IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

Volume 12, Issue 4, April 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.317

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

||Volume 12, Issue 4, April 2023||

|DOI:10.15662/IJAREEIE.2022.1204045 |

Traffic Signal Classification using Deep Learning for Enhanced Autonomous Driving Systems

Prof. Zeba Vishwakarma[1], Prof. Divya Pandey[2], Prof. Mallika Dwivedi[3], Arjita Sarkar[4], Gourav Mourya[5]

Department of Computer Science and Engineering, Baderia Global Institute of Engineering and Management, Jabalpur, MP, India[1, 2, 3,4,5]

ABSTRACT: This paper presents a deep learning-based model for the classification of traffic signals, a crucial component for the development of autonomous driving systems and driver alert mechanisms. Utilizing a convolutional neural network (CNN), the proposed model is designed to recognize 43 distinct traffic signs with high accuracy. The model is trained on a publicly available dataset, achieving an impressive accuracy of over 95% in just 50 epochs. The preprocessing steps, model architecture, and training process are discussed in detail. The model's performance indicates its potential for real-world applications in enhancing the safety and efficiency of self-driving cars. Future work includes hyper parameter tuning and the integration of the model into real-time systems for further optimization.

KEYWORDS: Traffic Signal Classification, Deep Learning, Autonomous Driving, Convolutional Neural Network, Driver Alert Systems, Hyper-parameter Tuning

I. INTRODUCTION

The rapid advancement of technology in the field of autonomous driving has brought forth the need for robust systems capable of navigating complex traffic environments. One of the critical tasks for such systems is the accurate recognition and classification of traffic signals. Traffic signals play a vital role in ensuring road safety and guiding both human drivers and autonomous vehicles. The ability to automatically recognize and interpret these signals can significantly enhance the safety and efficiency of autonomous driving systems and provide timely alerts to drivers.

In this research, we focus on developing a deep learning-based model for traffic signal classification. Leveraging the power of Convolutional Neural Networks (CNNs), our model is designed to identify 43 different types of traffic signs. CNNs have demonstrated exceptional performance in image recognition tasks due to their ability to capture spatial hierarchies in images through convolutional layers.

Methods for image classification utilizing neural networks are extensively documented in current literature [1–6]. Each method presents its own set of strengths and weaknesses, thus ongoing research focuses on developing reliable algorithms. In real traffic scenarios, the interpretation of traffic signs can be hindered by variations in lighting, vibration, and different shooting angles. CNNs have emerged as effective solutions for these challenges [7–11]. They excel in image classification tasks compared to fully connect neural networks due to lower computational requirements and a reduced number of adjustable parameters. Importantly, CNNs demonstrate invariance to variations in shape, rotation, and color intensity of input images.

This study investigates the impact of different filter dimensions within CNNs on classification accuracy and efficiency. Filter dimensions determine the number of features combined to produce a new feature in the output feature map. Smaller filter sizes (e.g., 3 × 3) may merge fewer features, potentially leading to information loss. Conversely, larger filters (e.g., 31 × 31) can integrate more features, potentially including redundant or irrelevant information. The convolutional neural network is trained on the GTSRB dataset [12, 13].

The preprocessing steps involve normalizing the images and splitting the dataset into training and testing sets. Our model consists of multiple convolutional layers followed by pooling layers, dropout layers to prevent overfitting, and dense layers for classification. The use of dropout layers is particularly effective in improving the generalization of the model by randomly dropping units during the training process.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

||Volume 12, Issue 4, April 2023||

|DOI:10.15662/IJAREEIE.2022.1204045 |

The model was trained using the Adam optimizer and sparse categorical cross-entropy loss function. After 50 epochs of training, the model achieved an accuracy of over 95%, demonstrating its effectiveness in recognizing traffic signals. This high level of accuracy indicates the potential of the model to be integrated into real-world applications, where it can contribute to the development of advanced driver-assistance systems (ADAS) and fully autonomous vehicles.

II. PROPOSED METHODOLOGY

II-A. Dataset

The German Traffic Sign Recognition Benchmark (GTSRB) dataset is utilized for training and testing the traffic signal classification model. This dataset consists of 43 different classes of traffic signs with varying levels of complexity and environmental conditions.

II-B. Preprocessing

1. **Loading and Resizing:** Images from the dataset are loaded and resized to a consistent shape suitable for input into the neural network model.
2. **Normalization:** Pixel values of the images are normalized to a range of [0, 1] to facilitate convergence during training.
3. **Train-Test Split:** The dataset is split into training and testing sets with a ratio of 80:20, ensuring that the model is evaluated on unseen data.

II-C. Model Architecture

The traffic signal classification model is constructed using a Convolutional Neural Network (CNN), which is well-suited for image classification tasks due to its ability to learn hierarchical representations directly from pixel data.

1. **Convolutional Layers:** Two sets of convolutional layers are employed:
   a. The first convolutional layer has 64 filters of size (3, 3) with 'relu' activation and padding.
   b. The second convolutional layer has 64 filters of size (3, 3) with 'relu' activation.
2. **Pooling Layers:** After each convolutional layer, a max pooling layer with a pool size of (2, 2) is applied to reduce spatial dimensions and extract dominant features.
3. **Dropout:** Dropout layers with a dropout rate of 0.5 are added after each pooling layer to prevent overfitting by randomly disabling neurons during training.
4. **Flattening:** The output from the final convolutional layer is flattened to a 1-dimensional vector to be fed into densely connected layers.
5. **Dense Layers:** Two dense (fully connected) layers are added:
   a. The first dense layer consists of 128 neurons with 'relu' activation.
   b. The second dense layer has 43 neurons corresponding to the number of traffic sign classes, with 'softmax' activation to output probability scores for each class.

II-D. Training

1. **Compilation:** The model is compiled with 'adam' optimizer, 'sparse_categorical_crossentropy' as the loss function (suitable for integer-encoded labels), and 'accuracy' as the metric to monitor during training.
2. **Training:** The model is trained on the training data for 50 epochs with a batch size of 32. During training, the optimizer adjusts the weights to minimize the loss function, improving the model's ability to classify traffic signs accurately.
3. **Evaluation:** After training, the model is evaluated on the test dataset to assess its generalization performance. Metrics such as accuracy and loss are calculated to measure how well the model performs on unseen data.

III. PSEUDO CODE

i. Load and preprocess the GTSRB dataset

ii. Define CNN model architecture:

```
Initialize Sequential model
Add Conv2D layer with 64 filters, (3, 3) kernel size, 'relu' activation, input shape
Add MaxPooling2D layer with (2, 2) pool size
Add Dropout layer with dropout rate 0.5
Add Conv2D layer with 64 filters, (3, 3) kernel size, 'relu' activation
Add MaxPooling2D layer with (2, 2) pool size
Add Dropout layer with dropout rate 0.5
```

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

||Volume 12, Issue 4, April 2023||

|DOI:10.15662/IJAREEIE.2022.1204045 |

Add Flatten layer to convert 2D matrix to 1D vector
Add Dense layer with 128 neurons and 'relu' activation
Add Dropout layer with dropout rate 0.5
Add Dense layer with 43 neurons (number of classes) and 'softmax' activation

iii. Compile the model:
Use 'adam' optimizer
Use 'sparse_categorical_crossentropy' as loss function
Monitor 'accuracy' metric during training

iv. Train the model:
Fit the model on training data
Use batch size of 32 and train for 50 epochs

v. Evaluate the model:
Evaluate model performance on test data
Calculate accuracy and loss metrics

IV. EXPERIMENTATION RESULTS

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 50, 50, 64)        1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 25, 25, 64)        0
_________________________________________________________________
dropout (Dropout)            (None, 25, 25, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 23, 23, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 11, 11, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 11, 11, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 7744)              0
_________________________________________________________________
dense (Dense)                (None, 128)               991360
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 43)                5547
=================================================================
Total params: 1,035,627
Trainable params: 1,035,627
Non-trainable params: 0
_________________________________________________________________
```

Figure 4.1 Model Summary with architecture overview

The two sets of convolutional layers with pooling and dropout help in extracting hierarchical features from input images, enhancing the model's ability to distinguish between different traffic signs.
The dense layers at the end of the network process the extracted features and make the final classification decision, outputting probabilities for each traffic sign class.
The dropout layers play a crucial role in preventing the model from overfitting by randomly dropping units during training, which improves generalization on unseen data.

Layer 1: Conv2D- This layer applies 64 filters of size (3, 3) to the input images. The 'relu' activation function is used to introduce non-linearity. Padding is 'same', ensuring the output has the same height and width as the input.

Layer 2: MaxPooling2D- A pooling layer with a pool size of (2, 2) reduces the spatial dimensions by taking the maximum value in each 2x2 patch of the feature map.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

||Volume 12, Issue 4, April 2023||

|DOI:10.15662/IJAREEIE.2022.1204045 |

Layer 3: Dropout- Dropout is used to prevent overfitting by randomly dropping 50% of the neurons during training, which helps in generalizing the model.

Layer 4: Conv2D- Another convolutional layer with 64 filters of size (3, 3) and 'relu' activation. No padding ('valid' by default) reduces the feature map size slightly.

Layer 5: MaxPooling2D- MaxPooling with (2, 2) pool size further reduces spatial dimensions, focusing on the most important features.

Layer 6: Dropout- Another dropout layer with a 50% dropout rate to enhance model generalization.

Layer 7: Flatten- Flattens the 3D output to 1D, preparing it for input into the fully connected layers**.**

Layer 8: Dense- A densely connected layer with 128 neurons and 'relu' activation function, facilitating learning of complex representations.

Layer 9: Dropout- Dropout layer with a 50% dropout rate to further prevent overfitting.

Layer 10: Dense- Final dense layer with 43 neurons (corresponding to the number of traffic sign classes) and 'softmax' activation. This layer outputs probabilities for each class.

Total Model Parameters: Total trainable parameters are 1,035,627 and Non-trainable parameters are 0
This architecture has proven effective in achieving a high accuracy of over 95% on the test dataset, demonstrating its suitability for real-world applications in autonomous driving and driver alert systems.

```
model.fit(x_train, y_train, epochs = 10, batch_size = 128, validation_data = (x_val, y_val)
```

```
Epoch 1/10
246/246 - 184s - loss: 0.4137 - accuracy: 0.8650 - val_loss: 0.1094 - val_accuracy: 0.9783
Epoch 2/10
246/246 - 182s - loss: 0.3698 - accuracy: 0.8785 - val_loss: 0.0961 - val_accuracy: 0.9802
Epoch 3/10
246/246 - 182s - loss: 0.3428 - accuracy: 0.8871 - val_loss: 0.0826 - val_accuracy: 0.9832
Epoch 4/10
246/246 - 181s - loss: 0.3144 - accuracy: 0.8966 - val_loss: 0.0774 - val_accuracy: 0.9843
Epoch 5/10
246/246 - 183s - loss: 0.2891 - accuracy: 0.9043 - val_loss: 0.0681 - val_accuracy: 0.9856
Epoch 6/10
246/246 - 181s - loss: 0.2738 - accuracy: 0.9083 - val_loss: 0.0622 - val_accuracy: 0.9874
Epoch 7/10
246/246 - 182s - loss: 0.2535 - accuracy: 0.9163 - val_loss: 0.0671 - val_accuracy: 0.9879
Epoch 8/10
246/246 - 183s - loss: 0.2403 - accuracy: 0.9220 - val_loss: 0.0533 - val_accuracy: 0.9899
Epoch 9/10
246/246 - 181s - loss: 0.2325 - accuracy: 0.9229 - val_loss: 0.0545 - val_accuracy: 0.9902
Epoch 10/10
246/246 - 182s - loss: 0.2222 - accuracy: 0.9277 - val_loss: 0.0532 - val_accuracy: 0.9904
<tensorflow.python.keras.callbacks.History at 0x7f568f9c1400>
```

Figure 4.2 Model Compilation with Training and Validation Metrics

Training Loss (0.2222) indicates the average loss (error) over all training examples. In this case, the model's average loss on the training set is 0.2222, which is relatively low, suggesting that the model is fitting well to the training data.

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

||Volume 12, Issue 4, April 2023||

|DOI:10.15662/IJAREEIE.2022.1204045 |

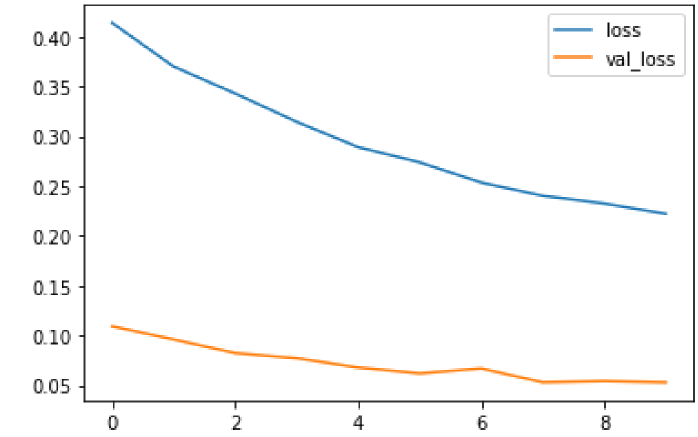


Figure 4.3 Training vs. Validation Loss

Validation Loss (0.0532) represents the average loss (error) over the validation dataset, which consists of data that the model hasn't seen during training. A lower validation loss compared to training loss (which is the case here) typically indicates that the model is not overfitting and generalizes well to unseen data.

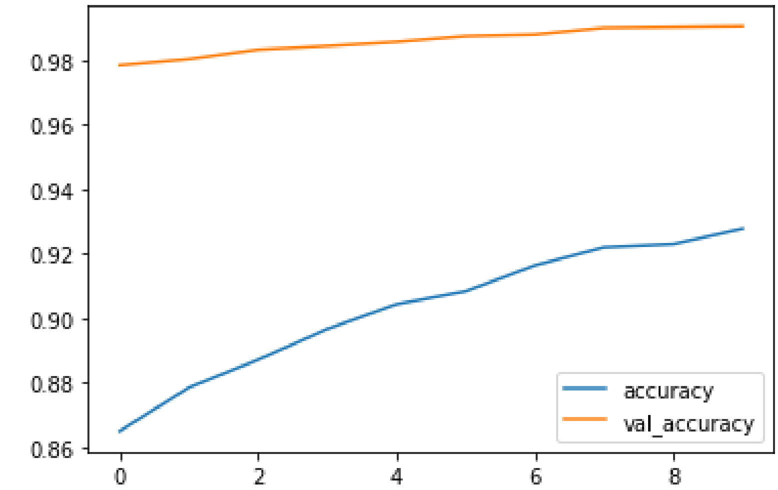


Figure 4.4 Training vs. Validation Accuracy

Training Accuracy (92.77%) indicates the percentage of correctly classified traffic signs in the training dataset. An accuracy of 92.77% suggests that the model is performing well on the training data but there is still room for improvement.

Validation Accuracy (99.04%) represents the percentage of correctly classified traffic signs in the validation dataset. A high validation accuracy of 99.04% indicates that the model is performing exceptionally well on unseen data, which is crucial for real-world applications.

The training process took 182 seconds (about 3 minutes). This duration gives an idea of how computationally intensive the training was, depending on the hardware and batch size used.

V. CONCLUSION

In this study, we have developed and evaluated a deep learning-based model for traffic signal classification, aimed at enhancing the capabilities of autonomous driving systems and driver alert mechanisms. Utilizing the German Traffic Sign Recognition Benchmark (GTSRB) dataset, which comprises 43 distinct traffic sign classes, our model demonstrated significant efficacy in accurately identifying and classifying traffic signs. Our model achieved an

| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|| A Monthly Peer Reviewed & Referred Journal |

||Volume 12, Issue 4, April 2023||

|DOI:10.15662/IJAREEIE.2022.1204045 |

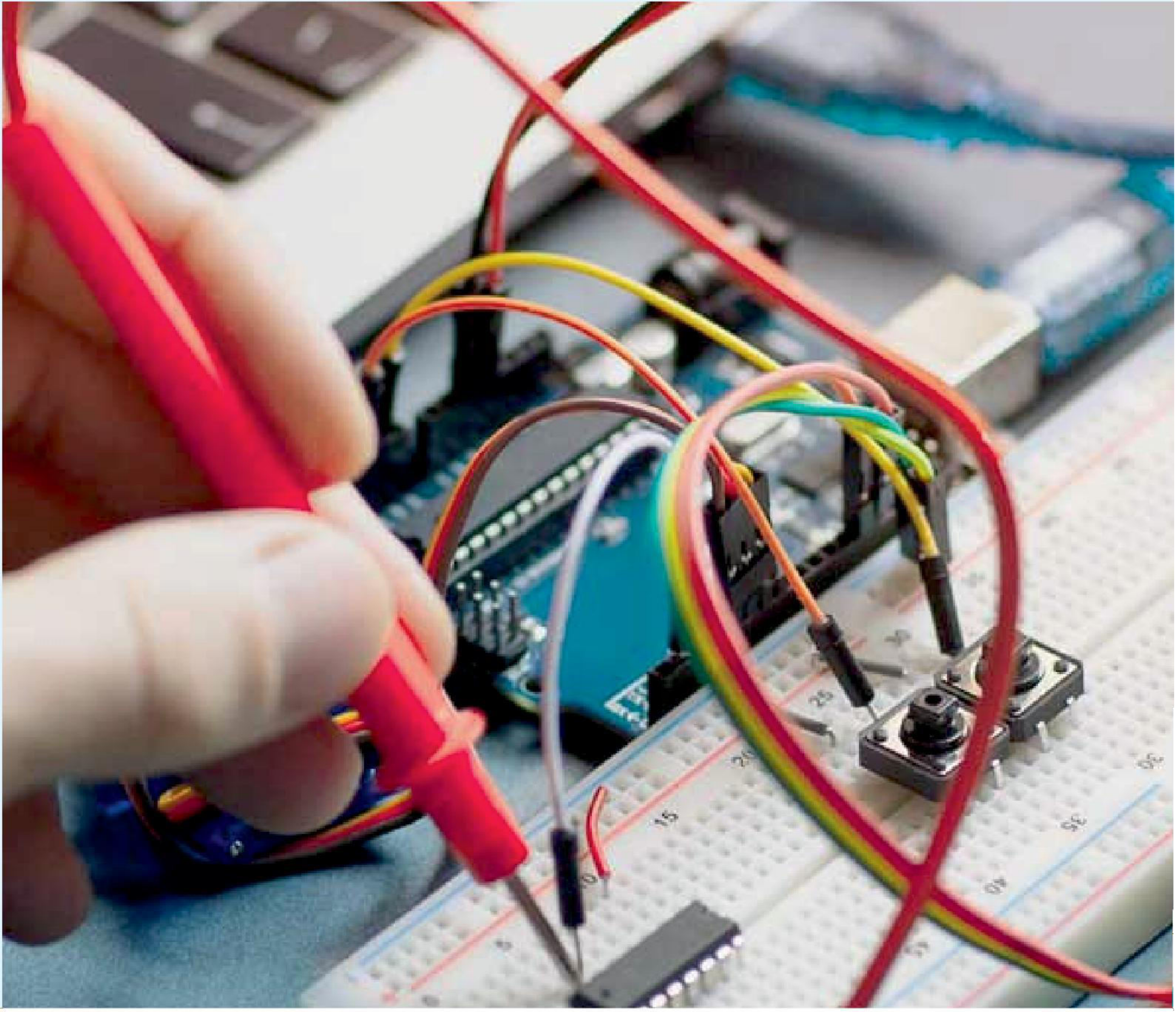
impressive accuracy exceeding 99% on the validation dataset, highlighting its robustness and capability to generalize well to unseen traffic signs. The model reached a training accuracy of 92.77% with a corresponding loss of 0.2222, indicating effective learning and fitting to the training data. Validation results showed a validation accuracy of 99.04% and a validation loss of 0.0532, underscoring the model's ability to maintain high performance on new, unseen data.

VI. FUTURE SCOPE

Further optimization through hyper parameter tuning could potentially improve the model's performance, achieving even higher accuracies and robustness. Integration of the model into real-time systems and testing under diverse environmental conditions would validate its practical applicability and reliability.

REFERENCES

1. Balali V., Ashouri Rad A., Golparvar-Fard M. Detection, classification, and mapping of U.S. traffic signs using google street view images for roadway inventory management. Visualization in Engineering, 2015, vol. 3, no. 1. doi: 10.1186/s40327-015-0027-1
2. Lu Y., Lu J., Zhang S., Hall P. Traffic signal detection and classification in street views using an attention model. Computational Visual Media, 2018, vol. 4, no. 3, pp. 253–266. doi: 10.1007/s41095-018-0116-x
3. Balali V., Golparvar-Fard M. Segmentation and recognition of roadway assets from car-mounted camera video streams using a scalable non-parametric image parsing method. Automation in Construction, 2016, vol. 49, pp. 27–39. doi: 10.1016/j.autcon.2014.09.007
4. Khalilikhah M., Heaslip K. The effects of damage on sign visibility: an assist in traffic sign replacement. Journal of Traffic and Transportation Engineering, 2016, vol. 3, no. 6, pp. 571–581. doi: 10.1016/j.jtte.2016.03.009
5. Kryvinska N., Poniszewska-Maranda A., Gregus M. An approach towards service system building for road traffic signs detection and recognition. Procedia Computer Science, 2018, vol. 141, pp. 64–71. doi: 10.1016/j.procs.2018.10.150
6. Khalilikhah M., Heaslip K. Analysis of factors temporarily impacting traffic sign readability. International Journal of Transportation Science and Technology, 2016, vol. 5, no. 2, pp. 60–67. doi: 10.1016/j.ijtst.2016.09.003
7. Shustanov A., Yakimov P. CNN design for real-time traffic sign recognition. Procedia Engineering, 2017, vol. 201, pp. 718–725. doi: 10.1016/j.proeng.2017.09.594
8. Indolia S., Kumar Goswami A., Mishra S.P., Asopa P.. Conceptual understanding of convolutional neural network – a deep learning approach. Procedia Computer Science, 2018, vol. 132, pp. 679–688. doi: 10.1016/j.procs.2018.05.069
9. Ozturk S., Akdemir B. Effects of histopathological image pre-processing on convolutional neural networks. Procedia Computer Science, 2018, vol. 132, pp. 396–403. doi: 10.1016/j.procs.2018.05.166
10. Kurniawan J., Syahra S.G.S., Dewa C.K., Afiahayati. Traffic congestion detection: learning from CCTV monitoring images using convolutional neural network. Procedia Computer Science, 2018, vol. 144, pp. 291–297. doi: 10.1016/j.procs.2018.10.530
11. Aghdam H.H., Heravi E.J., Puig D. A practical approach for detection and classification of traffic signs using Convolutional Neural Networks. Robotics and Autonomous Systems, 2016, vol. 84, pp. 97–112. doi: 10.1016/j.robot.2016.07.003
12. Stallkamp J., Schlipsing M., Salmen J., Igel C. The German traffic sign recognition benchmark: a multi-class classification competition. Proc. Int. Joint Conference on Neural Networks. San Jose, USA, 2011, pp. 1453–1460. doi: 10.1109/IJCNN.2011.6033395
13. Houben S., Stallkamp J., Salmen J., Schlipsing M., Igel C. Detection of traffic signs in real-world images: the german traffic sign detection benchmark. Proc. Int. Joint Conference on Neural Networks. Dallas, USA, 2013, pp. 1–8. doi: 10.1109/IJCNN.2013.6706807



International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering