DOI: https://doi.org/10.46632/daai/2/4/1

Data Analytics and Artificial Intelligence
Vol: 2(4), 2022
REST Publisher; ISBN: 978-81-948459-4-2
Website: http://restpublisher.com/book-series/data-analytics-and-artificial-intelligence/

Automatic Traffic Sign Detection and Recognition Using Deep Convolution Neural Network

[*] Shanmugapriya E, K. Arul Deepa
College of Engineering Guindy, Chennai, Tamil Nadu, India.
*Corresponding author Email: vijaylaya2000@gmail.com

Abstract. Traffic signs are significant security facilities on the road, which plays an important role in regulating traffic behavior, ensuring the safety of the road and guiding the smooth passage of vehicles and pedestrians, etc. As part of the intelligent transportation system, the detection of traffic signs is significant for the driving assistance system, traffic sign maintenance, autonomous driving and other spaces. There are a number of research works done for traffic sign detection in the world. But most of the works are only for certain categories of traffic signs, for example, the speed limit sign. Traffic sign detection is generally regarded as challenging due to various complexities, for example, diversified backgrounds of traffic sign images. Two major problems exist in the process of detection and recognition of traffic signals. Road signs are frequently occluded partially by other vehicles and many objects are present in traffic scenes which make the sign detection hard and pedestrians, other vehicles, buildings and billboards may confuse the detection system by patterns similar to that of road signs. Also color information from traffic scene images is affected by varying illumination caused by weather conditions, time (day night) and shadowing.
Keywords: Automatic traffic sign detection, Traffic Sign Detection, Traffic sign Classification

1. Introduction

The Automatic traffic sign detection and recognition is an interesting topic in computer vision and is especially important in the context of autonomous vehicle technology. To ensure road safety, it is crucial to driverless vehicles and modern driver support technologies. ATSDR is a challenging problem to tackle in real time because of the diversity in the types of traffic signs and varied environmental conditions presented in practical road conditions. Besides, motion artifacts introduced in a live traffic video feed make the task of detection and recognition even more difficult. The problem is being studied for a long time and various methods have been proposed to address it. Broadly, these methods divide the task into two parts, Traffic Sign Detection and Traffic Sign Recognition, and address each of them separately. The detection step constitutes localizing regions in an image that may contain traffic signs while the recognition step classifies these localized regions to specific sign types or backgrounds. Traffic Sign Detection using convolution neural networks approach has shown significant improvement in image recognition, segmentation and object detection. However, in the traffic sign detection datasets, traffic signs occupy less 2 than 3% of the image. Therefore, the object to image ratio poses a major difficulty in devising a model for segmenting such tiny objects. The method requires more computational power and large amounts of data to train really deep networks. Challenging environmental conditions affect image capture in practical road scenarios. The challenging conditions, combined with the size constraints of the signs and the sheer volume of data required to process, makes this very difficult. Deep learning algorithms based on Convolution Neural Networks have shown significant performance in computer vision tasks such as image recognition, segmentation and object detection across all benchmarks. Thus, it is a fruitful prospect to consider using these techniques in ATSDR systems. Another hurdle in using CNNs efficiently to approach this problem is the amount of data required to train really deep networks. A large number of images with high resolutions are used to facilitate research in the field of Traffic Sign Detection and Recognition. The localizer is implemented by Encoder-Decoder Architecture followed by a Fully Convolution Neural Network, which is inspired by SegNet and UNet Architecture. The network consists of two stages: an Encoder-Decoder and a neural network, which are connected by using a residual learning strategy. The Classifier module predicts the traffic sign from the Boundaries region detected by the Localizer. The Neural Networks are trained with images from German Traffic Sign Detection Benchmark Dataset and German Traffic Sign Recognition Benchmark Dataset. The GTSRB dataset consists of more than 50,000 images indicating 43 3 classes. The GTSDB dataset consists of 900 images which are used by localizer modules to extract and predict traffic sign region boundaries.

Problem Statement: Traffic Sign Detection using convolution neural networks approach has shown significant improvement in image recognition, segmentation and object detection. The entire project is divided into two main modules, localizer and classifier. The localizer is implemented by Encoder-Decoder and Fully Convolution Neural Network architecture. The encoder stage consists of several convolution blocks, each with two 3×3 convolution layers, a batch normalization layer and a ReLU activation layer. After each block, a 2×2 max pooling operation is performed for down sampling. The index of the maximum element is saved for use during up sampling. The number of feature channels is doubled at each down sampling step. Next decoder block is implemented. At each step of decoding, the feature map is up sampled by using a 2×2 up convolution. During up sampling, an element is copied to the index saved from the corresponding max pooling layer of the decoder. This up convolution halves the number of feature maps, which are then concatenated with the corresponding feature map of the decoder. Next, the concatenated feature map is passed through two 3×3 convolution layers and a ReLU activation layer.

In the final layer, a 1×1 convolution layer is used to map each component feature vector from the previous layer to 2 classes. Next a convolution Neural Network is implemented and added to the Encoder - Decoder Network. From the predicted boundaries by the Encoder – Decoder and 32 * 32 squared images are resized and generated from the predicted boundaries. This image is passed into another CNN. The architecture of the CNN used to recognise the localized traffic sign regions; it is inspired from the VGG-16 architecture, which is widely used for image recognition. It has two convolution blocks, each with two consecutive 3×3 convolution layers, a ReLU activation layer, and a max-pooling layer. A dropout layer, with probability of 0.25, is added after each block. Finally, the features are passed through two fully connected layers- the first one has a hidden size of 1024 with ReLU activation, and the latter has a hidden size of 15 with soft max activation.

2. Related Work

Our Project involves three Core Modules: Patch Extractor, Localizer and classifier. Apart from these modules we also need a basic frontend module which can act as facade to all core modules. Video object detection**:** Alberto Sabater, Luis Montesano, Ana C. Murillo proposed a Robust and efficient post-processing for video object detection in 2020 which introduces a novel post-processing pipeline that overcomes some of the limitations of previous post-processing methods by introducing a learning-based similarity evaluation between detections across frames. Our method improves the results of state-of-the-art specific video detectors, especially regarding fast moving objects, and presents low resource requirements. And applied to efficient still image detectors, such as YOLO, provides comparable results to much more computationally intensive detectors. In 2015 Qichang Hu, Sakrapee Paisitkriangkrai, Chunhua Shen, Anton van den Hengel, and FatihPorikl Proposed a Common Detection Framework for Fast Multiple Object Detection in Traffic Scenes that has strong correlations between target sizes and positions for traffic scenes, precise boundaries with high accuracy, and By optimizing the resolution of network input for the best trade-off between speed and accuracy. Detectors from three significant classes are included in the proposed framework, along with a dense feature extractor. Following their extraction, all detectors can use the dense features. Since all dense characteristics only need to be evaluated once during the testing phase, the advantage of adopting one common framework is that detection speed is substantially faster. Contrarily, the majority of earlier efforts used distinct detectors employing various properties for each of these three types. The framework adds spatially pooled characteristics as a component of aggregating channel features to increase the robustness of the feature against disturbances and picture deformations. An object subcategorization strategy is suggested as a way to capture the intraclass variance of objects in order to enhance generalisation performance. Experiments with three different detection applications—traffic sign, car, and bike detection to show the usefulness and effectiveness of the suggested framework. For object classes with a large intra-class variation like cars, the appearances and shapes of cars change significantly as viewpoints change. In order to deal with these variations that cannot be tackled by the conventional VJ framework, An object subcategorization method which aims to cluster the object class into visually homogeneous subcategories. The proposed subcategorization method applies an unsupervised clustering method to one specific feature space of the training samples to generate multiple subcategories. This method simplifies the original learning problem by dividing it into multiple sub-problems and improves model generalization performance.

Traffic Sign Detection*:* Hee Seok Lee and Kang Kim in 2017 developed a Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolution Neural Network which focuses on Detection time is towards real-time in video sequences, Traffic sign detection step can reduce the disturbing of complex background and the texts which are not in traffic signs and Adopt the latest architectures for object detection such as feature pyramid network. The boundary estimation of traffic sign is formulated as 2- D pose and shape class prediction problem, and this is effectively solved by a single CNN. With the predicted 2-D pose and the shape class of a target traffic sign in the input, Estimating the actual boundary of the target sign by projecting the boundary of a corresponding template sign image into the input image plane. By formulating the boundary estimation problem as a CNN-based pose and shape prediction task, our method is end-toend trainable, and more robust to occlusion and small targets than other boundary estimation methods that rely on contour estimation or image segmentation. With our architectural optimization of the CNN-based traffic sign detection network, the proposed method shows a detection frame rate higher than seven frames/second while providing highly accurate and robust traffic sign detection and boundary estimation. In 2018, Yingying Zhu ,Minghui Liao , Mingkun Yang , and Wenyu Liu developed a Cascaded Segmentation-Detection Networks for Text-Based Traffic Sign Detection. A novel text-based traffic sign detection framework with two deep learning components. More precisely, we apply a fully convolution network to segment candidate traffic sign areas providing candidate regions of interest (RoI), followed by a fast neural network to detect texts on the extracted RoI. The proposed method makes full use of the characteristics of traffic signs to improve the efficiency and accuracy of text detection. On one hand, the proposed two-stage detection method reduces the search area of text detection and removes texts outside traffic signs. On the other hand, it solves the problem of multi-scales for the text detection part to a large extent. Extensive experimental results show that the proposed method achieves the state-of-the-art results on the publicly available traffic sign data set: Traffic Guide Panel data set. Specifically, shallower stages can acquire features of smaller objects while deeper stages are able to catch features of larger objects. Inspired by the property of the multi-stage convolution network, a side output with up sampling to the same size as the input image is added after each stage to get multi-scale output. We observe that this nested multi-scale framework is insensitive to input image sizes. With a fixed-size input, our proposed method can detect signs of variable scales. The side output prediction up sampling is simply implemented with a deconvolution layer composed of a 1×1 convolution layer and an up sampling layer, where the parameters of up sampling are fixed to perform bilinear interpolation. Then the hierarchical fusion maps are generated by concatenating these up sampled side outputs along the channel. Finally, a 1 × 1 convolution layer is deployed on the top to aggregate the information over the channel and a sigmoid layer follows to make the pixel-level sign/non-sign prediction. This paper considers the traffic sign detection task as a pixel-level sign/non-sign classification. So at the training stage, pixels within the bounding box of each traffic sign are treated as the positive region and vice versa. That is to say, we label the pixels within the traffic sign region to 1, and the pixels outside the traffic sign region to 0. FCN in this stage does not focus on where texts are, but where traffic signs are. Ideally, we can filter most non-sign regions in this stage, which greatly enhances the efficiency of the latter text detection. Traffic sign Classification: In 2015 Yi Yang, Hengliang Luo, Huarong Xu, and Fuchao Wu proped a Real-Time Traffic Sign Detection and Classification which focuses on End-toend trainable means combine our sign detector and text detector into one unified network, Generalized the object bounding box detection problem. A color probability model to deal with color information of traffic signs, so as to enhance the specific colors (e.g., red, blue and yellow) of traffic signs and suppress background colors as well as to reduce the search space of following algorithms and detection time. To further reduce detection time, we propose to extract traffic sign proposals instead of sliding window detection and combine some machine learning algorithms, i.e., SVM and CNN, to detect and classify traffic signs.

3. Methodology

This chapter describes the overall system design and the detailed explanation about each of the modules and the corresponding underlying algorithms. The overall Architecture diagram for the proposed system is shown in fig.1. The proposed work is split into different modules namely Path Extractor, Localizer, Boundary Extraction and Classifier. The proposed system has four modules. Path Extractor, Localizer, Boundary Extractor and Classifier.

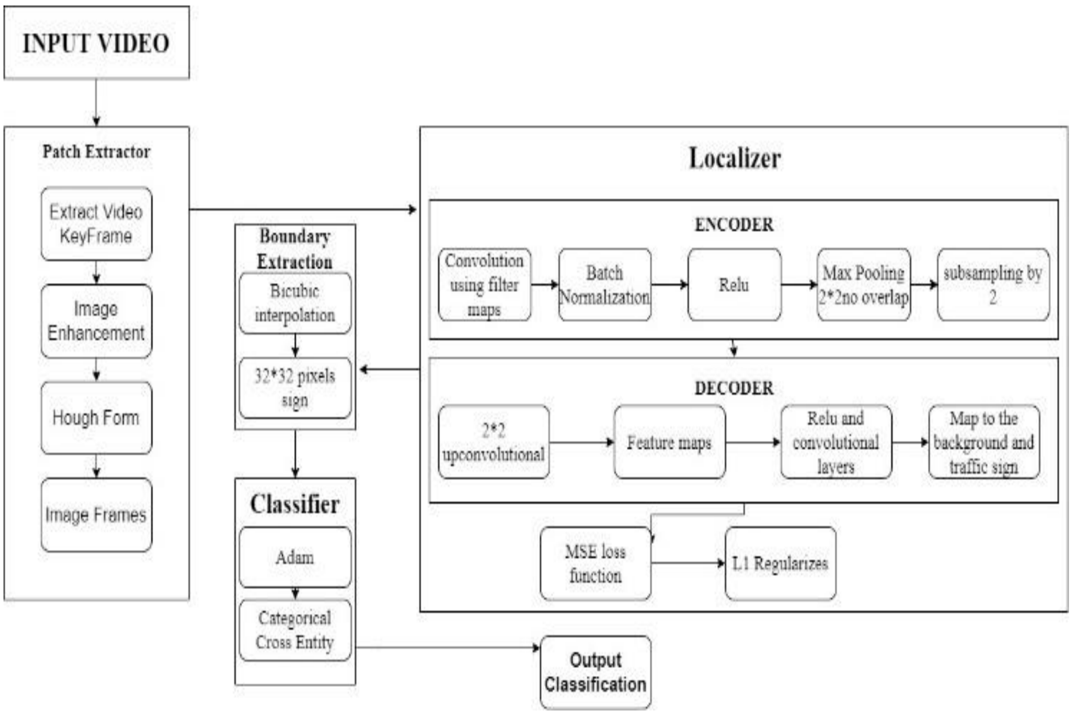

FIGURE 1. Architecture of Proposed model

Path Extractor Module: The Path Extractor approach is adopted in designing our pipeline so that we can independently work on different tasks of traffic sign detection. Extractor Design allows us to use different optimization schemes in different parts of the pipeline. Neural networks excel at performing in highly specific tasks. As a result, this approach gives us the opportunity to leverage the strengths of various architectures and optimize the performance of each network without affecting the accuracy of the others. The Neural Network is trained in constrained hardware environments; it is much more manageable to train several deep networks for each specific tasks instead of training a single very deep one for the overall system.

Localizer: The localizer is implemented by Encoder-Decoder Architecture, a Fully Convolution Neural Network architecture, which is inspired by Seg Net and U Net. The network consists of two stages: an encoder (down sampling stage) and a decoder (up sampling stage), which are connected by using a residual learning strategy.

Up sampling: The encoder stage consists of several convolution blocks, each with two 3×3 convolution layers, a batch normalization layer and a ReLU activation layer. After each block, a 2×2 max pooling operation is performed for the index of the maximum element is saved for use during up sampling. The number of feature channels is doubled at each down sampling step.

Down sampling: At each step of decoding, the feature map is up sampled by using a 2×2 up convolution. • During up sampling, an element is copied to the index saved from the corresponding max pooling layer of the decoder. • This up convolution halves the number of feature maps, which are then concatenated with the corresponding feature map of the decoder. • Next, the concatenated feature map is passed through two 3×3 convolution layers and a ReLU activation layer.

In the final layer, a 1×1 convolution layer is used to map each component feature vector from the previous layer to 2 classes. The Localizer is developed such that it resembles SegNet and U-Net architectures, which resulted in an increased detection of small signs. U-Net has its robustness against low resolution features of the image during training. It uses a lot of feature channels in its up sampling layers, allowing the network to pass contextual data to higher resolution layers. This is quite an attractive feature for localizing objects in noisy conditions where the target object may not be dominantly visible in the image. SegNet employs an elegant method of up sampling in the decoder stage. The decoders use the max-pooling indices received from their corresponding encoders to perform up sampling. This enables the network to segment pixels much more accurately than the conventional up convolution. A Convolution Neural Network is added to the encoder and decoder, to predict the boundary boxes. The CNN fallows few 3 * 3 convolution layers, 2 * 2 max pooling layers, Dense layers and final layer containing 4 neurons.

Boundary Extraction Module: The Boundary Extraction Module first converts the localizer predicted co-ordinates into full valued numbers. The traffic sign detection in image patches of size 800 x 1360 are valued from the bottom right of the input image. Next, we extract the proposed sign regions from the input image, which correspond to islands of high probability in the image mask. The coordinates of detected region are calculated. A new 32*32*3 square image is resized from the detected boundaries. The Square image is used for Traffic Sign Recognition.

Classification Module: The Classifier Module fallows the architecture similar to VGG-16. The VGG-16 architecture is generally defined as convolution blocks. Each convolution block contains 2 convolution layers followed by a Max Pooling layer. The convolution blocks are repeated 3 to 4 times. Next few Fully connected Dense Layers are attached to these convolution blocks. Finally, the 19 Output Layer with contains Softmax as activation function. Apart from the Output Layers all the other layers have ReLu as activation function. Totally the standard architecture contains 3 convolution blocks with 9 Layers, 3 Fully connected Layers and Finally 1 Output Layer. So totally 13 Layers are added to the classifier Module. Since more layers are used for very small input image. The Classifier Module fallows the architecture similar to VGG-16. The VGG-16 architecture is generally defined as convolution blocks. Each 19 convolution block contains 2 convolution layers followed by a Max Pooling layer. The convolution blocks are repeated 3 to 4 times. Next few Fully connected Dense Layers are attached to these convolution blocks. Finally the Output Layer with contains Softmax as activation function. Apart from the Output Layers all the other layers have ReLu as activation function. Totally the standard architecture contains 3 convolution blocks with 9 Layers, 3 Fully connected Layers and Finally 1 Output Layer. So totally 13 Layers are added to the classifier Module. Since more layers are used for very small input image. The Neural Network Accuracy must be very good.

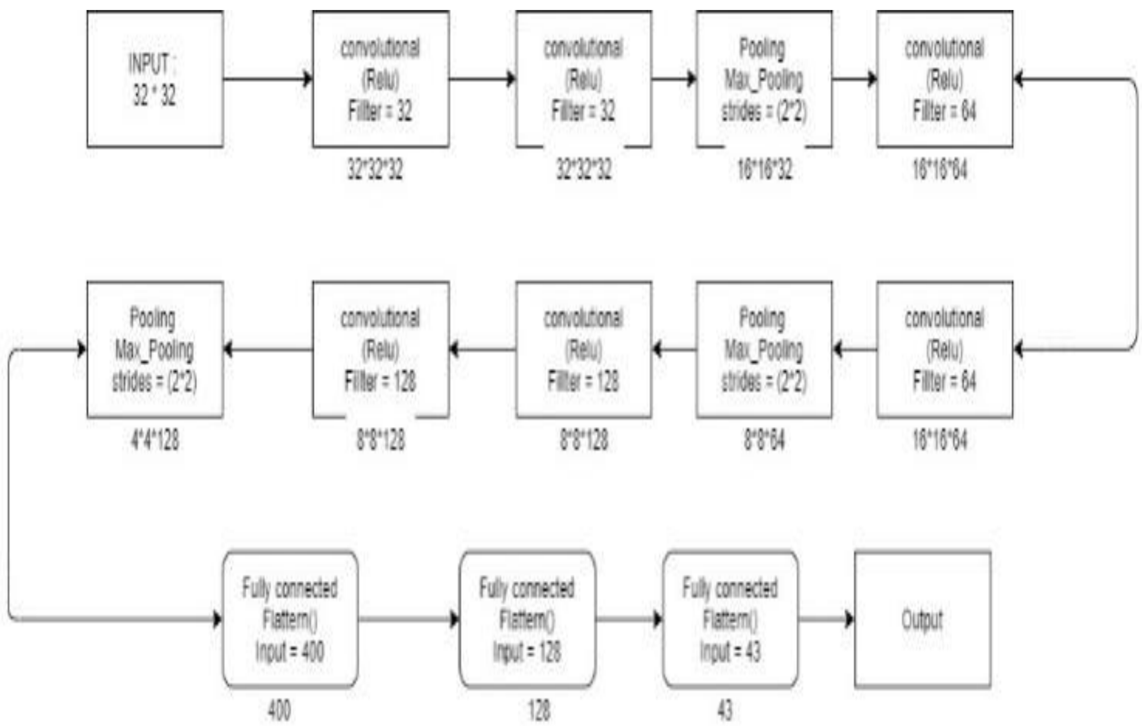

FIGURE 1 Block diagram of Classifier

The Classifier Module defined with 13 layers does a pretty good work, due to Deep structure for 32*32 Input Image. There are many chances for overfitting the Classifier Neural Network. The Neural Network is modified by adding few additional layers to reduce overfitting also to increase the Accuracy, Precision and Recall.

Loss function: Categorical Cross Entropy Loss function is used for multi-class classification. We will train a CNN to output a probability over the certain classes for each image. CCE is used when true labels are one-hot encoded.

$$CCE = -\frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{J} y_j \cdot log(\hat{y}_j) + (1 - y_j) \cdot log(1 - \hat{y}_j)$$

Above equation is used for calculating the loss between true labels predicted and actual labels. CCE Loss function helps the classifier network to improve the accuracy.

4. Result and Discussions

Dataset Description: German Traffic Sign Recognition Benchmark Dataset contains more than 50,000 images in total. GTSRB contains a total of 43 classes. The Classifier Module uses these images as inputs and classes hot one coding values as output. The images are real time cropped images and few images in the dataset are blurry, so proper preprocessing techniques are used to make images stabilize. German Traffic Sign Detection Benchmark Dataset contains 900 images. GTSDB Dataset images are mapped with the respective Traffic sign co-ordinates in the image. The Localizer Module uses these images as input and traffic sign coordinates as output. The images are taken from real time scenarios in German. Each image dimension is 800 * 1360. The image is divided into two categories one is traffic sign region and background region. Data Pre-processing: The following preprocessing techniques are applied to the GTSRB Dataset: Shuffling, Gray scaling, Local Histogram Equalization, Normalization. Shuffling: In general, shuffling the training data to increase randomness and variety in the training dataset, in order for the model to be more stable. The library sk learn is used to shuffle our data. Gray scaling: Using grayscale images instead of color improves ConvNet's accuracy. The Open CV is used to convert the training images into grayscale.

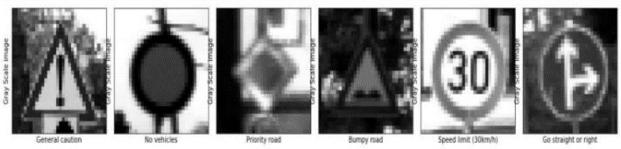

FIGURE 3 Local Histogram Equalization

Local Histogram Equalization - This technique simply spreads out the most frequent intensity values in an image, resulting in enhancing images with low contrast. Applying this technique will be very helpful in our case since the dataset in hand has real world images, and many of them have low contrast. The Library sk image to apply local histogram equalization to the training images.

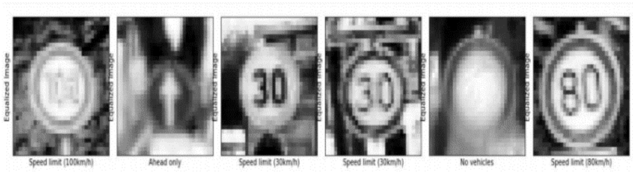

FIGURE 4 Normalization

Normalization - Normalization is a process that changes the range of pixel intensity values. Usually the image data should be normalized so that the data has mean zero and equal variance.

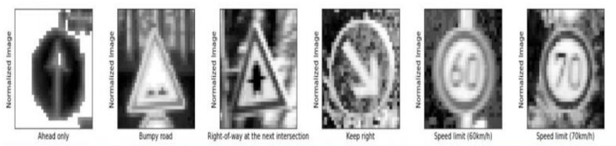

FIGURE 5

Results for Modules: For training the localizer, first, we extract all the image frames from the training Dataset which contain traffic signs. The Dataset used for training is GTSDB. This is done by using the annotation files generated for the corresponding image with the dataset. Each image has a frame size of 800 × 1360 pixels. As stated earlier, in the encoder stages of the localizer, the 2 ×2 max pooling operations halve the number of feature maps at each step. If the input dimensions of the features are not a power of 2, then, at some stage of the encoder, we may need to crop the feature map to match the dimensions of the pooling operation. This will result in a loss of context information during forward propagation and have a detrimental effect on the accuracy of the final output. The Localizer Module Contains Encoder Blocks Which Down samples the Input Image. The input dimensions are 800 * 1360 pixels. The Encoder Block 1 Down samples the Network to 400*680 pixels. The Encoder Block 2 again Down Samples the Network to 200*380 pixels. The Encoder block contains a 3*3 convolution layer followed by a 2*2 max pooling layer. Next 2 Decoding blocks are used to Up sample the images. Decoder Block concatenates two encoder blocks which are associated with different filters. The Decoder block also contains Dropout layers to avoid over fitting. The Localizer predicts the traffic sign coordinates in the image. The parameters used to measure this module are Accuracy and Mean Square Error. The Accuracy for the proposed Module is 0.6805.

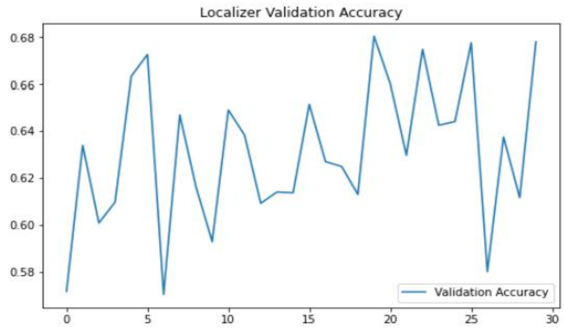

FIGURE 6 Localizer module Validation Accuracy

Mean Square Error tells how close the predicted coordinates are to the actual coordinates. MSE is calculated by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Ytrue - Ypred)^2$$

-------------- (1)

Where, MSE - Mean Square error,
N - Number of coordinates to be predicted
Y true - True coordinate values,
Y pred - Predicated Coordinate Values

The MSE for the proposed Module is 0.3040. Classifier Neural Network Module classifies the given image to the respective class. So, the evaluation is carried out using the basic parameters used for evaluating predictions namely accuracy, precision, recall and the f-measure. The Precision for the proposed Module is 0.9954. The Recall for the proposed Module is 0.9937. The Accuracy for the proposed Module is 0.9944. The F1 Score for the proposed Module is 0.9946. The VGG architecture is generally defined as convolution blocks. Each convolution block contains 2 convolution layers followed by a 2*2 Max Pooling layer. The convolution blocks are repeated 3 times. Next 3 Fully connected Dense Layers are attached to these convolution blocks. Finally, the Output Layer contains Softmax as activation function. We tried for the another for the classification with some dropouts The Classifier Neural Network contains same structure as VGG. To overcome the overfitting problem in the VGG Neural Network few layers are added and other few are modified, but the working architecture is almost same. We also compared our proposed model with some Classifiers like Lenetclassfier, VGG-16 Classifier, We also provided the comparison between each classifier.

Table 1 Standard metric comparison between classifier

Neural Network	Accuracy	Precision	Recall	F1-Score
CLASSIFIER	99.44	99.54	99.37	99.46
LENET	96.83	97.50	96.15	96.82
VGG	98.02	98.31	97.79	98.05

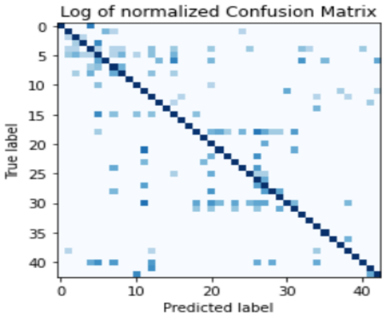

FIGURE 7 Confusion Matrixes for Classifier Model

5. Conclusion

It is clearly seen that the accuracy of the classifier module is certainly higher. From the following results we can see that the Classifier Module is doing a good job in classifying different types of traffic signs when the extracted signs are cropped perfectly from the image. Our approach lacks to give good results when the extracted signs from test images are cropped incorrectly. The Localizer Module should be further trained with large dataset and image Labelling should be improved to produce better Result. Future work: The Neural Network can be further trained with different real time challenging conditions like snow and rain. The Growth in distributive training can be used to train large dataset like CURE-TSD Dataset to improve the Network. Future improvements can be made for extracting signs from test images by using advanced segmentation methods.

Reference

[1]. Ahmed Madani and Rubiyah Yusof, (2014) "Malaysian traffic sign dataset for traffic sign detection and recognition systems", Journal of Telecommunication Electronic and Computer Engineering (JTEC), vol. 8, no. 11, pp. 137-143.

[2]. Aljarrah I. and Mohammad, D. (2018) "Video content analysis using convolution neural networks," 9th International Conference on Information and Communication Systems (ICICS), Irbid, pp. 122-126.

[3]. Bahlmann, C. Y., Ramesh, V. Pellkofer, M. and Koehler, T. (2005) "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information, " in Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 255-260.

[4]. Broggi, A. Cerri, P. Medici, P. Porta, P. P. and Ghisio, G. (2007) "Real time road signs recognition, " in Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 981-986.

[5]. Changzhen, X. Cong, W. Weixin M. and Yanmei, S. (2016) "A traffic sign detection algorithm based on deep convolution neural networks," IEEE International Conference on Signal and Image Processing (ICSIP), Beijing, pp. 676-679.

[6]. Chen, Y. Chuang Y. and Wu, A. (2020) "Online Extreme Learning Machine Design for the Application of Federated Learning," 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Genova, Italy, pp. 188-192.

[7]. Chen, L.C. Papandreou, G. Kokkinos, I. Murphy, K. and Yuille A. L. (2014). “Semantic image segmentation with deep convolution nets and fully connected CRFs”.39

[8]. Ciresan, D. C. Meier, U. Masci, J. and Schmidhuber, J. (2011) "A committee of neural networks for traffic sign classification, " in Proceedings of the IEEE International Joint Conference on Neural Networks, pp. 1918-1921.

[9]. Ciresan, D.C. Meier, U. Masci, J. and Schmidhuber, J. (2012) "Multicolumn deep neural network for traffic sign classification", Neural Networks, vol. 32, pp. 333-338.

[10]. Duan et al., M. (2019) "Astraea: Self-Balancing Federated Learning for Improving Classification Accuracy of Mobile Deep Learning Applications," IEEE 37th International Conference on Computer Design (ICCD), Abu Dhabi, United Arab Emirates, pp. 246-254.

[11]. Ellahyani, A. El Ansari, M. and El Jaafari, I. (2016) “Traffic sign detection and recognition based on random forests,” Appl. Soft Comput., vol. 46, pp. 805–815.

[12]. Fleyeh H. and Davami, E. (2011) "Eigen-based traffic sign recognition," in IET Intelligent Transport Systems, vol. 5, no. 3, pp. 190-196.

[13]. Garcia-Garrido, M. A. Ocaña, M. Llorca, D. F. Sotelo, M. Arroyo, E. and Llamazares, A. (2011) “Robust traffic signs detection by means of vision and v2i communications,” in Proc. 14th Int. IEEE Conf. Intell. Transp.Syst. (ITSC), pp. 1003–1008.

[14]. Greenhalgh J. and Mirmehdi, M. (2015) “Recognizing text-based traffic signs,” IEEE Trans. Intell. Transp. Syst., vol. 16, no. 3, pp. 1360–1369.

[15]. He, K. Zhang, X.Ren, S. and Sun, J. “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR), pp. 770–778.

[16]. Jianming Zhang, Manting Huang, Xiaokang Jin and Xudong Li, (2017) "A real-time chinese traffic sign detection algorithm based on modified YOLOv2", Algorithms, vol. 10, no. 4, pp. 127.40

[17]. Kamal, U. Tonmoy, T. I. Das S. and Hasan, M. K. (2020) "Automatic Traffic Sign Detection and Recognition Using SegU-Net and a Modified Tversky Loss Function With L1-Constraint," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1467-1479.

[18]. Lillo-Castellano, J. Mora-Jiménez, I. M. Figuera-Pozuelo, C. and RojoÁlvarez, J. L. (2015) “Traffic sign segmentation and classification using statistical learning methods,” Neurocomputing, vol. 153, pp. 286–299.

[19]. Mogelmose, A. Trivedi M. M. and Moeslund, T. B. (2012) "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey", IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1484-1497.

[20]. Sun, Y. Ge P. and Liu, D. (2019) "Traffic Sign Detection and Recognition Based on Convolution Neural Network," Chinese Automation Congress (CAC), Hangzhou, China, pp. 2851-2854.

[21]. Szegedy et al., C. (2015) “Going deeper with convolution s,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 1–9.

[22]. Tabernik D. and Skočaj, D. (2020) "Deep Learning for Large-Scale Traffic-Sign Detection and Recognition," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 4, pp. 1427-1440.

[23]. Temel, D. Chen M. H. and AlRegib, G. (2020) "Traffic Sign Detection Under Challenging Conditions: A Deeper Look into Performance Variations and Spectral Characteristics," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 3663-3673.

[24]. Timofte, R. Zimmermann, K. and Van Gool, L. (2014) “Multi-view traffic sign detection, recognition, and 3D localisation,” Mach. Vis. Appl., vol. 25,no. 3, pp. 633–647.

[25]. Yang Y. and Wu, F. (2014) "Real-time traffic sign detection via color probability model and integral channel features" in Pattern Recognition, New York, NY, USA: Springer-Verlag, vol. 484, pp. 545-554.41

[26]. Yang, Y. Luo, H. Xu H. and Wu, F. (2014) "Towards real-time traffic sign detection and classification", Proc. IEEE 17th ITSC, pp. 87-92.

[27]. Zaklouta F. and Stanciulescu, B. (2012) "Real-time traffic -sign recognition using tree classifiers", IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1507-1514.

[28]. Zaklouta F. and Stanciulescu, B. (2014) “Real-time traffic sign recognition in three stages,” Robot. Auto. Syst., vol. 62, no. 1, pp. 16–24.

