Contents lists available at *Dergipark*

Journal of Scientific Reports-A

journal homepage: *https://dergipark.org.tr/tr/pub/jsr-a*

E-ISSN: 2687-6167 ***Number 58, September 2024***

RESEARCH ARTICLE

Receive Date: 26.06.2024 *Accepted Date: 04.09.2024*

Enhanced license plate recognition using deep learning and block-based approach

Gülistan ARSLAN[a]*, Fırat AYDEMİR[b], Seyfullah ARSLAN[c]

[a]*Kütahya Dumlupınar University, Faculty of Engineering, Computer Engineering Dept., 43100, Kütahya, Türkiye*
ORCID: 0000-0001-6498-1635
[b]*Kütahya Dumlupınar University, Faculty of Engineering, Computer Engineering Dept., 43100, Kütahya, Türkiye*
ORCID: 0000-0002-8965-1429
[c]*Kütahya Dumlupınar University, Faculty of Engineering, Computer Engineering Dept., 43100, Kütahya, Türkiye*
ORCID: 0000-0002-2573-273X

Abstract

This study investigates the effectiveness of current deep learning techniques in license plate detection and makes essential contributions. Instead of classifying the characters on Turkish license plates with a single classifier, the characters are divided into blocks of numbers and letters using various image processing techniques, and a separate classifier is used for each block. The proposed approach improves character classification accuracy and license plate recognition accuracy. This approach eliminated the possibility of misclassifying similar letters and numbers and improved the character classification accuracy from 95.9% to 99.6%. In addition, a new character feature dataset was created, and a deep learning model was trained on this dataset. Integrating this model into the system increased the classification accuracy to 99.7%. The YOLOv8 object detection model, trained using CUDA technology, achieved a mAP of 98.9%. The overall accuracy of the whole system in license plate recognition reached 97.3%. This study proves the effectiveness of current deep learning methods and the proposed block-based character recognition approach in license plate recognition.

Keywords: License plate recognition; image processing; convolutional neural networks; object detection; deep learning

* Corresponding author. Tel.: +90-274-443-4258
E-mail address: gulistan.colak@dpu.edu.tr

1. Introduction

Vehicle license plates serve as a unique identification, critical for traffic management and safety in modern societies [1]. Due to their uniqueness, license plates are used as a fundamental tool for registering, tracking, and, where necessary, identification of vehicles. License plates are important in many areas, such as traffic control, security investigations, crime-fighting, and commercial activities. Identifying vehicles that violate traffic rules, identifying drivers exceeding speed limits, optimizing traffic flow, or controlling entry and exit to and from public spaces requires identifying the vehicles' license plates. In this way, traffic accidents can be reduced, traffic congestion can be managed, and overall traffic efficiency can be increased. Since manual identification of vehicle license plates is often time-consuming, error-prone, and inefficient, technological tools such as automatic license plate recognition systems help detect and prevent security threats by instantly recognizing license plate numbers [2]. However, license plate recognition systems that work with traditional image processing methods are sensitive to the adverse effects of weather conditions. In particular, weather conditions such as heavy rain, snow, and fog can reduce license plate visibility and recognition accuracy.

Thus, the high-accuracy license plate recognition system should detect and record the license plates of vehicles. AI-based vehicle license plate recognition systems can help achieve these goals toward efficient management of traffic and security operations. Various image processing techniques and machine learning algorithms are used to detect vehicle license plates in license plate recognition systems. With the recent developments in deep learning, one of the machine learning methods, deep learning-based systems are frequently used in areas such as computer vision, voice recognition, and natural language processing [3]. Deep learning algorithms and image processing techniques have been used to develop more accurate and reliable systems for license plate recognition. [4]. These systems include various techniques such as object detection algorithms, multiple morphological operations, and optical character recognition.

This research aims to design a license plate recognition system that achieves a higher level of accuracy and reliability than the existing works in the literature with the YOLOv8 object detection algorithm [5], [6], which is the latest version of the YOLO [7], [8] and aims to improve deep learning techniques such as CNN-based classifiers for the character classification task.

The main contributions of this study are as follows:

- Instead of using a single classifier for all the characters in the license plate, the license plate image is divided into blocks of numbers and letters, and by using a separate classifier for each block, the character classification accuracy is improved from 95.9% to 99.6% on average for two classes.
- An mAP of 98.9% was achieved with the trained YOLOv8 object detection model.
- The overall license plate recognition accuracy of the system is 97.3%.
- A new character image dataset called License Plate Character Dataset was created for this study.
- To improve the classification accuracy of similar characters, a new dataset containing some pixel-based features of the characters called Pixel-based Character Feature Dataset was created.
- A deep learning model was trained using the dataset containing these pixel-based features and the probability values obtained from this deep learning model were added to the prediction results during testing. With the contribution of this deep learning model, the average classification accuracy was increased to 99.7%.
- The results show that when the classifier based on pixel features alone is used without the CNN classifier, it works with a classification accuracy as high as 90%.

2. Literature review

License plate recognition systems are widely used today, especially in security, traffic management, and automation. The recent development of these systems has been accelerated by the advancement of machine learning and artificial intelligence technologies [9]. This section presents a literature review on machine learning and deep learning methods used in automatic license plate detection applications.

In license plate recognition systems, one of the first and most essential operations is to locate the license plate in the image. License plates usually have fixed shapes and similar colors [10], and a study using these visual features presents a modified version of the GrabCut algorithm to locate the license plate region [11]. To implement this algorithm, the geometric information of the aspect ratios of the license plates was used, and using a dataset of 500 different vehicle images, 99.8% accuracy was achieved in predicting the location of license plates. However, this method seems quite slow compared to current deep learning methods, with an average license plate detection time of 0.21 seconds. Studies conducted with current deep learning methods have achieved much faster detection times, such as 18.75 ms [12]. Apart from these algorithms, recent deep learning-based object detection algorithms such as YOLO, SSD, and Faster-RCNN have been widely used in license plate detection applications in recent years [13], [14], [15]. These deep learning-based algorithms estimate the coordinate points where the object to be detected is located in the image and the class to which it belongs.

Another essential process for license plate recognition systems is the segmentation and classification of the characters in the localized license plate image. In the algorithms used in license plate recognition systems, the characters in the license plate are first segmented separately. Then, the segmented character images are classified using classifiers such as CNN, and character recognition is performed [16], [17].

In the literature, a Chinese license plate recognition research paper was conducted with license plate region detection using SVM and character classification based on CNN networks. The results show that the detection rate of the license plate region was 99%, and character recognition accuracy was 97.1%. The overall accuracy rate of the system was 95% [18]. In another research work, edge detection and SVM algorithms were used. This method achieved 96% accuracy in license plate recognition [19]. YOLOv5 was used for the detection of license plates in another study, and CNN was used for the recognition of characters in the detected license plates. The overall accuracy rate of this method was 92.8% [20]. In another study using the YOLO object detection algorithm, an average recognition rate of 96.9% was achieved on eight datasets from five different regions [21]. In this work, the Fast-YOLOv2 model was used to detect license plates and classify them according to regions, and the CR-NET [22] model was used to recognize the characters in the detected license plates. Fast-YOLO architecture with the CR-NET model is used in another work for the detection and segmentation of license plates[23]. In a study combining Xception and LSTM networks for license plate recognition, the system achieved an overall accuracy rate of 90.5% [24]. A research work proposed to classify license plates and characters simultaneously with two fully convolutional one-stage object detectors and ResNet50 was used as the backbone network. In this study, 96.4% license plate recognition accuracy was obtained [25]. In a paper using the YOLOv5 algorithm, which is stated to be suitable for real-time use, the license plate recognition accuracy was 93% [26]. Another work used YOLOv3 for license plate detection and the ILPRNET algorithm for license plate recognition and achieved an overall accuracy of 93.6% [27]. A study that used the YOLOv3 algorithm both for license plate detection and character detection achieved a 95.05% accuracy rate in license plate recognition [28]. The newest research on license plate recognition very often applies deep learning methods, especially various versions of the YOLO algorithm [29].

In this study, the last version of YOLO architecture, YOLOv8, is applied to the image to detect the coordinates of license plates in the image. Then characters in the obtained license plate images were segmented and classified using the CNN model.

3. Material and method

Information about datasets and deep learning methods used in the study is given in this section. It also provides the hyperparameters for the trained YOLOv8 and classifier models, evaluation metrics applied during the evaluation of the models, and details of the improvements made.

3.1. Dataset

The YOLOv8 object detection model for license plate detection is trained on a dataset containing 24242 marked license plate images [30]. This dataset includes vehicle images and YOLOv8 annotation files containing the coordinates of the license plates in the image. The dataset consists of 21174 training, 2048 validation, and 1020 test data. These values have remained the same as determined by the researchers who provided the dataset. This dataset is referred to as the License Plate Detection Dataset in this study.

After detecting license plates with the YOLOv8 model, locating the characters on the license plates and determining which character they are is necessary. This classification problem requires another data set consisting of character images. The fonts of the characters on the license plates vary around the world. [31]. Using any other license plate character dataset in the world during the training phase may affect the feature extraction and classification success of Turkish license plate characters. For this reason, there is a need for a dataset containing the license plate characters in Türkiye. Since a dataset containing the license plate characters in Türkiye is needed, this dataset is not taken as ready-made but created by the authors. The characters in the images of the license plates in Türkiye are cropped, grayscaled, and resized to 22x44 dimensions, which are the input dimensions of the classifiers. Finally, all the images were labeled to create a license plate character dataset, and the created dataset is named as License Plate Character Dataset. Sample images from the License Plate Character Dataset are shown in Fig. 1.

The License Plate Character Dataset consists of 16500 images, 500 images for each character. There are 32 character image samples in Fig. 1. Since the same character is used for the number 0 and the letter O in Turkish license plates, the images of this character are used for both the number 0 and the letter O. In this way, the dataset contains a total of 16500 images. Since the proposed method classifies characters as numbers and letters, this dataset is divided into two different sub-datasets: number and letter datasets. The sub-dataset consisting of number images is referred to as License Plate Character Dataset-N, and the sub-dataset consisting of letter images is referred to as License Plate Character Dataset-L. The total number of data and the number of training, test, and validation data of these datasets are shown in Table 1.

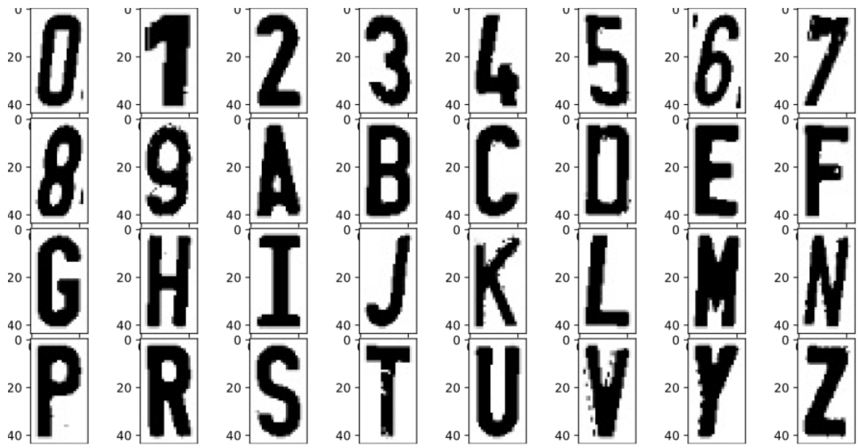


Fig. 1. Sample character images from the License Plate Character Dataset.

In addition to the License Plate Character Dataset, another dataset created by the authors, the Pixel-based Character Feature Dataset contains the data obtained from the pixel-based features of the character images. Eight pixel-based features were extracted for each character image in the License Plate Character Dataset and a new dataset of 16500 rows and nine columns -one for label values- was created.

Data samples from the Pixel-based Character Feature Dataset, are shown in Fig. 2, and the correlation matrix of the data is presented in Fig. 3. Similar to the License Plate Character Dataset, this dataset is divided into number and letter subsets. The sub-dataset consisting of features of number images is called Pixel-based Character Feature Dataset-N and the sub-dataset consisting of features of letter images is called Pixel-based Character Feature Dataset -L.

	Upper Ratio	Lower Ratio	Left Ratio	Right Ratio	Top Left Ratio	Top Right Ratio	Bottom Left Ratio	Bottom Right Ratio	Label
0	0.38	0.62	0.47	0.53	0.17	0.21	0.30	0.32	0
1	0.40	0.60	0.29	0.71	0.04	0.36	0.25	0.35	0
2	0.42	0.58	0.38	0.62	0.09	0.33	0.28	0.29	0
3	0.41	0.59	0.56	0.44	0.24	0.17	0.31	0.27	0
4	0.42	0.58	0.42	0.58	0.14	0.28	0.28	0.30	0
...	...	...	...	...	...	...	...	...	...
9195	0.50	0.50	0.50	0.50	0.19	0.31	0.31	0.18	22
9196	0.52	0.48	0.49	0.51	0.15	0.37	0.34	0.14	22
9197	0.52	0.48	0.52	0.48	0.18	0.34	0.35	0.14	22
9198	0.50	0.50	0.51	0.49	0.20	0.30	0.31	0.19	22
9199	0.51	0.49	0.51	0.49	0.19	0.32	0.32	0.17	22

9200 rows × 9 columns

Fig. 2. Sample of the Pixel-based Character Feature Dataset.

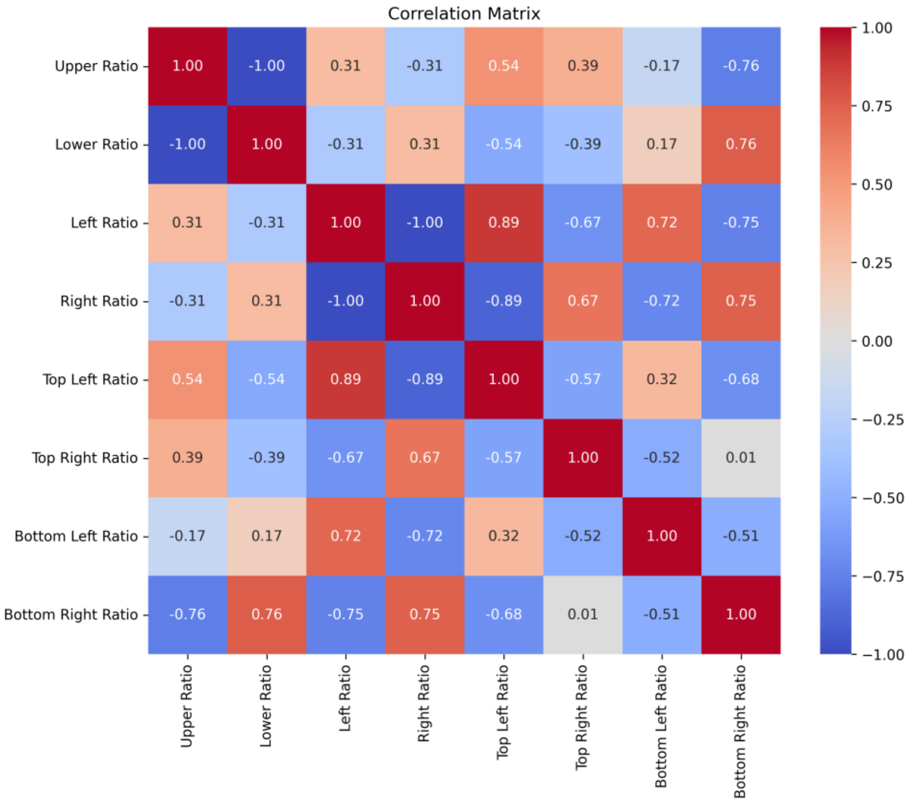


Fig. 3. Correlation matrix of the Pixel-based Character Feature Dataset.

Table 1, presented below, contains information about all the datasets used in this study. The datasets used for character classification are divided into two different sub-datasets, numbers and letters, in accordance with the method proposed in the paper. The proportions of the datasets for training, testing, and validation sub-sets can also be seen in the table.

Table 1. Summary of the datasets used.

Dataset	Total number of data	Training data	Validation data	Test data
License Plate Detection Dataset [30]	24242 (%100)	21174 (%87.3)	2048 (%8.5)	1020 (%4.2)
License Plate Character Dataset-N	5000 (%100)	3500 (%70)	500 (%10)	1000 (%20)
License Plate Character Dataset-L	11500 (%100)	8050 (%70)	1150 (%10)	2300 (%20)
Pixel-based Character Feature Dataset-N	5000 (%100)	3500 (%70)	500 (%10)	1000 (%20)

Pixel-based Character Feature Dataset-L	11500 (%100)	8050 (%70)	1150 (%10)	2300 (%20)

3.2. Proposed model

This section presents the proposed license plate recognition architecture design, the details of the deep learning models used, and the improvements made. To summarize the steps of the license plate recognition process, first, the coordinates of the license plate in the camera image are determined. The license plate region within these coordinates is cropped from the image, and the license plate recognition process is completed due to operations such as character segmentation, cropping, and classification of characters from the image. The model proposed in this study uses a block structure to improve classification accuracy. The structure of the proposed model is shown in Fig. 4.

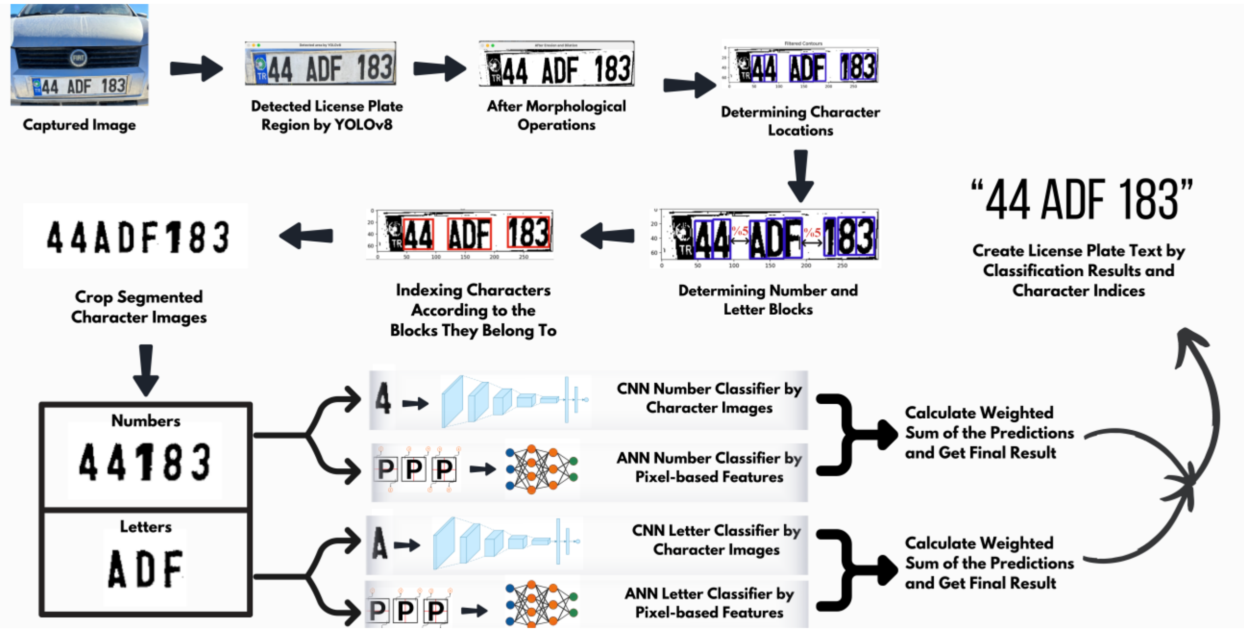


Fig. 4. Proposed model.

3.2.1 License plate detection

The YOLOv8 object detection model is trained to detect license plates in the image. The general structure of the YOLOv8 architecture is presented in Fig. 5.

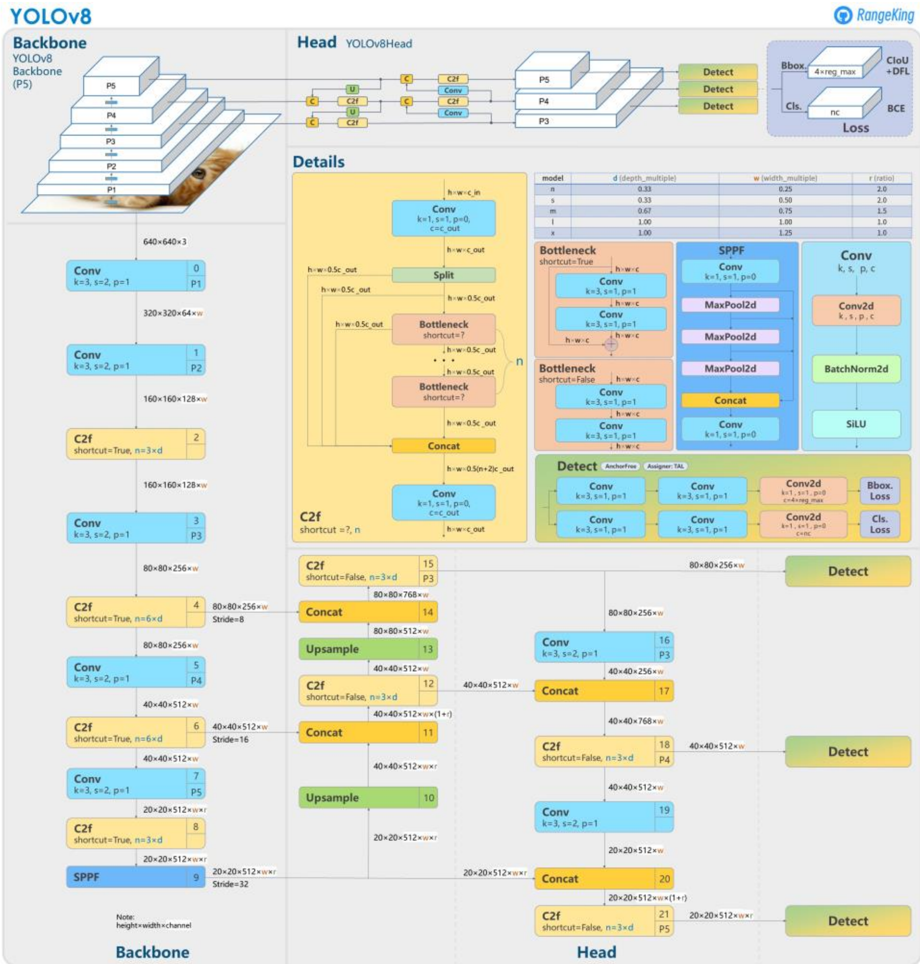


Fig. 5. YOLOv8 architecture [32].

The training was performed on an RTX 3060 graphics card with 3584 CUDA cores using CUDA [33] technology, and the training of the YOLOv8 model took 8 hours and 10 minutes for 21174 images. The parameters of the trained model are presented in Table 2. The trained model was tested on multiple computers. An example of the license plate region detected with the trained YOLOv8 model is shown in Fig. 6.

Table 2. YOLOv8 training hyperparameters.

Parameter	Value
Epoch	100
Batch	8
Optimizer	SGD
Learning rate	0.01

Momentum	0.937
Image size	640x640

Fig. 6. Example of plate region detected with YOLOv8 model.

3.2.2 Character segmentation

After the license plate region is detected and cropped, morphological operations are performed to facilitate the segmentation of the characters on the license plate. First, the image is resized to 300x75. Then, OTSU thresholding [34], erosion with a 3x3 kernel, and dilation with a 3x3 kernel are applied. Erosion and dilation are applied one after the other to reduce noise, emphasize details, and facilitate contour processing. The plate image after the operations is shown in Fig. 7.

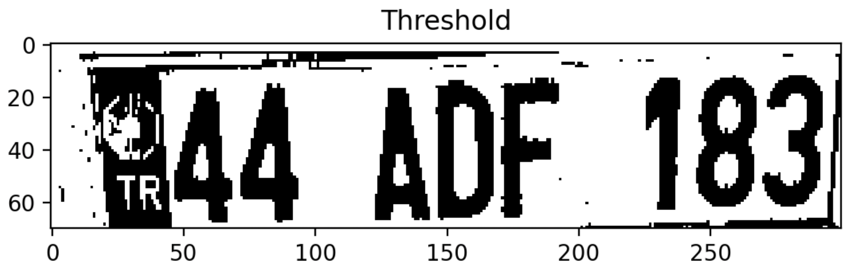


Fig. 7. License plate image after morphological operations.

To locate the characters in the image after thresholding and noise reduction, the contours in the image were found. First, all contours in the image were found with the help of the OpenCV [35] library, and then the contours that may contain characters were filtered. This filtering process was done according to the ratio of the found contour sizes to the total size of the license plate. Upper and lower bounds were set for the height and width of the contours that might contain characters, and then the contours outside these bounds were ignored. The upper and lower bounds were determined after tests with the obtained license plate images. Information about these values can be seen in Table 3.

Table 3. Lower and upper bound values for character contour filtering.

Boundary value type	Boundary value
Width lower bound	w*0.166
Width upper bound	w*0.5
Height lower bound	h*0.1
Height upper bound	h*0.66

In the table, w is the total width of the license plate image, and h is the total height of the license plate image. After these operations, the license plate image contains the filtered contours and shows only the position of the characters that need to be recognized on the license plate. The contour information before and after the filtering process is shown in Fig. 8.

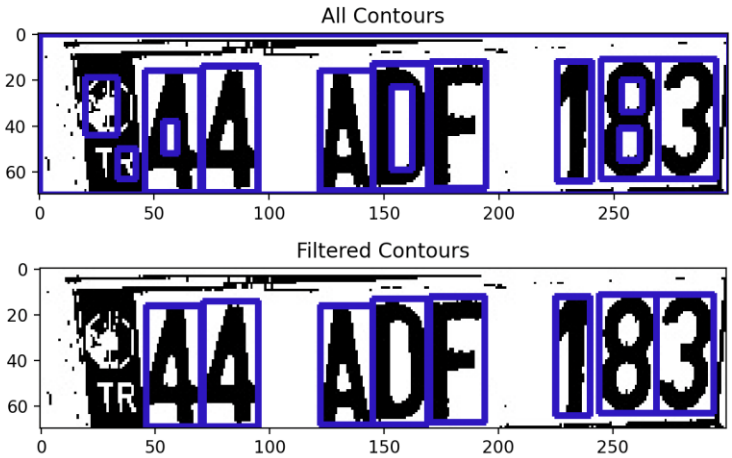


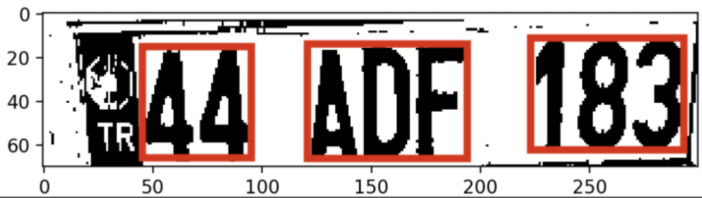
Fig. 8. Contours before and after filtering.

3.2.3 Determination of the blocks

After the positions of the characters are determined, the classification process is started. However, when a single classifier is trained for all characters in the License Plate Character Dataset, the accuracy rate is found to be insufficient, and after the tests, it was determined that similar characters such as 0 and O, 4 and A, and 1 and I affect the recognition accuracy.

In this study, a method for recognizing the letters and numbers on license plates with separate classifiers was developed to improve recognition accuracy. The license plates in Türkiye consist of 3 blocks in total: the first block is a number block indicating the province to which the license plate belongs, the second block is a letter block consisting of 1 to 3 letters, and the last block is a number block again.

With the help of the determined contours, the distance between the characters in the license plate was calculated, and the relevant blocks were determined. As a result of the tests, it was seen that the distance between the blocks was at least 5% of the total width of the plate. Therefore, in cases where the distance between contours is more than 5%, a block change is assumed, and the character images in each block are classified with the help of the relevant classifier. The positions of the blocks on the license plate image and the console outputs of the block detection process are shown in Fig. 9.




```
Total Width: 248 px
From character 0 to character 1 distance: 1 px -- Distance/Total Width Ratio: 0.004
From character 1 to character 2 distance: 27 px -- Distance/Total Width Ratio: 0.109
Block is Changed
From character 2 to character 3 distance: 0 px -- Distance/Total Width Ratio: 0.000
From character 3 to character 4 distance: 1 px -- Distance/Total Width Ratio: 0.004
From character 4 to character 5 distance: 31 px -- Distance/Total Width Ratio: 0.125
Block is Changed
From character 5 to character 6 distance: 4 px -- Distance/Total Width Ratio: 0.016
From character 6 to character 7 distance: 0 px -- Distance/Total Width Ratio: 0.000
Number Indices: [0, 1, 5, 6, 7]
Letter Indices: [2, 3, 4]
```

Fig. 9. The process of identifying plate blocks.

3.2.4 Character classification by images

Character recognition is started after determining which classifier the characters should be sent to according to their indexes. A CNN model was created to classify the number and letter images for character recognition. The architecture of the CNN model used for letter classification is shown in Fig. 10.

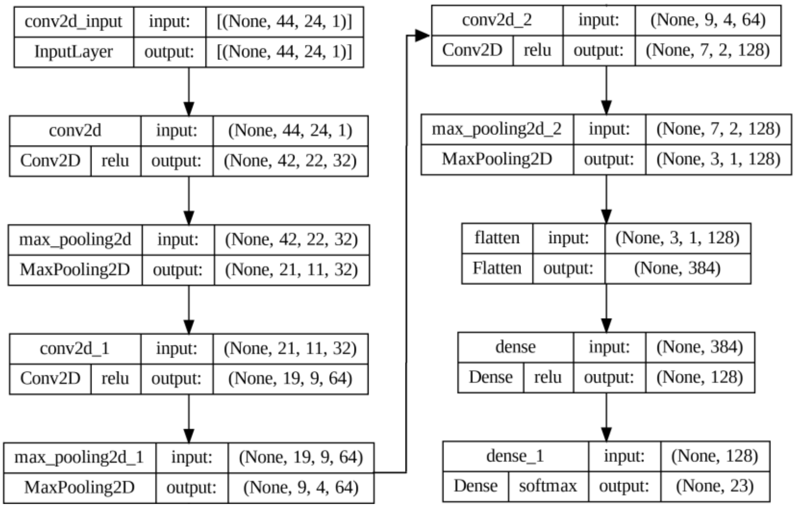


Fig. 10. CNN architecture used for letter classification.

The CNN architecture given above was created for letter classification with a 23-dimensional output layer. The model created for the classification of numbers has the same parts up to the output layer, which is 10-dimensional. The training and data augmentation parameters of the trained models are presented in Table 4.

Table 4. Training and data augmentation parameters of character classifier models.

Parameter	Value
Epoch	10
Batch	16
Optimizer	Adam
Learning rate	0.001
Image size	44x24
Rescale	1/255
Shear range	0.2
Zoom range	0.2
Horizontal flip	True
Loss	Categorical Crossentropy

3.2.5 Character classification by pixel-based features

The recognition process is completed after the detection of the license plate, identification of the blocks in the license plate, and classification of the characters. However, to improve the recognition accuracy, in addition to the block structure, a dataset based on pixel-based features of the images called Pixel-based Character Feature Dataset was created, and an ANN model was trained to classify them according to this dataset. Fig. 11 shows the selection of 8 pixel-based features extracted from the image.

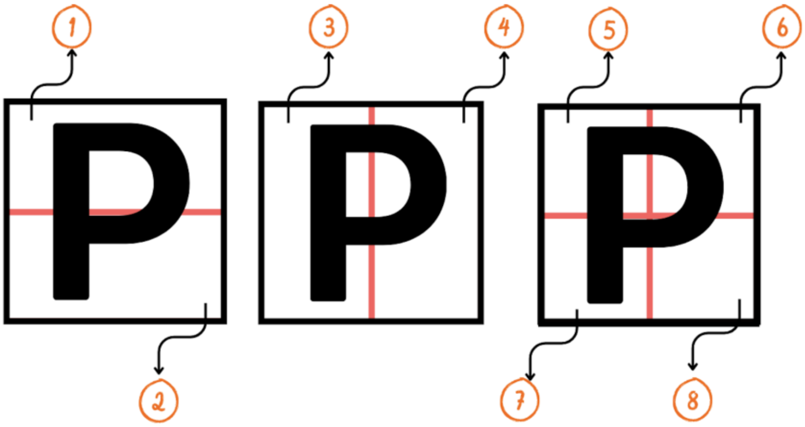


Fig. 11. Selected areas of pixel-based features for character images.

The Pixel-based Character Feature Dataset was created according to the above regions. The character image was first divided horizontally, and the ratio of black pixels in Region 1 to all black pixels in the image was calculated; the same was done for the black pixels in Region 2. Then, the character image was divided vertically, and the ratios of black pixels in Region 3 and Region 4 were calculated. Finally, the image was divided into four equal regions, and the proportion of black pixels in each region was calculated and added to the dataset. The main purpose of dividing the image into 4 regions is to facilitate the discrimination of characters such as the character "P". The pixel ratios may be equal in horizontal or vertical division, but when divided into four regions, it is seen that Region 8 contains fewer black pixels than all other regions. Fig. 12 shows the feature data for the character "P".

	Upper Ratio	Lower Ratio	Left Ratio	Right Ratio	Top Left Ratio	Top Right Ratio	Bottom Left Ratio	Bottom Right Ratio	Label
6277	0.65	0.35	0.62	0.38	0.32	0.33	0.30	0.05	15
6278	0.68	0.32	0.62	0.38	0.35	0.33	0.27	0.05	15
6279	0.66	0.34	0.63	0.37	0.33	0.33	0.31	0.04	15
6280	0.66	0.34	0.63	0.37	0.33	0.33	0.31	0.04	15
6281	0.69	0.31	0.65	0.35	0.36	0.33	0.29	0.02	15
6282	0.56	0.44	0.60	0.40	0.28	0.28	0.32	0.12	15

Fig. 12. Pixel-based feature data for character P.

As expected in the generated dataset, the lowest value is observed in region 8, called the Bottom Right Ratio. In addition, region 1 has more black pixels than region 2, and region 3 has more black pixels than region 4. Using this feature data, the CNN model contributes to the classification of characters such as 0 and D, which are very similar and difficult to classify by the CNN model. The final result is obtained by weighted summing the classification probabilities obtained from the pixel feature data set and the probabilities of the CNN network. In the weighted sum process, the CNN model is given a weight value of 0.9, and the ANN model is given a weight value of 0.1. The neural network's architecture that trained on the Pixel-based Character Feature Dataset is shown in Fig. 13.

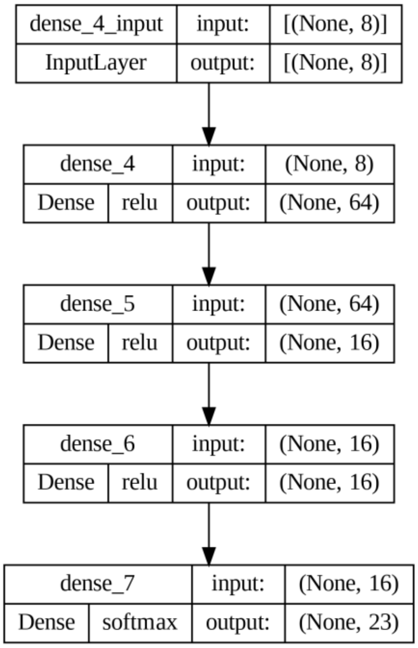


Fig. 13. ANN architecture for character classification based on pixel-based features.

The training parameters of the model used in the classification process are given in Table 5. Evaluation metrics for all models are presented in the next section.

Table 5. Training parameters of the character classification model based on pixel-based features.

Parameter	Value
Epoch	500
Batch	8
Optimizer	Adam
Learning rate	0.001
Loss	Categorical Crossentropy

3.3. Evaluation metrics

This section gives the evaluation metrics of the developed deep learning models. For the YOLOv8 model, Precision, Recall, and Mean Average Precision are some of the critical evaluation criteria. Where P is precision, it denotes the ratio of the TP value of a model to all predicted positive values (TP+FP), hence showing the proportion of samples classified as positive by the model that are positive. Recall R is the model's TP value as a ratio to actual positive values (TP+FN) and determines the number of positive samples that are classed correctly. This means there is a trade-off between these two evaluation metrics. Precision and Recall are calculated respectively as follows [36], [37]:

$$P = \frac{TP}{TP+FP} \quad (1)$$

$$R = \frac{TP}{TP+FN} \quad (2)$$

Average Precision (AP) is the average of the Precision values at different Recall levels, and it indicates the performance of a model at detecting objects at different confidence thresholds. Mean Average Precision (mAP) is the mean of the AP values for all classes, which describes the model's performance. The values of AP and mAP are obtained by [38]:

$$AP = \int_0^1 P(R)\, dR \quad (3)$$

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_c \quad (4)$$

In the given equations $\mathrm{P(R)}$ is the Precision value at a given Recall level, N is the number of classes, and $\mathrm{AP_c}$ is the average Precision value for the class c. For the trained CNN and ANN models, there are several evaluation metrics in addition to Precision and Recall metrics. These are Accuracy (A) and Specificity (S) metrics. Accuracy is the ratio of the sum of the TP and TN values to all samples in the test set and represents the overall correct prediction rate. Specificity is the ratio of the TN value to all true negative values (TN + FP) and measures how accurately truly negative samples are classified. Accuracy and Specificity metrics are calculated as follows respectively [39]:

$$A = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$S = \frac{TN}{TN+FP} \quad (6)$$

Categorical Crossentropy (CCE), used in the classification of CNN and ANN models, is a loss function used when there are two or more label classes and is calculated as follows [40]:

$$CCE = -\sum_{C=1}^{M} y_{o,c} \, log(p_{o,c}) \quad (7)$$

In the specified loss function, M represents the number of classes. y is a binary representation (0 or 1) indicating whether the class label c is the correct classification for observation o, and p represents the probability that observation o is predicted to belong to class c. In the next section, we present and discuss the performance of the deep learning models according to these evaluation metrics.

4. Results and discussion

This section presents the performance of the trained deep learning models according to the evaluation metrics. The complexity matrix obtained after training the YOLOv8 model is shown in Fig. 14. The performance values are presented in Table 6.

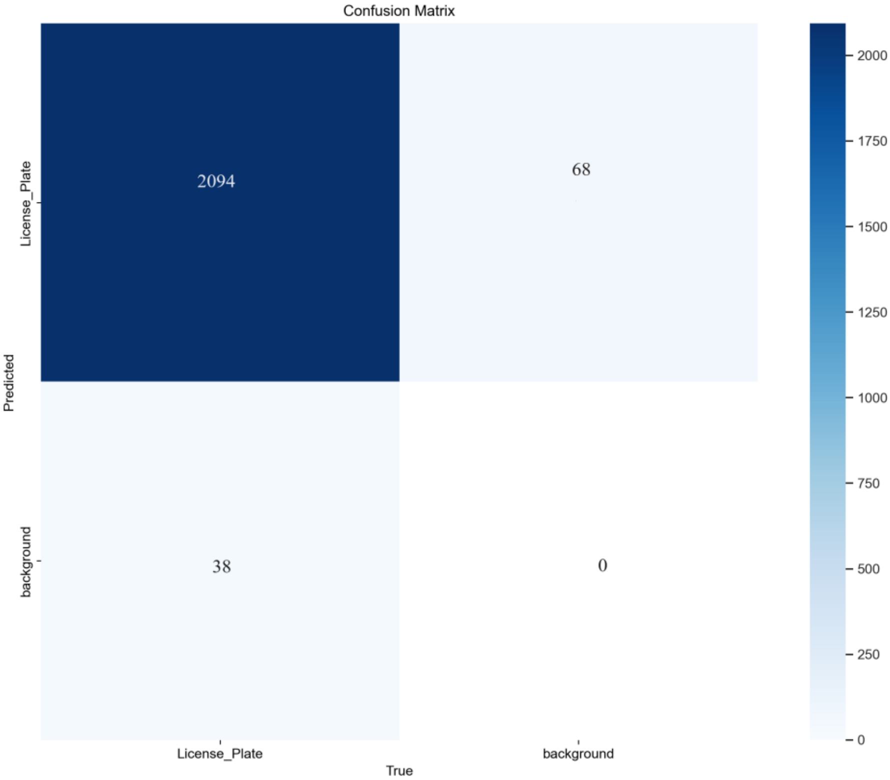


Fig. 14. Confusion matrix of YOLOv8 model.

Table 6. YOLOv8 object detection model test performance values.

Metric	Value
Precision	0.979
Recall	0.973
mAP	0.989

The evaluation results show that the trained model is 98.9% accurate in detecting license plates in the image. To better understand the balance achieved by the model for Precision and Recall metrics, the Precision-Recall curve of the model is presented in Fig. 15.

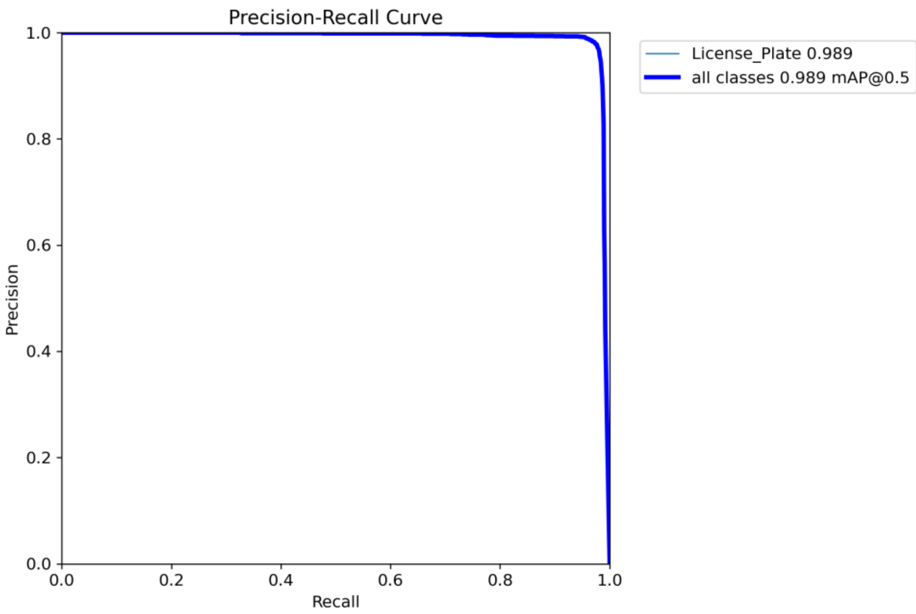


Fig. 15. Precision-Recall curve of YOLOv8 model.

The Precision-Recall curve indicates the trade-off between Precision and Recall for various threshold values. The high area under the curve indicates that both Recall and Precision values are high. Here, high Precision is associated with a low FP rate, and high Recall is associated with a low FN rate. Graphs of the changes in all evaluation metrics during training can be seen in Fig. 16.

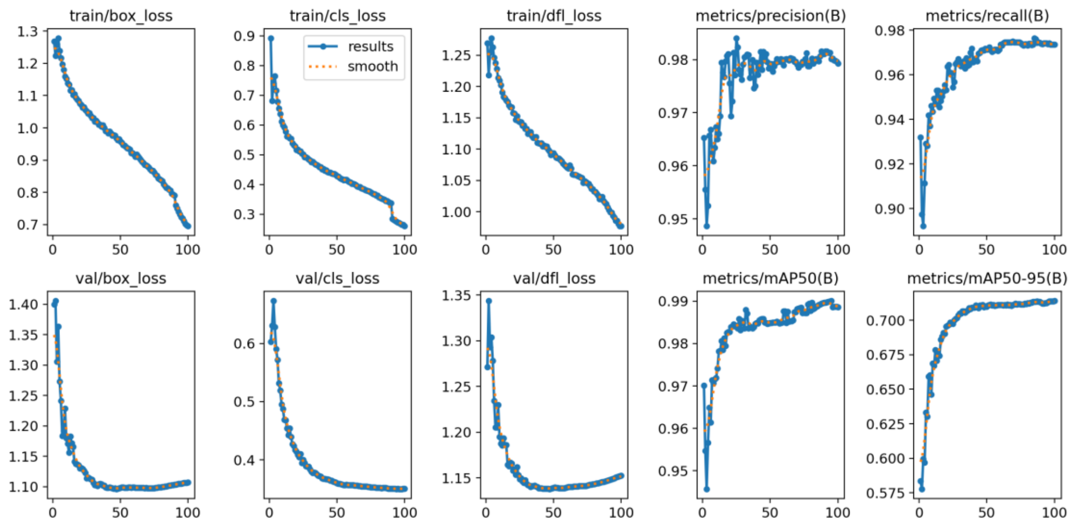


Fig. 16. Training and validation performances of the YOLOv8 model.

When the results are analyzed, it is seen that the loss values in the training and validation phases decrease sharply after the first 20 epochs and the mAP value increases. Considering the data size and the noisy samples in the data, it is seen that the trained model achieves very successful results. After the training of the license plate detection model was completed, the training of the CNN models that classify the number and letter images on the license plate was completed. In order to make a comparison, we first trained a single classifier that classifies all characters with the same architecture. Then, two separate classifiers were trained for letters and numbers, and their performances were analyzed. The accuracy and loss values obtained during the training of the classifier that classifies all characters are given in Fig. 17. The accuracy and loss values obtained during the training of the classifiers that classify numbers and letters separately are shown in Fig. 18.

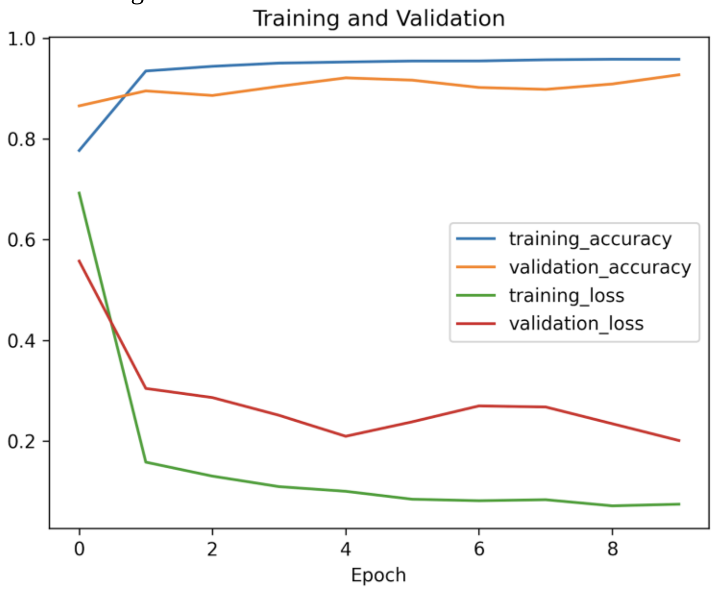
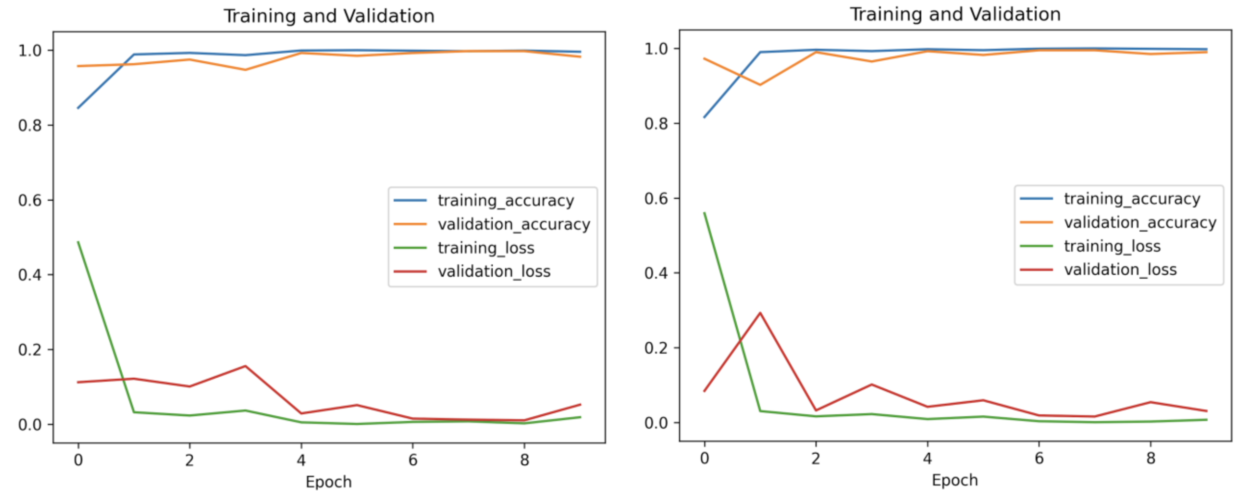


Fig. 17. Training graph of the single CNN model classifying all characters



(a) **(b)**

Fig. 18. Training graphs of the letter classifier CNN model (a) and the number classifier CNN model (b).

When the training graphs are analyzed, it can be seen that using two different classifiers, number and letter classifiers, has higher accuracy and much lower loss values than a single classifier. The complexity matrices obtained from the tests of these classifiers on the test data are given below. Fig. 19 shows the complexity matrix of the single classifier, and Fig. 20 shows the complexity matrices of the proposed two-classifier structure.

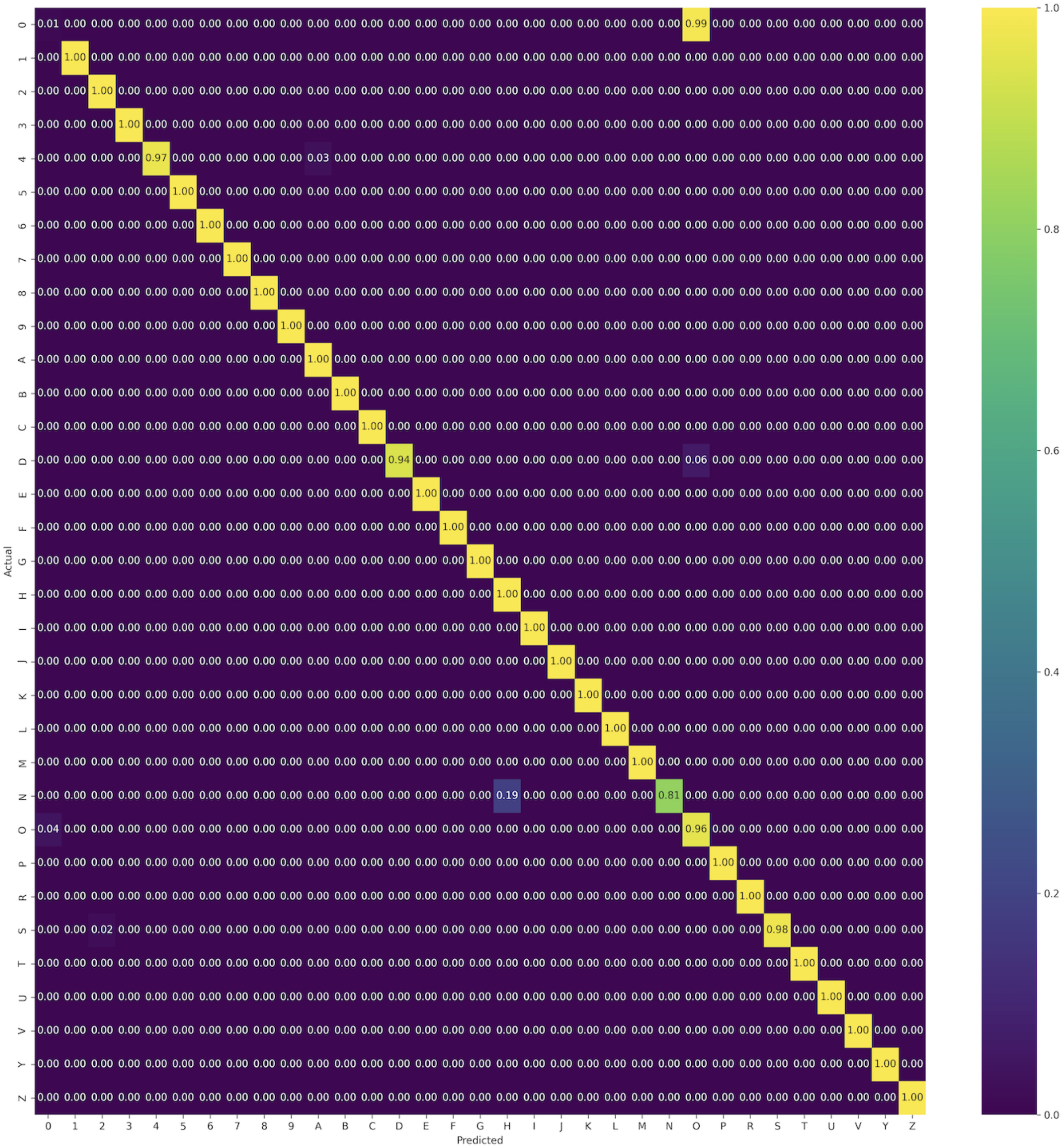
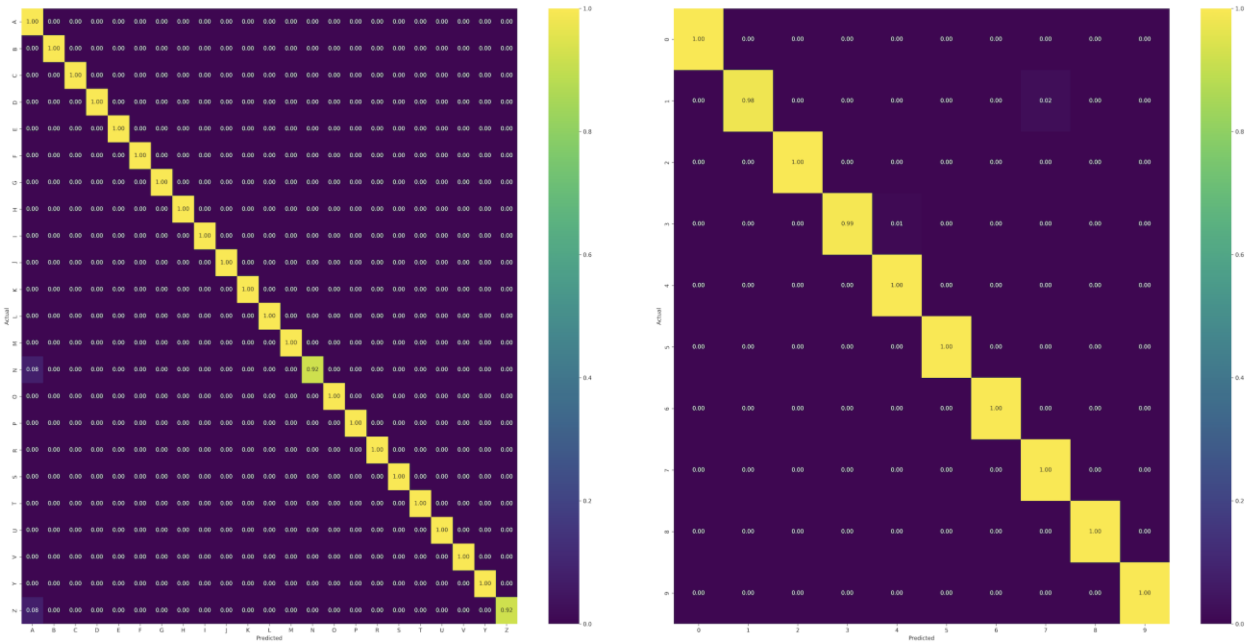


Fig. 19. Complexity matrix of the single CNN model classifying all characters.



(a) (b)

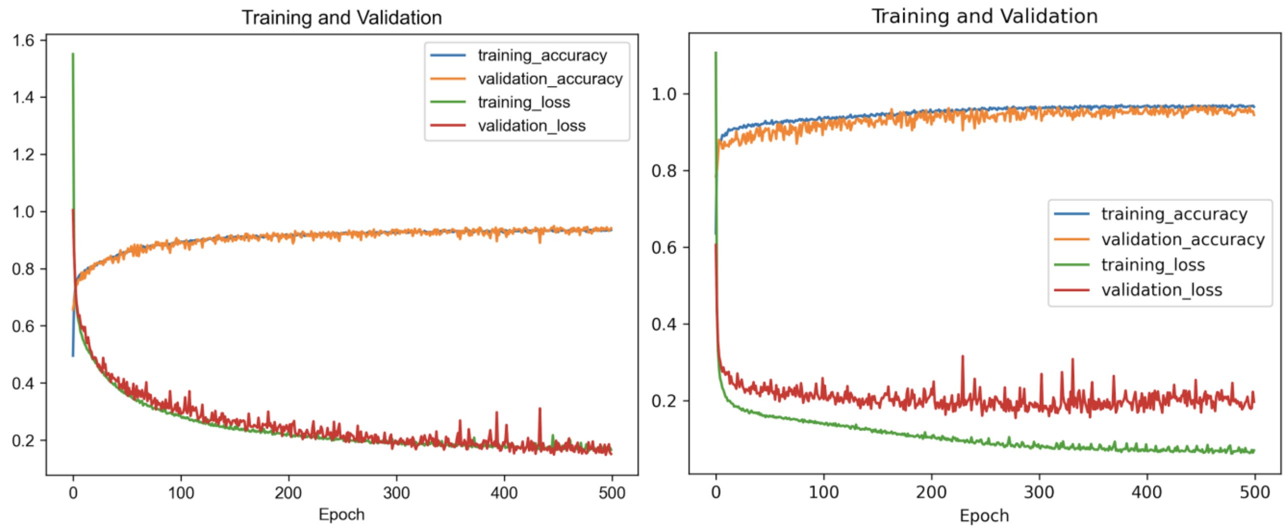
Fig. 20. Complexity matrices of the letter classifier CNN model (a) and the number classifier CNN model (b).

When the complexity matrices are examined, it is seen that the model that classifies all characters classifies almost all of the '0' characters as the letter 'O' and has a high misclassification rate in the classification of 'N' and 'H' characters. It is seen that the models trained separately for letters and numbers make predictions with much higher accuracy. The performance values obtained from the complexity matrix of the models can be seen in Table 7.

Table 7. Test performance of CNN classifiers.

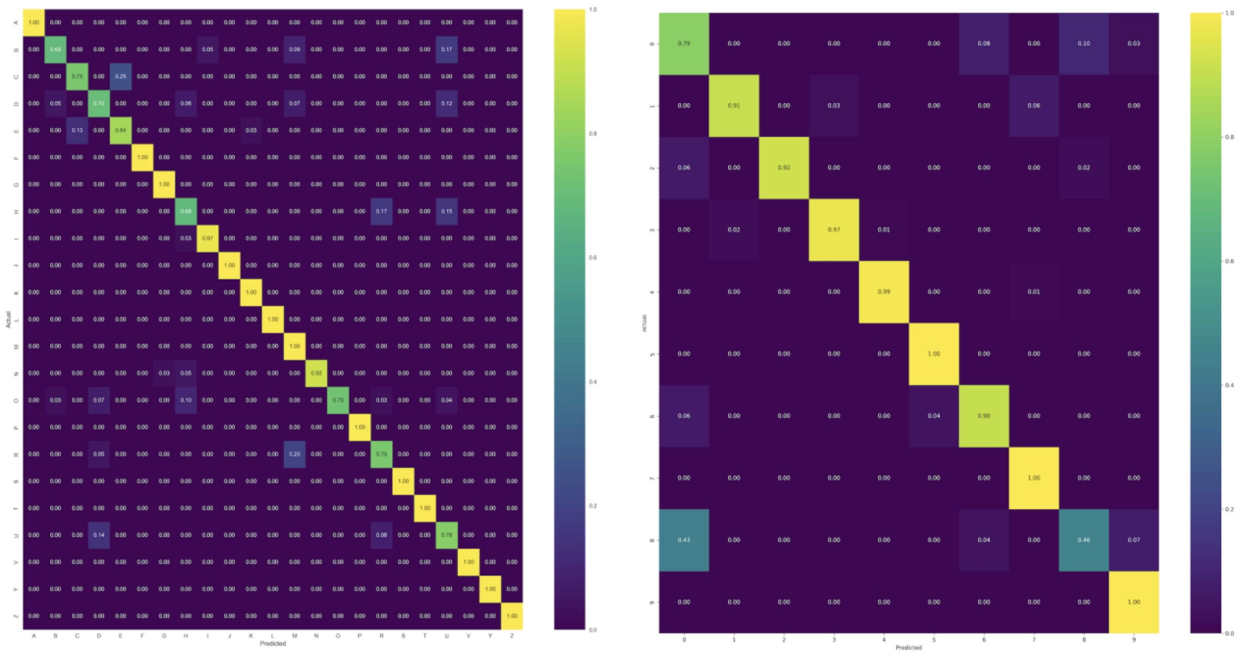
Metric	Precision	Recall	Specificity	Accuracy
All Characters Classifier	0.95	0.96	0.95	0.959
Letter Classifier	0.99	0.99	0.98	0.993
Number Classifier	0.99	0.99	0.99	0.999

When the results are analyzed, it is seen that the proposed method of dividing the license plate into blocks and using different classifiers for each block achieves a very high accuracy rate compared to the traditional method of classifying all characters with a single classifier. In addition to the CNN models, ANN models that classify using pixel-based features were also trained and tested. The training graphs can be seen in Fig. 21, and the complexity matrix values obtained after the test can be seen in Fig. 22.



(a) **(b)**

Fig. 21. Training plots of letter classifier ANN model (a) and number classifier ANN model (b).



(a) **(b)**

Fig. 22. Complexity matrices of the letter classifier ANN model (a) and the number classifier ANN model (b).

When the training graphs and complexity matrices are examined, it is seen that although the accuracy of the ANN model is high and the loss value is low during classification, the loss values are high when the validation data is used. This shows that the model is subject to overfitting; it cannot generalize the data well enough. However, the letter classifier ANN model performed well despite the low data size and limited feature diversity. When the complexity matrices are analyzed, it is seen that there is difficulty in classifying symmetrical characters in both horizontal and vertical axes. For example, it is noticeable that incorrect predictions were made between the characters '0' and '8'. The performance values of the trained ANN models are given in Table 8.

Table 8. Test performance of ANN classifiers.

Metric	Precision	Recall	Specificity	Accuracy
Letter Classifier	0.91	0.90	0.90	0.90
Number Classifier	0.90	0.89	0.89	0.89

Although the accuracy values are relatively low, sufficient classification accuracy is achieved with low computational cost and low data set size using eight simple features extracted from the image. As a result of the weighted sum of the probability values obtained from CNN and ANN models, the accuracy rate in the character classification process was increased from 99.6% to 99.7%.

After these steps, there is one last step left to be done. This is to reveal the license plate text by combining the characters obtained from the classification results of the character images on the license plate. A vehicle image with license plate recognition completed is shown in Fig. 23.

Fig. 23. Image of a vehicle with completed license plate recognition.

As a result of the tests, the system's overall license plate recognition accuracy was found to be 97.3%. Table 9 compares the results obtained with those of other studies in the literature.

Table 9. Comparison of the proposed method with other studies in the literature.

Study	License Plate Detection (mAP)	Character Recognition Accuracy	Overall License Plate Recognition Accuracy
Cheng et al. [18]	**0.990**	0.971	0.950
Khan et al. [20]	0.950	0.928	0.928
Tourani et al. [28]	0.978	0.991	0.950
Zaafouri et al. [41]	0.955	0.994	0.945
Proposed Model	0.989	**0.997**	**0.973**

The comparison shows that the classification process using the proposed block structure outperforms the studies in the literature in terms of license plate recognition accuracy. Although the detection accuracy of license plates is 0.1% higher in only one study, the proposed model outperforms this study in overall license plate recognition accuracy by giving much better results in recognizing the characters in the detected license plates. In addition, the fact that the YOLOv8 model, which is the most up-to-date object detection algorithm in the literature, was efficiently trained using CUDA cores with a large amount of data and achieved a high detection rate significantly affected the accuracy of the license plate recognition system. The approach of classifying the classification process by dividing it into sub-problems instead of using a single model provides solutions with low computational cost due to the use of smaller size models and high accuracy value due to the reduction in complexity.

5. Conclusion

This work contributes to using current deep-learning methods for license plate detection. Among the main contributions, instead of classifying the characters in the license plate with a single character classifier, the license plate image is divided into blocks of numbers and letters so that separate classifiers are used for numbers and separate classifiers for letters.

This approach reduces the probability of misrecognizing similar characters such as 0 and O, 4 and A, 1 and I to zero. Character classification accuracy increased from 95.9% for a single classifier to 99.6% on average using two classifiers. In addition, a new character feature dataset, Pixel-based Character Feature Dataset was created to improve character classification accuracy. The addition of the deep learning model trained using the new dataset containing pixel-based features increased the classification accuracy of the system to 99.7%. Furthermore, a high mAP value of 98.9% was achieved with the trained YOLOv8 object detection model. The overall license plate recognition accuracy of the system was 97.3%. All these findings demonstrate the effectiveness and accuracy of deep learning methods used in license plate detection.

Future research could address issues such as increasing the model's generalization capability by using more data, developing a detailed object detection algorithm to improve small object detection performance, or conducting field studies to improve system performance in conditions that make camera images unfavorable for image processing. Such studies could lead to more comprehensive steps toward improving the reliability and usability of license plate recognition systems.

Acknowledgements

This study is derived from Gülistan Arslan's master's thesis and published as an academic output of a project with project number 2022-14 supported by the Scientific Research Projects Coordination Unit of Kütahya Dumlupınar

University. The sample vehicle license plate shown in the study belongs to the authors, and no ethical violations were made.

References

[1] M. Arora, A. Jain, S. Rustagi, and T. Yadav, "Automatic number plate recognition system using optical character recognition," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 986–992, 2019.
[2] B. Lavanya and G. Lalitha, "Multileveled ALPR using Block-Binary-Pixel-Sum Descriptor and Linear SVC," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 5, 2022.
[3] Y. Kumar, K. Kaur, and G. Singh, "Machine learning aspects and its applications towards different research areas," in *2020 International conference on computation, automation and knowledge management (ICCAKM)*, IEEE, 2020, pp. 150–156.
[4] W. Weihong and T. Jiaoyang, "Research on license plate recognition algorithms based on deep learning in complex environment," *IEEE Access*, vol. 8, pp. 91661–91675, 2020.
[5] S. Rahman, J. H. Rony, J. Uddin, and M. A. Samad, "Real-Time Obstacle Detection with YOLOv8 in a WSN Using UAV Aerial Photography," *J Imaging*, vol. 9, no. 10, p. 216, 2023.
[6] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
[8 C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.
[9] G. Aslı and A. DOĞAN, "License Plate Recognition System Based on Artificial Intelligence with Different Approach," *El-Cezeri*, vol. 10, no. 1, pp. 121–136, 2023.
[10] H. Shi and D. Zhao, "License Plate Recognition System Based on Improved YOLOv5 and GRU," *IEEE Access*, vol. 11, pp. 10429–10439, 2023.
[11] A. O. Salau, T. K. Yesufu, and B. S. Ogundare, "Vehicle plate number localization using a modified GrabCut algorithm," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 4, pp. 399–407, 2021.
[12] T.-A. Pham, "Effective deep neural networks for license plate detection and recognition," *Vis Comput*, vol. 39, no. 3, pp. 927–941, 2023.
[13] N. Duan, J. Cui, L. Liu, and L. Zheng, "An end to end recognition for license plates using convolutional neural networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 2, pp. 177–188, 2019.
[14] N. Awalgaonkar, P. Bartakke, and R. Chaugule, "Automatic license plate recognition system using ssd," in *2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA)*, IEEE, 2021, pp. 394–399.
[15] N. P. Ap, T. Vigneshwaran, M. S. Arappradhan, and R. Madhanraj, "Automatic number plate detection in vehicles using faster R-CNN," in *2020 International conference on system, computation, automation and networking (ICSCAN)*, IEEE, 2020, pp. 1–6.
[16] B. Pu, "Research on Chinese license plate recognition algorithm based on convolution neural network," in *Journal of Physics: Conference Series*, IOP Publishing, 2021, p. 032055.
[17] M. Driss, I. Almomani, R. Al-Suhaimi, and H. Al-Harbi, "Automatic Saudi Arabian License Plate Detection and Recognition Using Deep Convolutional Neural Networks," in *International Conference of Reliable Information and Communication Technology*, Springer, 2021, pp. 3–15.
[18] C. Cheng, L. Mei, and J. Zhang, "License plate recognition via deep convolutional neural network," in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing, 2018, p. 062030.
[19] A. Singh and P. Singh, "License Plate Recognition for Traffic Management," *Journal of Management and Service Science (JMSS)*, vol. 1, no. 2, pp. 1–14, 2021.
[20] I. R. Khan *et al.*, "Automatic License Plate Recognition in Real-World Traffic Videos Captured in Unconstrained Environment by a Mobile Camera," *Electronics (Basel)*, vol. 11, no. 9, p. 1408, 2022.
[21] R. Laroca, L. A. Zanlorensi, G. R. Gonçalves, E. Todt, W. R. Schwartz, and D. Menotti, "An efficient and layout- independent automatic license plate recognition system based on the YOLO detector," *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 483–503, 2021.
[22] S. Montazzolli and C. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in *2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, IEEE, 2017, pp. 55–62.
[23] S. M. Silva and C. R. Jung, "Real-time license plate detection and recognition using deep convolutional neural networks," *J Vis Commun Image Represent*, vol. 71, p. 102773, 2020.
[24] L. Zhang, P. Wang, H. Li, Z. Li, C. Shen, and Y. Zhang, "A Robust Attentional Framework for License Plate Recognition in the Wild," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 6967–6976, Nov. 2021, doi: 10.1109/TITS.2020.3000072.
[25] Q. Huang, Z. Cai, and T. Lan, "A Single Neural Network for Mixed Style License Plate Detection and Recognition," *IEEE Access*, vol. 9, pp. 21777–21785, 2021, doi: 10.1109/ACCESS.2021.3055243.
[26] J. A *et al.*, "Enhancing Public Safety through License Plate Recognition for Counter terrorism through Deep Learning Technique," in *2023 4th International Conference on Signal Processing and Communication (ICSPC)*, IEEE, Mar. 2023, pp. 96–100. doi: 10.1109/ICSPC57692.2023.10125687.
[27] Y. Zou *et al.*, "License plate detection and recognition based on YOLOv3 and ILPRNET," *Signal Image Video Process*, vol. 16, no. 2, pp.

473–480, Mar. 2022, doi: 10.1007/s11760-021-01981-8.
[28] A. Tourani, A. Shahbahrami, S. Soroori, S. Khazaee, and C. Y. Suen, "A robust deep learning approach for automatic iranian vehicle license plate detection and recognition for surveillance systems," *IEEE Access*, vol. 8, pp. 201317–201330, 2020.
[29] M. M. Khan, M. U. Ilyas, I. R. Khan, S. M. Alshomrani, and S. Rahardja, "A review of license plate recognition methods employing neural networks," *IEEE Access*, 2023.
[30] P. Roboflow Universe, "License Plate Recognition Dataset." Accessed: Mar. 02, 2024. [Online]. Available: https://universe.roboflow.com/roboflow-universe-projects/license-plate-recognition-rxg4e
[31] R. K. Varma P, S. Ganta, H. K. B, and P. Svsrk, "A Novel Method for Indian Vehicle Registration Number Plate Detection and Recognition using Image Processing Techniques," *Procedia Comput Sci*, vol. 167, pp. 2623–2633, 2020, doi: 10.1016/j.procs.2020.03.324.
[32] K. Range, "YOLOv8 by RangeKing." Accessed: Aug. 09, 2024. [Online]. Available: https://github.com/RangeKing
[33] R. G. Jha and A. Samlodia, "GPU-acceleration of tensor renormalization with PyTorch using CUDA," *Comput Phys Commun*, vol. 294, p. 108941, 2024.
[34] Z. Y. Tan, S. N. Basah, H. Yazid, and M. J. A. Safar, "Performance analysis of Otsu thresholding for sign language segmentation," *Multimed Tools Appl*, vol. 80, pp. 21499–21520, 2021.
[35] J. Sigut, M. Castro, R. Arnay, and M. Sigut, "OpenCV basics: A mobile application to support the teaching of computer vision concepts," *IEEE Transactions on Education*, vol. 63, no. 4, pp. 328–335, 2020.
[36] S. Dörterler, "Hybridization of k-means and meta-heuristics algorithms for heart disease diagnosis," *New Trends in Engineering and Applied Natural Sciences*, p. 55, 2022.
[37] S. Dörterler, H. Dumlu, D. Özdemir, and H. Temurtaş, "Hybridization of Meta-heuristic Algorithms with K-Means for Clustering Analysis: Case of Medical Datasets," *Gazi Journal of Engineering Sciences*, vol. 10, no. 1, pp. 1–11, Apr. 2024, doi: 10.30855/gmbd.0705N01.
[38] M. Safran, A. Alajmi, and S. Alfarhood, "Efficient Multistage License Plate Detection and Recognition Using YOLOv8 and CNN for Smart Parking Systems," *J Sens*, vol. 2024, 2024.
[39] F. Aydemir and S. Arslan, "A System Design With Deep Learning and IoT to Ensure Education Continuity for Post-COVID," *IEEE Transactions on Consumer Electronics*, 2023.
[40] O. Sahin, "Music Genre Classification Based on Song Titles with Long Short-Term Memory," in *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, IEEE, 2023, pp. 1-5.
[41] A. Zaafouri, M. Sayadi, and W. Wu, "A Vehicle License Plate Detection and Recognition Method Using Log Gabor Features and Convolutional Neural Networks," *Cybern Syst*, vol. 54, no. 1, pp. 88–103, 2023.