2019, pp. 231–234, 10.1109/ACIRS.2019.8935997.

[9] M. Atikuzzaman, M. Asaduzzaman, and M. Z. Islam, "Vehicle Number Plate Detection and Categorization Using CNNs," in *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2019, pp. 1–5, doi: 10.1109/STI47673.2019.9068049.

[10] J. Wang, B. Bacic, and W. Q. Yan, "An effective method for plate number recognition," *Multimed. Tools Appl.*, vol. 77, no. 2, pp. 1679–1692, 2018, doi: 10.1007/s11042-017-4356-z .

[11] M. R. Hidayah, I. Akhlis, and E. Sugiharti, "Recognition Number of The Vehicle Plate Using Otsu Method and K-Nearest Neighbour Classification," *Sci. J. Informatics*, vol. 4, no. 1, pp. 66–75, 2017, doi: 10.15294/sji.v4i1.9503.

[12] J. Wang, X. Liu, A. Liu, and J. Xiao, "A deep learning-based method for vehicle licenseplate recognition in natural scene," *APSIPA Trans. Signal Inf. Process.*, vol. 8, 2019, doi: 10.1017/ATSIP.2019.8 .

[13] F. Patel, J. Solanki, V. Rajguru, and A. Saxena, "Recognition of vehicle number plate using image processing technique," *Control Syst. Eng*, vol. 2, no. 1, pp. 1–7, 2018.

[14] G. Sharma, "Performance analysis of vehicle number plate recognition system using template matching techniques," *J. Inf. Technol. Softw. Eng.*, vol. 8, no. 2, pp. 10–4172, 2018, doi: 10.4172/2165-7866.1000232.

[15] F. Öztürk and F. Özen, "A new license plate recognition system based on probabilistic neural networks," *Procedia Technol.*, vol. 1, pp. 124–128, 2012, doi: 10.1016/j.protcy.2012.02.024.

[16] M. M. S. Rahman, M. Mostakim, M. S. Nasrin, and M. Z. Alom, "Bangla License Plate Recognition Using Convolutional Neural Networks (CNN)," in *2019 22nd International Conference on Computer and Information Technology (ICCIT)*, 2019, pp. 1–6, doi: 10.1109/ICCIT48885.2019.9038597.

[17] R. N. Babu, V. Sowmya, and K. P. Soman, "Indian Car Number Plate Recognition using Deep Learning," in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies*, 2019, doi: 10.1109/ICICICT46008.2019.8993238.

[18] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*, 2018, pp. 117–122, 10.1109/IIPHDW.2018.8388338.

[19] A. Asperti and C. Mastronardo, "The effectiveness of data augmentation for detection of gastrointestinal diseases from endoscopical images," *arXiv Prepr. arXiv1712.03689*, 2017, doi: 10.5220/0006730901990205.

[20] W. Li, C. Chen, M. Zhang, H. Li, and Q. Du, "Data augmentation for hyperspectral image classification with deep CNN," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 593–597, 2018, doi: 10.1109/LGRS.2018.2878773.

[21] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, p. 60, 2019, doi: 10.1186/s40537-019-0197-0.

[22] T. C. Pham, C. M. Luong, M. Visani, and V. D. Hoang, "Deep CNN and data augmentation for skin lesion classification," in *Asian Conference on Intelligent Information and Database Systems*, 2018, pp. 573–582, doi: 10.1007/978-3-319-75420-8_54.

[23] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[24] F. Gao, T. Huang, J. Wang, J. Sun, A. Hussain, and E. Yang, "Dual-branch deep convolution neural network for polarimetric SAR image classification," *Appl. Sci.*, vol. 7, no. 5, p. 447, 2017, doi: 10.3390/app7050447.

[25] A. A. M. Al-Saffar, H. Tao, and M. A. Talab, "Review of deep convolution neural network in image classification," in *2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2017, pp. 26–31, doi: 10.1109/ICRAMET.2017.8253139 .

[26] H. H. Sultan, N. M. Salem, and W. Al-Atabany, "Multi-classification of brain tumor images using deep neural network," *IEEE Access*, vol. 7, pp. 69215–69225, 2019, doi: 10.1109/ACCESS.2019.2919122.

[27] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding batch normalization," in *Advances in*

*Neural Information Processing Systems*, 2018, pp. 7694–7705.
[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv Prepr. arXiv1502.03167*, 2015.
[29] L. Huang, D. Yang, B. Lang, and J. Deng, "Decorrelated batch normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 791–800.
[30] J. S. Paul, A. J. Plassard, B. A. Landman, and D. Fabbri, "Deep learning for brain tumor classification," in *Medical Imaging 2017: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 10137, 2017.
[31] I. Ullah and H. J. Lee, "Moving vehicle detection and information extraction based on deep neural network," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, 2017, pp. 102–107.
[32] N. Jmour, S. Zayen, and A. Abdelkrim, "Convolutional neural networks for image classification," in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, 2018, pp. 397–402.
[33] S. Park and N. Kwak, "Analysis on the dropout effect in convolutional neural networks," in *Asian conference on computer vision*, 2016, pp. 189–204, doi: 10.1007/978-3-319-54184-6_12.
[34] A. Achille and S. Soatto, "Information dropout: Learning optimal representations through noisy computation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2897–2905, 2018.
[35] F. Lei, X. Liu, Q. Dai, H. Zhao, L. Wang, and R. Zhou, "A Multi-view Images Classification Based on Shallow Convolutional Neural Network," in *International Conference on Brain Inspired Cognitive Systems*, 2019, pp. 23–33, doi: 10.1007/978-3-030-39431-8_3.
[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[37] F. Siddique, S. Sakib, and M. A. B. Siddique, "Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers," in *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, 2019, pp. 541–546, doi: 10.1109/ICAEE48663.2019.8975496.
[38] S. Sharma, "Activation functions in neural networks," *Towar. Data Sci.*, vol. 6, 2017.
[39] S. Iqbal, M. U. Ghani, T. Saba, and A. Rehman, "Brain tumor segmentation in multi-spectral MRI using convolutional neural networks (CNN)," *Microsc. Res. Tech.*, vol. 81, no. 4, pp. 419–427, 2018, doi: 10.1002/jemt.22994.
[40] L. Metz, N. Maheswaranathan, J. Nixon, D. Freeman, and J. Sohl-Dickstein, "Understanding and correcting pathologies in the training of learned optimizers," in *International Conference on Machine Learning*, 2019, pp. 4556–4565.
[41] S. Bera and V. K. Shrivastava, "Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification," *Int. J. Remote Sens.*, vol. 41, no. 7, pp. 2664–2683, 2020, doi: 10.1080/01431161.2019.1694725.
[42] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, "Adam optimization algorithm for wide and deep neural network," *Knowl. Eng. Data Sci.*, vol. 2, no. 1, pp. 41–46, 2019, doi: 10.17977/um018v2i12019p41-46.
[43] A. Kemal and S. Kiliçarslan, "Performance analysis of optimization algorithms on stacked autoencoder," in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2019, pp. 1–4, doi: 10.1109/ISMSIT.2019.8932880.
[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014.