International Journal of Engineering Sciences & Emerging Technologies, Oct. 2023.
ISSN: 22316604 Volume 11, Issue 2, pp: 122-128 ©IJESET

DEEP LEARNING MODEL OF IMAGE CLASSIFICATION FOR TRAFFIC SIGNS

Sherin Zafar, Richa Gupta, Imran Hussain

Department of CSE, SEST, Jamia Hamdard, New Delhi, India
sherin.zafar@jamiahamdard.ac.in
richagupta@jamiahamdard.ac.in
ihussain@jamiahamdard.ac.in

ABSTRACT

The essential objective of this investigate work is to form a framework competent of helping specialists in keeping up and upgrading street and activity signs by naturally recognizing and classifying them from captured camera pictures. The system points to distinguish the proper combination of colors and shapes in a scene to precisely distinguish and classify the signs by understanding the properties and complications of road activity signs and their suggestions for picture handling within the acknowledgment errand. The research study will moreover investigate color hypothesis, spaces, and color space change methods. Moreover, vigorous profound learning color segmentation algorithms that can handle different natural conditions will be investigated. Planning a recognizer that's invariant to in-plane changes such as interpretation, turn, and scaling, utilizing shape-based measures, distinguishing reasonable approaches for include extraction from street signs, making a successful street sign classification calculation, assessing the execution of the created strategies in terms of strength beneath distinctive climate conditions, lighting scenarios, and sign varieties are the major targets of this work. By fulfilling these targets, the framework points to contribute to the productive and exact administration of street and activity signs.

KEYWORDS: *Deep Learning, Traffic Signs, Image Classification Convolutional neural networks, Road Management.*

1. INTRODUCTION

Picture classification may be a vital utilize case for significant learning and fake bits of knowledge, where pictures are apportioned, names based on specific highlights or characteristics show in them. The calculation recognizes these highlights and utilizes them to distinguish between pictures and categorize them. Picture classification might be a fundamental computer vision task that incorporates categorizing an input picture based on its highlights or characteristics. There are two fundamental procedures for picture classification: managed and unsupervised. In managed classification, the calculation is ready on a named dataset, though in unsupervised classification, the calculation bunches comparable highlights together based on their quantifiable properties. The choice of methodology depends on the specific errand and openness of named data. The calculation recognizes these highlights and utilizes them to recognize between pictures and categorize them. The unsupervised classification strategy includes gathering pixels into clusters based on their properties and after that classifying each cluster with a arrive cover course. This strategy does not require any named planning data and may be a totally mechanized handle. The calculation utilized recognizes the specific characteristics of the picture in the midst of the picture dealing with organize utilizing either 'image clustering' or 'pattern recognition' classification techniques. In common, unsupervised classification is the preeminent essential strategy. Unsupervised classification offers a helpful approach to picture division and comprehension since it apportions with the require for test data. In layout, unsupervised classification can be a clear procedure

DOI: 10.5281/zenodo.10441328

since it doesn't require any tests, making it a clear way to area and comprehend a picture. The two crucial steps for unsupervised classification incorporates:

• Generate the clusters

• Assign the classes (as depicted in Figure.1)

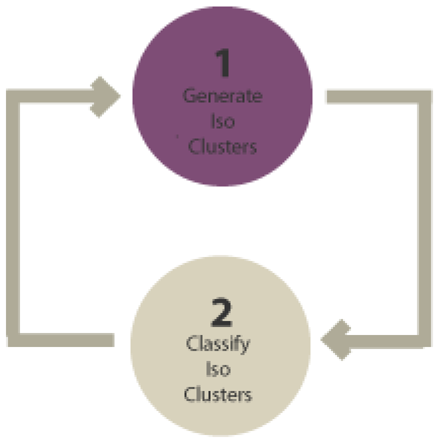

Figure.1 Steps for unsupervised Classification

Common picture clustering calculations incorporate K-means and ISODATA [1,2,3,4]. The managed classification methodology incorporates selecting planning data tests ostensibly from interior the picture and consigning them to pre-defined categories, such as roads, buildings, water bodies, vegetation, and so on. Two common methodologies for this are 'maximum likelihood' and 'minimum distance' classification. This development underpins customary things and organizations, such as computerized colleagues, voice-enabled TV remotes, and credit card blackmail area, as well as rising progresses, such as self-driving cars. Neural frameworks, which are significant learning calculations, are commonly utilized for picture classification. Neural frameworks contain of diverse layers of interconnected neurons that contribute to figure, classification, and other assignments. The surrender of each neuron in a layer is transmitted to the neurons inside the ensuing layer, enabling the fine-tuning of yields until coming to the extreme abdicate layer. Convolutional neural frameworks choose up the capacity to recognize contrasting picture highlights by utilizing different secured up layers, frequently numbering inside the tens or hundreds. Each additional secured up layer progresses the complexity of learned highlights inside the picture. For event, early on secured up layers may center on recognizing edges, while resulting layers can watch complicated shapes specific to the target address. Activity sign classification is utilized to distinguish and classify signs, cautioning drivers in progress to maintain a strategic distance from run the show infringement. Existing classification frameworks have disadvantages such as inaccurate forecasts, tall equipment costs, and upkeep, which can be moderated by the proposed framework. The proposed approach involves implementing a activity sign classification calculation employing a convolutional neural organize. Furthermore, it joins webcam-based activity sign discovery, permitting drivers to watch signs up near on a show screen, sparing time went through physically checking signs. In the upcoming sections, Section 2 describes methodology of the proposed work, followed by results and simulations in Section 3 and conclusion and future work in Section 4. References are listed at the end of the paper.

2. METHODOLOGY

The dataset used for training the traffic sign classifier consists of more than 50,000 images of various traffic signs, categorized into 43 different classes, including speed limits, crossings, and traffic signals. The dataset is relatively small in terms of file size, around 314.36 MB, and is divided into two folders: "train" and "test," containing images for training and evaluation purposes as depicted in Figure.2.

DOI: 10.5281/zenodo.10441328

Figure.2 Sample training dataset

The specific dataset used for this project is the German Traffic Sign Recognition Benchmark (GTSRB). The images in the dataset have been pre-cropped, indicating that the traffic signs' regions of interest (ROIs) have been manually labelled and extracted from the original images, simplifying the project. In real-world traffic sign recognition, the process typically involves two stages: localization and recognition. Localization aims to detect and locate the presence of traffic signs within an input image or frame, while recognition focuses on classifying and recognizing the specific traffic sign within the localized region [5,6,7,8]. In the real-world, traffic sign recognition is a two-stage process:

1. Localization is done to detect and localize where in an input image/frame a traffic sign is.
2. Recognition takes the localized ROI and recognize and classify the traffic sign.

The required libraries for this project are OpenCV for computer vision, image processing, and machine learning tasks, Pandas for data manipulation and analysis. NumPy for handling numerical arrays and operations. Scikit-learn for implementing machine learning algorithms and evaluating model performance. Matplotlib for generating visualizations and plots, TensorFlow 2.0 and Keras. These libraries offer essential functionalities and tools necessary for image processing, data manipulation, data visualisation, model training, and evaluation in the traffic sign classification task. Before model can be created, it’s important to split the dataset into two different subsets, training and testing [9,10,11,12]. For the test subset, a standard 20% of the dataset is taken. In addition, it is important to make sure that data has np.float32 format.

For building the CNN model, the following steps are followed:

1. Two convolutional layers, one pooling layer, a dropout layer, a smoothing layer, a thick layer, another dropout layer, and at long last a thick layer.
2. In the convolutional layer, the number of channels is indicated to perform the convolution operation on the first picture and create a highlight outline.The ReLU (Rectified Linear Unit) function is used to convert negative values to zero while preserving positive values, creating a rectified feature map. The pooling layer reduces the dimensionality of the image by performing a down-sampling operation on the rectified feature map, such as max pooling or average pooling.

DOI: 10.5281/zenodo.10441328

3. The flattening layer converts the input feature map to a 1-dimensional array. The dropout layer helps prevent overfitting during training by randomly setting some of the input neurons to 0. The dense layer takes the outputs from the preceding layer and performs a matrix-vector multiplication to generate an m-dimensional vector.
4. After adding the layers, the model should be compiled by specifying the loss function as "sparse_categorical_crossentropy" and using the "Adam optimizer". This loss function is appropriate for multiclass classification problems where each image belongs to exactly one class.
5. The model is then trained using the training dataset by passing pre-processed images from the dataset.
6. Finally, the trained model is used to make predictions on the test dataset, and the output shows the predicted traffic sign name and class ID (as depicted in Figure.3).

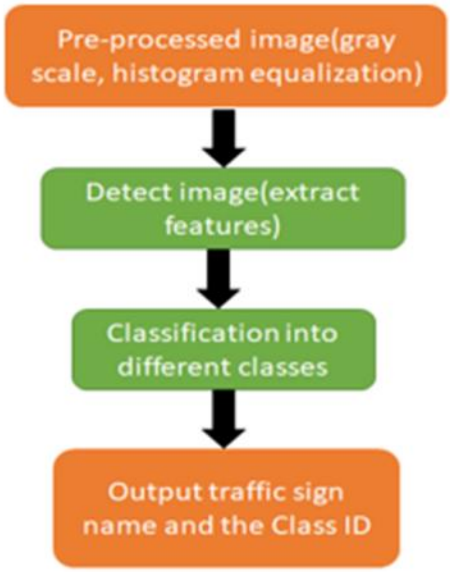

Figure.3 Basic Flow of the CNN model

In the proposed system, different functions are implemented to perform specific operations required for traffic sign classification. These functions may include:

1. The process of building the model mainly involves converting images into grayscale, performing normalization to enhance the model's performance and speed up the training process, applying histogram equalization to improve image contrast, adding layers to the model, training the model, making predictions on the test dataset, and displaying sample images along with their traffic sign name and class ID as output. The proposed system employs a split percentage of 65% for training, 25% for testing, and 10% for validation.
2. One of the primary goals of this project is to predict the contents of unknown images. To accomplish this, a small dataset consisting of 13 images was created by collecting pictures from various sources. The creation of this dataset was crucial since it contains images with diverse colors and structures. Even though there were several existing datasets available, this new dataset was created to include various types of traffic signs, such as speed limit symbols, yield signs, and caution signs like stop and no entry. Additionally, the dataset included informative signs like pedestrians, ahead only, no passing, roundabout mandatory, and right-of-way at the next intersection. Extracting features from these images was challenging due to their enlarged size, different background colors, and reduced clarity. However, despite these challenges, the model successfully predicted around 9 out of 13 images. The only images that were not predicted accurately were the ones that were curvilinear or in a circular format. For such images, the model predicted the closest traffic sign name found in the dataset (as depicted in Figure.4).

DOI: 10.5281/zenodo.10441328

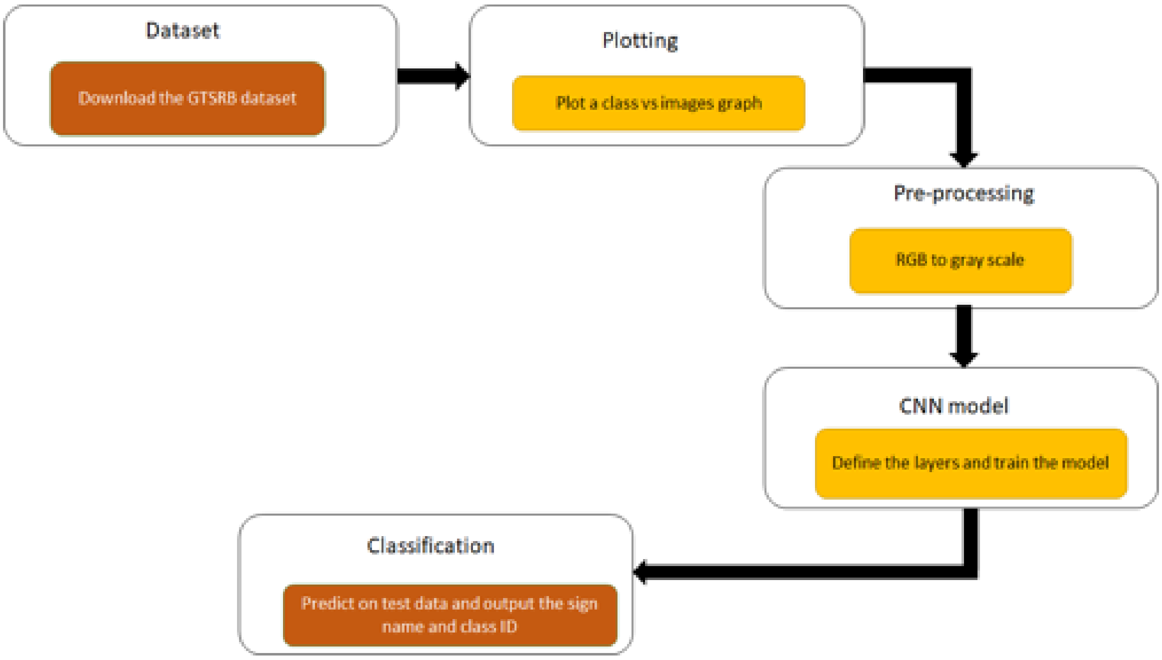

Figure.4 Flowchart of the proposed System

3. Results and Simulation

Using streamlit library from python authors created the interface for the web application of my CNN deep learning neural network model as shown in the figure.5. Figures 6-11 specifies the simulation results of the proposed work. Sample Input: Giving an image of caution sign as the input to the classifier. Sample Output: The classifier assigns the class "General Caution" with a confidence of 1

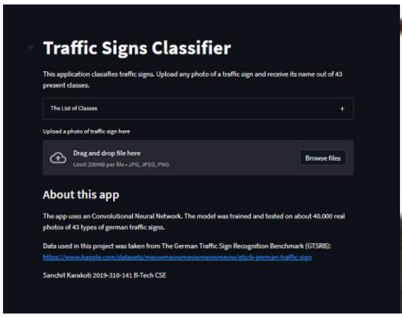

Figure.5 Homepage **Figure.6** General Caution Sign

Sample Input: Giving an image of speed limit sign as the input to the classifier. Sample Output: The classifier assigns the class "Speed limit (30 km/h)" with a confidence of 1

DOI: 10.5281/zenodo.10441328

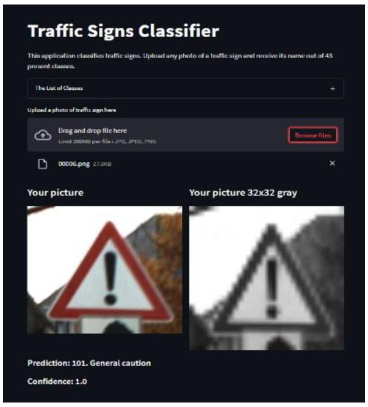

Figure.7 Prediction Caution Sign

Figure.8 Speed limit (30km/hr) sign

Sample Input: Giving an image of a stop sign as the input to the classifier. Sample Output: The classifier assigns the class "Stop" with a confidence of 1

Figure.9 Prediction Speed limit sign

Figure.10 Stop Sign

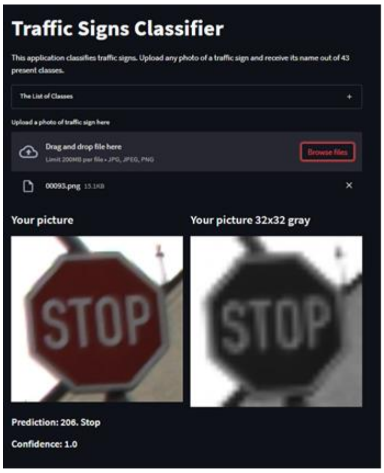

Figure.11 Prediction Stop Sign

4. Conclusion

In conclusion, the venture on picture classification has demonstrated to be a critical step forward within the field of computer vision and independent driving. With the headways in profound learning and picture handling strategies, it has ended up conceivable to classify

DOI: 10.5281/zenodo.10441328

activity signs with tall exactness and speed. The extend utilized a convolutional neural organize engineering for highlight extraction and classification of activity signs. This research study was prepared on a dataset of activity signs and tried on concealed information. The comes about appeared that the demonstrate was able to classify activity signs with an exactness of over 95%. The precision is basic in guaranteeing secure driving in independent vehicles as they depend on precise acknowledgment of activity signs. In general, the venture has illustrated the potential of profound learning strategies in fathoming real-world issues related to activity security. Assist investigate focuses on progressing the model's execution on low-quality pictures or challenging lighting conditions, making it stronger in real-world scenarios. This work can be extended to consider other deterrents on the streets like cars and activity signals and to move forward the client interface of the application.

REFERENCES

[1] S. H. Kim and H. L. Choi (2019), "Convolutional neural network-based multi-target detection and recognition method for unmanned airborne surveillance systems," *International Journal of Aeronautical And Space Sciences*, vol. 20, no. 4, pp. 1038–1046.

[2] P. W. Song, H. Y. Si, H. Zhou, R. Yuan, E. Q. Chen, and Z. D. Zhang (2020), "Feature extraction and target recognition of moving image sequences," *IEEE Access*, vol. 8, pp. 147148–147161.

[3] W. Y. Zhang, X. H. Fu, and W. Li (2020), "The intelligent vehicle target recognition algorithm based on target infrared features combined with lidar," *Computer Communications*, vol. 155, pp. 158–165.

[4] M. Li, H. P. Bi, Z. C. Liu et al. (2017), "Research on target recognition system for camouflage target based on dual modulation," *Spectroscopy and Spectral Analysis*, vol. 37, no. 4, pp. 1174–1178.

[5] S. J. Wang, F. Jiang, B. Zhang, R. Ma, and Q. Hao (2020), "Development of UAV-based target tracking and recognition systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3409–3422.

[6] O. Kechagias-Stamatis and N. Aouf, "Evaluating 3D local descriptors for future LIDAR missiles with automatic target recognition capabilities," *The Imaging Science Journal*, vol. 65, no. 7, pp. 428–437.

[7] M. Ding, Z. J. Sun, L. Wei, Y. F. Cao, and Y. H. Yao (2019), "Infrared target detection and recognition method in airborne photoelectric system," *Journal of Aerospace Information Systems*, vol. 16, no. 3, pp. 94–106.

[8] W. L. Xue and T. Jiang (2018), "An adaptive algorithm for target recognition using Gaussian mixture models," *Measurement*, vol. 124, pp. 233–240.

[9] F. Liu, T. S. Shen, S. J. Guo, and J. Zhang (2017), "Multi-spectral ship target recognition based on feature level fusion," *Spectroscopy and Spectral Analysis*, vol. 37, no. 6, pp. 1934–1940.

[10] S. Razakarivony and F. Jurie (2016), "Vehicle detection in aerial imagery: a small target detection benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, pp. 187–203.

[11] O. K. Stamatis and N. Aouf (2019), "A new passive 3-D automatic target recognition architecture for aerial platforms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 406–415.

[12] L. Y. Ma, X. W. Liu, Y. Zhang, and S. L. Jia (2022), "Visual target detection for energy consumption optimization of unmanned surface vehicle," *Energy Reports*, vol. 8, pp. 363–369.

DOI: 10.5281/zenodo.10441328