Scientific Journal of Silesian University of Technology. Series Transport
Zeszyty Naukowe Politechniki Śląskiej. Seria Transport

Volume 119 **2023**

p-ISBN:: 0209-3324

e-ISBN:: 2450-1549

DOI: https://doi.org/10.20858/sjsutst.2023.119.12

Journal homepage: http://sjsutst.polsl.pl

Article citation information:
Parse, M., Pramod, D. Edge detection technique based on bilateral filtering and iterative threshold selection algorithm and transfer learning for traffic sign recognition. *Scientific Journal of Silesian University of Technology. Series Transport*. 2023, **119**, 199-222. ISSN: 0209-3324. DOI: https://doi.org/10.20858/sjsutst.2023.119.12.

Milind PARSE[1], Dhanya PRAMOD[2]

EDGE DETECTION TECHNIQUE BASED ON BILATERAL FILTERING AND ITERATIVE THRESHOLD SELECTION ALGORITHM AND TRANSFER LEARNING FOR TRAFFIC SIGN RECOGNITION

Summary. The traffic sign identification and recognition system (TSIRS) is an essential component for autonomous vehicles to succeed. The TSIRS helps to collect and provide helpful information for autonomous driving systems. The information may include limits on speed, directions for driving, signs to stop or lower the speed, and many more essential things for safe driving. Recently, incidents have been reported regarding autonomous vehicle crashes due to traffic sign identification and recognition system failures. The TSIRS fails to recognize the traffic signs in challenging conditions such as skewed signboards, scratches on traffic symbols, discontinuous or damaged traffic symbols, etc. These challenging conditions are presented for various reasons, such as accidents, storms, artificial damage, etc. Such traffic signs contain an ample amount of noise, because of which traffic sign identification and recognition become a challenging task for automated TSIRS systems. The proposed method in this paper addresses these challenges. The sign edge is a helpful feature for the recognition of traffic signs. A novel traffic sign edge detection algorithm is introduced based on bilateral filtering with adaptive

---

[1] Symbiosis International (Deemed University) (SIU), Pune, India. Email: parsemv@gmail.com. ORCID: https://orcid.org/0000-0002-3946-2291
[2] Symbiosis Centre for Information Technology (SCIT), Symbiosis International (Deemed University), Pune, India. Email: director@scit.edu. ORCID: https://orcid.org/0000-0003-3451-9794

thresholding and varying aperture size that effectively detects the edges from such noisy images. The proposed edge detection algorithm and transfer learning is used to train the Convolutional Neural Network (CNN) models and recognize the traffic signs. The performance of the proposed method is evaluated and compared with existing edge detection methods. The results show that the proposed algorithm achieves optimal Mean Square Error (MSE) and Root Mean Square Error (RMSE) error rates and has a better Signal to Noise Ratio (SNR) and Peak Signal to Noise Ratio (PSNR) ratio than the traditional edge detection algorithms. Furthermore, the precision rate, recall rate, and F1 scores are evaluated for the CNN models. With the German Traffic Sign Benchmark database (GTSRB), the proposed algorithm and Inception V3 CNN model gives promising results when it receives the edge-detected images for training and testing.

Keywords: bilateral filtering, edge detection, transfer learning, traffic sign identification and recognition

1. INTRODUCTION

Traffic signs play a significant role in creating a secure travel environment and controlling the flow of vehicles running on the road worldwide. Traffic signs are the visual representation of regulations, warnings, and information, which is beneficial for drivers to drive cars safely on roads [1]. Traffic signs have different shapes and provide some vital information to drivers. The commonly used forms are circles, triangles, hexagons, squares containing numbers, text or images, or icons [1, 2]. Over the past decade, powering the automotive industry with artificial intelligence-enabled technologies has been gaining the attention of academicians and industries. Increased awareness of road safety and the global inclination towards adapting more intelligent technology have encouraged researchers to focus on automated solutions for recognizing traffic signs under different challenging conditions and developing appropriate and timely responses according to the situation at hand to avoid road accidents and save people's lives. Even future smart cars can take the input from TSRIS and adjust the driving to prevent fatalities on the road [3-5]. Image processing involves several steps that are carried out to extract meaningful information from the digital images for analysis. Typically, these steps are acquiring, importing, manipulating, and analyzing the image. When an image is processed, its results could be helpful information about parameters, features, or a new resulting image. The resulting image or information helps build specific practical industrial applications [6]. The edge in an image is a continuous set of pixels. It is also denoted using the change in pixel intensity across two disjoint regions. Edge detection is a segmentation method. With the help of brightness, sharpness, or pixel intensity changes, one can classify or identify continuous or discontinuous regions in the image [7]. Over the past years, researchers have developed several edge detection algorithms/operators/methods like Robert, Sobel, Prewitt, and Canny. These methods fall into two categories: Gradient and Gaussian [8-10]. An image may contain noise introduced for various reasons such as lighting conditions, dust on the camera, images taken during rain, objects in the background, etc. The different types of noise in an image are Gaussian, poison, impulse, salt and pepper, and speckle noise. Thus, edge detection in images becomes a more demanding and challenging job [4]. However, images can be denoised using filters and a few other techniques. Nevertheless, excessive filtering for denoising images can reduce the edge strength [11, 12]. In the literature, many methods are available that accurately detect edges in the image. A fast edge computing algorithm introduced for the first time was Robert's algorithm. To detect the sharp edges, Canny algorithm is developed and most widely used in image processing research applications [8-10, 13, 14]. In the literature, numerous studies are available that employ neural network-based classifiers to recognize the traffic sign. The proposed method can be used for feature extraction in conjunction with neural network-based traffic sign detection. Edge images are proved to produce better results as compared to the scenario where unprocessed images are provided as input to the classification model. The popular machine learning library, Keras has various pretrained models which are trained on ImageNet dataset having 1000 classes of images. In this study researchers have used two such pretrained models, VGG-16 and Inception V3 for the performance evaluation. Weights of VGG-16 and Inception V3 models are fine-tuned for traffic sign recognition task using transfer learning technique. Transfer learning has gained the attention of researchers in recently as it saves the time required to train and validate the dataset. The significant contributions of this study include identifying the suitability of existing techniques to detect the edges of images that contain skewed signboards, scratches on traffic symbols, and deformed or discontinuous traffic signs in Indian traffic sign images. An image dataset is created by capturing traffic sign images using a mobile phone camera. A new technique is implemented based on bilateral filtering, adaptive thresholding, and varying aperture size. Then the proposed algorithm is combined with two different CNN models, and the results obtained on the GTSRB dataset are presented and evaluated. Also, the proposed edge detection method's performance is evaluated and compared with existing techniques on quality metric parameters.

2. LITERATURE REVIEW

This section describes the popular edge detection techniques and relevant work available in the literature.

2.1. Edge detection technique

The edge detection techniques identify the true and false edges along the boundary of the image. The Gradient and Second derivative operators are the two types in which all the edge detection algorithms are classified [8-10]. Some of the extensively used edge detection are as follows:

2.1.1. Roberts edge detection operator

Lawrence Robert, in 1963 introduced Roberts cross operator for edge detection. It is a differential operator that approximates an image's gradient via discrete differentiation, which is accomplished by calculating the sum of the squares of the differences between diagonally adjacent pixels. The Roberts edge detection operator quickly calculates a 2-D spatial gradient of an input image in a straightforward. It highlights the strong spatial gradient zones that represent the corresponding edges. This operator takes grayscale images as input and produces grayscale edge image output. The magnitude of the input image's spatial gradient at a particular position is represented by the pixel value at each place in the output image. Unfortunately, it is susceptible to noise and inaccurate edge detection [8-10, 13].

2.1.2. Sobel edge detection

The Sobel edge detection method is a widely used technique where image processing is done separately in the X and Y direction. Then a new image is formed that represents the sum of the edges in X and Y directions in the processed image. Commonly, this method encounters an issue of noise in the final processed image. To reduce the noise in the image, one can use an averaging filter for smoothening the image and then apply the Sobel operator again and compare the differences [8-10, 13].

2.1.3. The Prewitt edge detection

Judith M. S. Prewitt developed the Prewitt operator to detect an object's edge in an image. Prewitt operator detects horizontal edges along the x-axis and vertical edges on the y-axis. An edge is detected whenever there is a change in the intensity of the pixel; hence, differentiation is used to calculate the edge. The local maxima or minima represent the edge in the Prewitt mask result. The first order and second order derivatives in the Prewitt masks have the following properties:

- Maximum weight means more chances of the detection of edge.
- Both (+ and -) signs should be present in the mask.
- The sum of the mask values must be equal to zero.

Prewitt operator provides us with two masks, one for detecting edges in the horizontal direction and another for detecting edges in a vertical order [8-10, 13].

2.1.4. Canny edge detection

It is a popular edge-detection method today because it is robust and flexible. The algorithm itself follows a three-stage process for extracting edges from an image. Then, add image blurring, a necessary pre-processing step to reduce noise. This makes it a four-stage process, including Noise Reduction, Calculating the Intensity Gradient of the Image, Suppression of False Edges, and Hysteresis Thresholding [8-10, 13].

2.2. Relevant works

Yu Z., Feng C., Liu M.Y., and Ramalingam S. (2017) addressed the category awareness problem in semantic edge detection. Unlike traditional frameworks, which use multiple classes, the authors developed a learning framework based on multiple labels that improve edge detection. The proposed CASENet architecture improves DSN architecture. The feature extraction layer replaces the bottom-side classification module. Supervision is imposed at the top layer in the network. Finally, sliced concatenation is replaced by shared concatenation [15].

Liu Y. and et al., (2017) presented a CNN-based richer convolutional features (RCF) framework that uses a hierarchy of image features for edge detection. Here, any size image is taken as input, and the method produces the same size output image with detected edges. This framework utilizes fine details, semantic features, and fine details of images. The RCF framework is also suitable for other computer vision problems, as it provides efficient and accurate results. Furthermore, when applied to classical segmentation, the RCF offers promising results [16].

Lakhani K., Minocha B., & Gugnani N. (2016) analyzed and evaluated edge extraction techniques on dental X-ray images. The authors addressed the problem of misaligned images. Sobel and Prewitt's results showed a notable difference between the original and processed images when the authors applied a Gaussian filter. Similarly, smoothening and sharpening images help diagnose dental diseases and medical X-ray images [17].

Srujana P., Priyanka J., Patnaikuni V.S., & Vejendla N. (2021) in their research work, used Sobel, Prewitt, Roberts, Log, and Canny operators to find the edges of images. The authors used building and flower images for edge detection using all methods. The authors found the Canny edge detector as a suitable method for detection when the results of all of these methods were compared using the parameters like *SNR*, *PSNR*, and *MSE* [18].

Halder A., Bhattacharya P., Kundu A. (2019) used Richardson's extrapolation technique to detect the edge and quantify the edge strength of neighbouring pixels. The extrapolation is used for numerical analysis and to compute the convergence rate. This technique is also used to identify the respective edges of a binary image [19].

Srinithyee S.K., Srivarsha E., Priyadharsini R., & Beulah A. (2021) came up with an augmented edge detection technique that skipped the smoothing process to reduce the computational time. In the proposed method, Sobel operator is modified, and the Kernal is made up of a fractional calculus-based mask instead of the classical one. As a result, the algorithm can work with any image and resist noise. The order of the fractional calculus decides the quality of the output image. It is measured in Pratt's Figure of merit [20].

Carmelo M., Pierce S.G., Summan R. (2019) introduced a novel filtering method based on Spatial FFT (Fast Fourier Transform) for edge reconstruction and a boundary point detection algorithm. The proposed FFT-based edge reconstruction helps eliminate the polynomial curve fitting problem. Independent of threshold values, the Boundary Points Detection (BPD) algorithm detects boundary points using the maximum number of points on the region of interest in boundary points and internal points. The proposed algorithm provides satisfactory results when less than 75% noise [21].

Zheng Z., Zha B., Yuan H., Xuchen Y., Gao Y., and Zhang H. (2020) have implemented an improved model based on grey pixel values in neighboring pixels and a grey prediction method to remove noise and edge localization problems. The model employs 24 directions on the image edge to extract texture features of the edge in the image. To obtain the enhanced edge, the model replaces the values of pixels in the original image with the predicted value of the grey model. Through the global iteration method, the model selects the adaptive threshold and then removes the unwanted points and blur in the image [22].

Shi J., Jin H., Xiao Z. (2020) proposed a fusion of 'polarimetric constant false alarm rate' and 'weighted gradient-based detector to remove speckle noises and detect weak or strong edges in synthetic aperture radar (SAR) images. The Constant False Alarm Rate (CFAR) detector based on KK distribution is proposed to detect weak images, and the weighted gradient-based detector detects the changes in the intensity of strong pixels across heterogeneous areas [23].

Xizhen S., Wei Z., Yiling G., and Shengyang Y. (2021) came up with a solution to the problems associated with the Canny operator, such as poor adaptive ability and image edge determination in noisy images. First, the authors de-noise the images by improving the filtering method and computing the gradient amplitude using a 4-direction template. Then, Otsu's inter-class algorithm combines the image blocks to obtain the low and high threshold values. Finally, the experiment uses leaf images with complex backgrounds [24].

Bausys R., Kazakeviciute-Januskeviciene G., Cavallaro F., Usovaite A. (2020) worked on solving the task of detecting and monitoring the region of interest from satellite images. The proposed method adaptively selects an edge detection algorithm based on a decision matrix.

Also, the authors used the multi-criteria decision-making method. Finally, the authors compared the results of the several edge detection operators, and the Canny edge detector was more accurate and applicable for most of the images. However, it is difficult to predict which edge detector is more appropriate for a particular image content [25].

Dhillon D., and Chouhan, R. (2022) discussed a unique way of finding the best-connected edges and noise removal from the input image. The authors modified the Canny operator that accepts the same parameters but provides better-connected edges using the Stochastic Resonance threshold and window mapping. The proposed method has a limitation. It cannot work without a threshold value; however, the authors suggest using automatic thresholding techniques like Otsu's method. The algorithm performs well on the Barcelona Images for Perceptual Edge Detection Dataset (BIPED) benchmarking dataset [26].

How D.N.T., Sahari K.S.M., Hou Y.C., and Basubeit O.G.S. (2019) applied transfer learning to the Malaysian Traffic Sign dataset, which contained five different classes using the different deep CNN pre-trained model. For most deep CNN models, the authors got an accuracy above 90%. Among all the models, the highest accuracy was obtained by the DenseNet169 pre-trained model, which is 98.33% [30].

Palavanchu S. (2021) has explored a few transfer learning models in the Keras library to detect, recognize and classify the GTSRB traffic signs. The author achieved 95.04% accuracy and 0.2311 loss value for Xception network compared to InceptionV3 Networks, Residual Networks ResNet50, VGG-16, and EfficientNetB0 models [31].

2.3. Limitations of existing edge detection techniques

Roberts's edge detection method is sensitive to noise because it uses a small kernel, and unless the edges are sharp, its detection of edges is very poor. The Soble operator produces inaccurate results with an increase in noise, and hence degrades the edge magnitude. The Prewitt method is also noise sensitive and cannot accurately detect edges with thin and smooth. However, most research proves that Canny edge detection is better than the other three methods. Still, it has greater computational complexity, higher time consumption, and is sometimes inaccurate in edge detection [8-10, 13].

3. PROPOSED WORK

The primary objective was to find the suitability of existing edge detection techniques on the images that present challenges in detecting the traffic symbol, such as skewed signboards, scratches on traffic symbols, and discontinuous or damaged traffic symbols. Initially, the four popular edge detection techniques, viz. Roberts, Sobel, Prewitt, and Canny are implemented and tested [8-10, 13]. A novel Edge detection technique based on bilateral filtering [27-29] with adaptive thresholding [6] and aperture size is proposed. The algorithm contains several steps which are discussed in subsequent sections. First, the algorithm makes the slide adjustment to select the aperture size, the maximum, and the lower threshold to get the object edges. The aperture size and threshold values are iterated until the continuous edge of the symbol is obtained. The results are presented, and the performance of the proposed algorithm with the four existing algorithms is evaluated using SNR, PSNR, MSE, and RMSE parameters [14]. Then the proposed algorithm is used to pre-process the German Traffic Sign Benchmark dataset and using transfer learning with Inception V3 and VGG-16 the traffic sign recognition task is carried out. The image classification results are compared against F1 score, Precision and Recall metrics.

3.1. Dataset and methods

For this research, an image dataset was created that contains 1000 images of traffic signboards captured using a mobile phone. These images are randomly captured using a mobile phone on roads, presenting many challenges in recognizing the traffic symbol [4]. Some challenges include varying lighting conditions, color variance, contrast, occlusion by objects like poles, electricity lines, trees, leaves, improper shape, etc. [4, 29]. This dataset is mainly used to test the proposed bilateral filtering-based edge detection algorithm.

Secondly, the proposed algorithm and transfer learning on VGG-16 and InceptionV3 applied to the German Traffic Sign Benchmark dataset. The German Traffic Sign Benchmark dataset has 50,000 images consist of 43 categories of traffic symbols and is widely used by researchers worldwide for traffic sign recognition tasks. The German Traffic Sign Benchmark dataset is used to find how the proposed edge detection algorithm works on such a standard dataset.

3.2. Proposed edge detection method

The proposed method is based on bilateral filtering, adaptive thresholding, and aperture size. The explanation of important concepts involved in the proposed method is discussed below.

3.2.1. Reducing noise using bilateral filtering

The edge detection process is noise sensitive and requires the image to be smoothened. Image convolution and Gaussian Kernel are widely used for smoothening the images [27]. The size of the Kernel is responsible for deciding blur visibility [28-29]. The blur visibility decreases with lesser kernel size. The Gaussian filter with kernel size $(2k+1)\times(2k+1)$ is given by equation (1).

$$H_{ij} = \frac{1}{2\pi\sigma^2}\, exp\left(-\frac{\left(i-(k+1)\right)^2+(j-(k+1))^2}{2\pi\sigma^2}\right);\ 1 \le i, j \le (2k+1) \tag{1}$$

Using Gaussian filters helps to reduce noise to some extent, but they still present a challenge in detecting object edges accurately. Whereas, the bilateral filter has a spatial kernel and range kernel, which are more effective in noise removal [24]. In bilateral filtering, the, spatial kernel uses the Gaussian function given in equation 1 to smoothen the image. The spatial kernel describes the distance dependence between the image pixels (Euclidean distance). In contrast, the Range Kernel gives the intensity similarity between two image pixels [26-27].

Thus, the spatial kernel defines the spatial proximity measurement. The spatial kernel ($P\sigma s$) is given by equation (2).

$$P_{\sigma s} = exp\ (-(\|a-b\|^2)2\sigma^2 s) \tag{2}$$

While the Range Kernel ($K\sigma r$) defines the weights of pixels with the intensity difference in size pixel under consideration with the image's centre pixel, each pixel range kernel is given by the equation (3).

$$K_{\sigma r} = exp\ (-(|I(a)-I(b)|^2)/2\sigma^2 s\ ) \tag{3}$$

Thus, the Bilateral Filter BFb can be defined using equation (4) and the normalization factor in equation (5)

$$BF_b = \sum_{a\ b\ \in N(p)} K_{\sigma s}\ ((\|a-b\|)\ K_{\sigma r}\ (|I_a - I_b|)\ I_b \tag{4}$$

$$F_n = \Sigma_{\ b} \in N\ (p)\ K_{\sigma s}\ ((\|a-b\|)\ K(|I_a - I_b|) \tag{5}$$

Where,

I is a Grey level image
N(p) represents neighbouring pixel p
Ia and *Ib* give the pixel intensities
Fn is the factor of normalization that ensures pixel weights sum is 1.

3.2.2. L2 gradient

The L2 Gradient is calculated to find pixel intensity and edge direction in the image. Edge connectivity is decided by the change in the intensity of pixels. The bilateral filter is applied to highlight the pixel intensity changes along the horizontal X-axis and vertical Y-axis [27]. Image smoothing is achieved using the Spatial kernel and Range kernel as per equation (2) and equation (3). The magnitude of the Gradient is given by equation (6), and slope θ is given by equation (7).

$$|G| = \sqrt{I_x^2 + I_y^2} \tag{6}$$

$$\theta(x,y) = arctan(\frac{I_y}{I_x}) \tag{7}$$

The resulting image may contain a few thick edges, non-max suppression is used to make them thinner. In addition, the gradient intensity levels are non-uniform, meaning that they may be between 0 and 255. Therefore, the resulting image should have the same intensity across the edges to form a connected edge. To overcome the non-uniformity of intensity levels, double thresholding is used [27-29].

3.2.3. Double thresholding

Some level of intensity for an image pixel is defined using thresholding. Mainly, it is used to separate the edge pixels from the background pixels. Several types of thresholding techniques are available in image processing [6]. To overcome the problem of the non-uniform pixel intensity, double thresholding technique is more effective, and hence it is used in this experiment. In double thresholding, strong, weak, and irrelevant image pixels are found:

- Strong pixels have higher intensity and they contribute to edge formation.
- The pixels whose intensity values are not high enough to consider as strong and not too small enough to consider as non-relevant to form the edge are regarded as weak pixels.
- Pixels that are not strong or not weak are considered non-relevant pixels, and these do not contribute to edge formation.

Thus, in double thresholding, the following are determined:

- Decide the value for the high threshold to determine the strong pixels, i.e., the pixels having a higher intensity value than the high threshold value.
- Decide the value for the low threshold to determine the non-relevant pixels, i.e., the pixels having a lower intensity value than the low threshold value.
- The pixels having intensity values between high threshold and low threshold are considered as weak pixels.

Thus, there are two threshold characters *ST*= High threshold and *WT*=Low threshold. To get the edge region when the pixel's grey intensity value is greater than *ST*, i.e., it represents a strong edge pixel. A non-edge area is obtained when the pixel's grey intensity level is less than *WT*, i.e., it represents the weak edge pixel. The neighbouring pixel's grey intensities are considered to determine the strong edge or weak edge pixel when the grey intensity value lies between *ST* and *WT*. The result of double thresholding produces an image with two-pixel intensity values, i.e., strong and weak.

3.2.4. Iterating threshold values

To ensure the intensity levels, threshold values are iterated. Then the L2 Gradient is calculated for the iterated values of the lower threshold and upper threshold each time. For example, the lower threshold from 100 pixels to 700 pixels is repeated and the value of the upper threshold from 200 pixels to 500 pixels with an iteration of 10 pixels, concerning aperture, sizes 3, 5, and 7 each time. In general, determining the optimal threshold value becomes difficult with images containing noise. To reduce the noise interference, threshold is selected by iterating the values for the aperture sizes [33]. Suppose the original image with noise is given by equation (8)

$$I\ (p,q) = f(p,q) \tag{8}$$

then the algorithm has the following specific steps.

3.3. Algorithm

The specific steps in the algorithm are listed below.

(1) Apply the bilateral filter on the input image to smoothen the image so that sharp edges are detected and preserved, which reduces the false edge detection probability.
(2) Find the L2 Gradient and slope using equations (6) and (7).
(3) Segment the image into two parts, I1 and I2, using a set threshold; to get the image parts as shown in equations (9) and (10).

$$I\_1\ (p,q)=f\_1\ (p,q) \tag{9}$$

$$I\_2\ (p,q)=f\_2\ (p,q) \tag{10}$$

(4) Obtain the grey values for I1 and I2 using equations (11) and (12).

$$E\{I\_1\ (p,q)\}=E\{f\_1\ (p,q)+e(p,q)\} \tag{11}$$

$$E\{I\_2\ (p,q)\}=E\{f\_2\ (p,q)+e(p,q)\} \tag{12}$$

(5) Calculate the iterative loop. The initial threshold value is given by equation (13).

$$T = \frac{G\ max + G_{min}}{2} \tag{13}$$

Where,

T represents the initial value of threshold
Gmax is maximum grey level value
Gmin is minimum grey level value

(6) For '*N*' iterations, calculate *ST* and *WT*. Divide the image into *ST* and *WT* such that *ST* is greater than the initial threshold and *WT* is less than the initial value of the threshold, which is given by equations (14) and (15).

$$ST=\{f(p,q)|f(p,q)>T\} \tag{14}$$

$$WT=\{f(p,q)|f(p,q)<T\} \tag{15}$$

(7) Calculate the upper (*Tu*) and lower (*Tl*) threshold values using equations (16) and (17).

$$T_u = \frac{\Sigma f(p,j) \geq T(i,j)}{\Sigma f(p,j) \geq T^{N0}(i,j)} \tag{16}$$

$$T_l = \frac{\Sigma f(p,j) < T(i,j)}{\Sigma f(p,j) < T^{N1}(i,j)} \tag{17}$$

Where *f(p,j)* gives the Grey level value of image *I(i,j)* for the pixel points *N_0(i,j)* and *N_1(i,j)* that satisfies the following conditions given in equations (18) and (19).

$$N_0(i,j)= \begin{cases} 1, I(i,j) \geq T \\ 0, otherwise \end{cases} \tag{18}$$

$$N_1(i,j)= \begin{cases} 1, I(i,j) < T \\ 0, otherwise \end{cases} \tag{19}$$

(8) The iteration continues until the conditions are satisfied and achieve the initial value of the threshold. Once the iteration terminates, the optimal double threshold values *Tu* and *Tl* are obtained.

4. RESULTS

The effectiveness of the proposed edge detection algorithm is evaluated by comparing its output with that of other popular edge detection techniques, namely Prewitt , Sobel, Roberts and Canny. The proposed edge detection algorithm is then combined with VGG-16 and Inception V3 models to evaluate the improvement resulting in final traffic sign classification activity.

4.1. Edge detection using the proposed algorithm

A dataset of 1000 images was created which were captured using a mobile phone camera. The major focus of this work is to detect the edges of traffic symbols on signboard images in which the traffic signboard is skewed, scratched and the traffic sign or board is deformed. Also, most of these images contain trees or other objects in the background. Figure 1. (a), (b), (c), and (d) shows a few images that are used as input for the proposed algorithm.

Figure 2 (a-d) to Figure 6 (a-d) shows the results of the existing edge detection algorithms and the proposed traffic sign edge detection algorithm. The Prewitt and Sobel methods behave differently with the curved edges and noise. Both methods are less effective in handling the noise, background objects, and curved edges. Comparatively, Roberts and Canny Edge detectors can effectively detect the traffic symbol edges. Also, both methods do not deal with background noise effectively. The proposed method is more effective in handling the background noise and detecting the edges of traffic symbols in traffic sign images of signboards having scratches on the symbol, deformed and Skewed signboards.

(a)Scratches on Sign Board	(b)Skewed Sign Board	(c)Skewed Sign Board	(d) Deformed Sign Board

Fig 1. Input Image – Indian Traffic Signs (a-d)

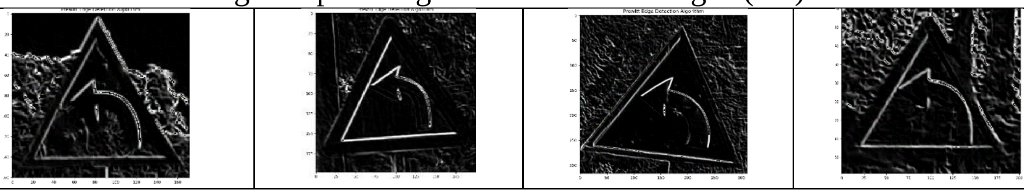


Fig 2. The output of the Prewitt Method (a-d)

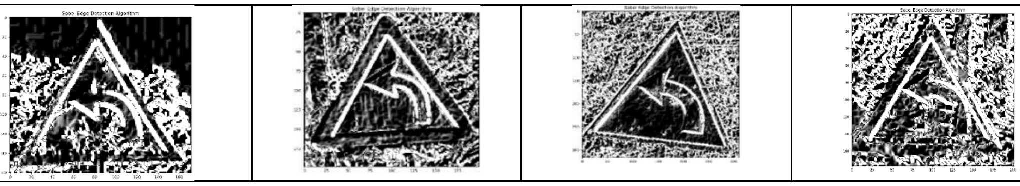


Fig 3. The output of the Sobel Method (a-d)

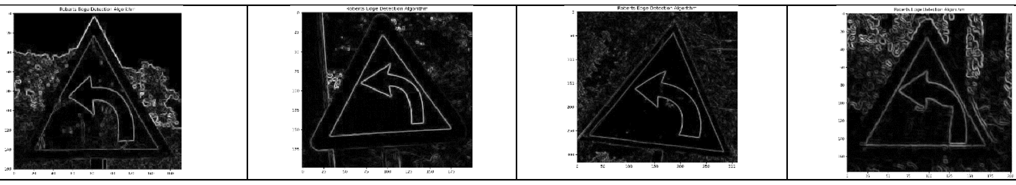


Fig 4. The output of Roberts Method (a-d)

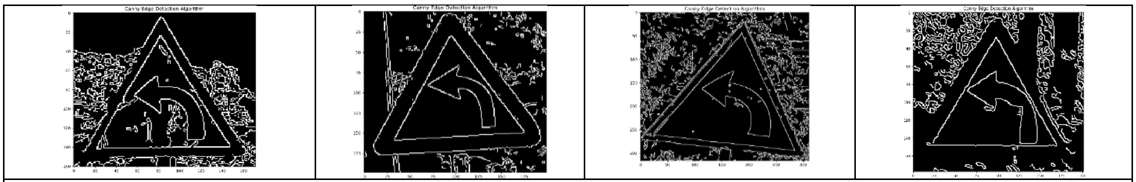


Fig 5. The output of the Canny Method (a-d)

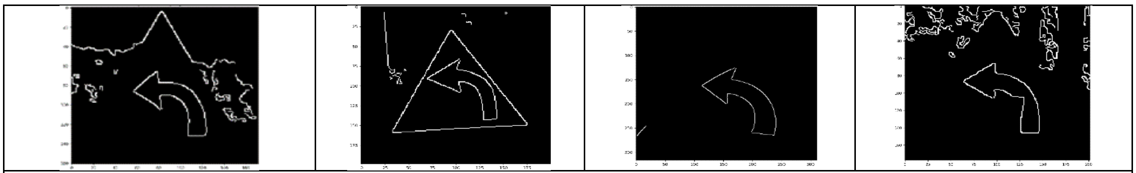


Fig 6. The output of the proposed method (a-d)

4.2. Combining proposed edge detection technique with CNN for traffic sign recognition

The task of traffic sign recognition is carried out using the transfer learning method. Transfer learning allows using a pre-trained model to solve a similar problem by certain fine-tuning parameters across the model layers and learning from new data. The images in GTSRB are categorized across 43 classes and resized to 224 x 224 pixels. All data is randomly divided into the training set and testing with the ratio of 80:20, i.e., 80% for training and 20% for testing. The training set is used for model training and parameter learning. The model is tested in the training process, and the network fine-tuning is performed according to the model test results. The test set is used to test the model's identification capability and the model's generalization ability. In this work, the two pre-trained CNN models VGG-16 and Inception V3 are used. The fully-connected layers are fine-tuned, and the output layer is reconfigured for the traffic sign recognition task on the same CNN models.

Tab. 1

Results before edge detection

Model	Training accuracy	Validation accuracy	Testing accuracy
VGG-16	93.04%	93.53%	91.78%
Inception V3	96.81%	98.81%	98.67%

Later, the input images are pre-processed using a proposed edge detection algorithm to transform the original images into edged images, the pre-processed images are provided as input to the CNN models. Weights of the convolutional layers are pre-trained on ImageNet. The fully-connected layers are fine-tuned, and the output layer is reconfigured for the traffic sign recognition task on the same CNN models; the following results are obtained.

Tab. 2

Results after edge detection

Model	Training accuracy	Validation accuracy	Testing accuracy
VGG-16	96.36%	97.21%	94.32%
Inception V3	99.40%	99.03%	99.49%

Comparing the results in Table 1 and Table 2 shows that the same model gets higher training and testing accuracy if the original images are replaced with the edge images. Though edge images are obtained from original images, they give different information to the Deep CNNs model. More specifically, it is observed that the edge images have more than 99% accuracy because they focus on the inner edge and shape of the object, which is more distinguishable. The below section describes the results of the Inception V3 model in detail. Even in the case of VGG-16 model, when the edge detection algorithm is applied in pre-processing the model performs better and achieves higher accuracy as compared to the model which takes the GTSRB images directly without pre-processing.

4.2.1. Inception V3 implementation without edge detection

As shown in Figure 7, there are 43 different classes of traffic signs in the German traffic sign benchmark dataset. As shown in Figure 8, it is observed that during the training, the loss continues to decrease, and accuracy continues to increase. Similarly, the loss continues to fall after each epoch, and validation accuracy continues to grow after each epoch when the Inception V3 model is trained and validated using the images in the GSTB dataset. The training accuracy of 96.81% is achieved after 10 epochs, with a loss of 0.66. at the same time, the validation accuracy is 98.81%, with a loss of 0.16. Figure 9(a-b) shows the respective graphs for Training and Validation loss, Training and Validation Accuracy on Inception V3 model without edge detection.

Comparing the results in Table 1 and Table 2 shows that the same model gets higher training and testing accuracy if the original images are replaced with the edge images. Though edge images are obtained from original images, they give different information to the Deep CNNs model. More specifically, it is observed that the edge images have more than 99% accuracy because they focus on the inner edge and shape of the object, which is more distinguishable.
The below section describes the results of the Inception V3 model in detail. Even in the case of VGG-16 model, when the edge detection algorithm is applied in pre-processing the model performs better and achieves higher accuracy as compared to the model which takes the GTSRB images directly without pre-processing.

Fig. 7. Input image classes of GSTB dataset

```
Epoch 1/10
269/269 [========================] - 2394s 9s/step  - loss: 4.9019 - accuracy: 0.7647 - val_loss: 2.1001 - val_accuracy: 0.8988
Epoch 2/10
269/269 [========================] - 69s 256ms/step - loss: 1.9076 - accuracy: 0.8312 - val_loss: 1.8756 - val_accuracy: 0.8656
Epoch 3/10
269/269 [========================] - 70s 260ms/step - loss: 1.4123 - accuracy: 0.8979 - val_loss: 1.8056 - val_accuracy: 0.9159
Epoch 4/10
269/269 [========================] - 70s 261ms/step - loss: 0.9860 - accuracy: 0.9386 - val_loss: 0.9656 - val_accuracy: 0.9328
Epoch 5/10
269/269 [========================] - 67s 250ms/step - loss: 0.9140 - accuracy: 0.9399 - val_loss: 0.6934 - val_accuracy: 0.9384
Epoch 6/10
269/269 [========================] - 67s 249ms/step - loss: 0.8932 - accuracy: 0.9479 - val_loss: 0.5277 - val_accuracy: 0.9450
Epoch 7/10
269/269 [========================] - 67s 250ms/step - loss: 0.9547 - accuracy: 0.9467 - val_loss: 0.4137 - val_accuracy: 0.9489
Epoch 8/10
269/269 [========================] - 67s 249ms/step - loss: 0.8979 - accuracy: 0.9495 - val_loss: 0.3187 - val_accuracy: 0.9756
Epoch 9/10
269/269 [========================] - 69s 257ms/step - loss: 0.8890 - accuracy: 0.9565 - val_loss: 0.2089 - val_accuracy: 0.9829
Epoch 10/10
269/269 [========================] - 69s 255ms/step - loss: 0.6614 - accuracy: 0.9681 - val_loss: 0.1092 - val_accuracy: 0.9881
```

Fig. 8. Training and validation loss, training and validation accuracy on Inception V3 model without edge detection

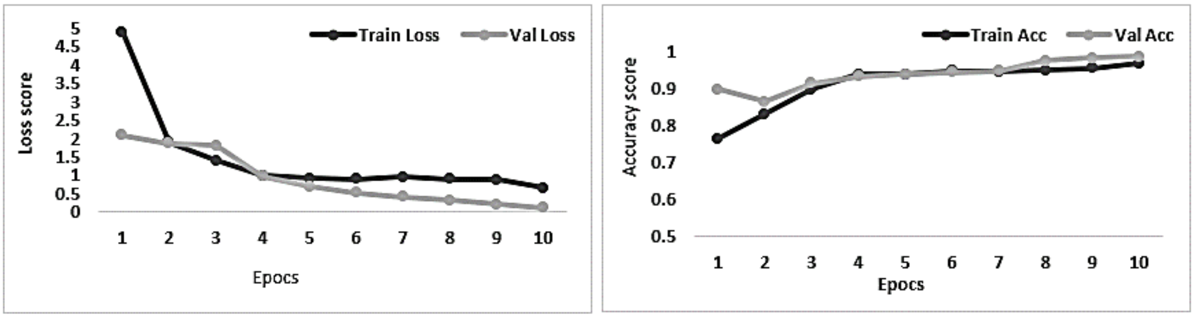
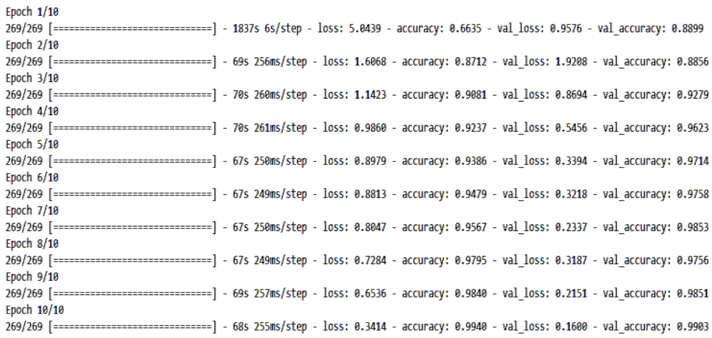


Fig. 9. Training and validation loss, training and validation accuracy on inception V3 model without edge detection

4.2.2. Inception V3 implementation with edge detection

Figure 10 shows the edge detected images. All the images of the German traffic sign benchmark dataset are first pre-processed using the proposed edge detection algorithm. As shown in Figure 11, it is observed that during the training, the loss continues to decrease, and accuracy continues to increase. Similarly, the loss continues to fall after each epoch, and validation accuracy continues to grow after each epoch during training and validation of the Inception V3 model using the images in the GSTB dataset. The training accuracy achieved is 99.40% after 10 epochs, with a loss of 0.34. at the same time, the validation accuracy is 99.03%, with a loss of 0.16. Figure 12(a-b) shows the respective graphs. For the test dataset, the optimal test loss is 0.11 and test accuracy is 99.49% as shown in Figure 13. Figure 14(a-b) shows the graphs for training loss, test loss, training accuracy and test accuracy. The model accuracy improved when the edge detected images were input to the model during training, validation, and testing.

Fig. 10. Input image classes of GSTB dataset with edge detected



```
Epoch 1/10
269/269 [==============================] - 1837s 6s/step - loss: 5.0439 - accuracy: 0.6635 - val_loss: 0.9576 - val_accuracy: 0.8899
Epoch 2/10
269/269 [==============================] - 69s 256ms/step - loss: 1.6068 - accuracy: 0.8712 - val_loss: 1.9208 - val_accuracy: 0.8856
Epoch 3/10
269/269 [==============================] - 70s 260ms/step - loss: 1.1423 - accuracy: 0.9081 - val_loss: 0.8694 - val_accuracy: 0.9279
Epoch 4/10
269/269 [==============================] - 70s 261ms/step - loss: 0.9860 - accuracy: 0.9237 - val_loss: 0.5456 - val_accuracy: 0.9623
Epoch 5/10
269/269 [==============================] - 67s 250ms/step - loss: 0.8979 - accuracy: 0.9386 - val_loss: 0.3394 - val_accuracy: 0.9714
Epoch 6/10
269/269 [==============================] - 67s 249ms/step - loss: 0.8813 - accuracy: 0.9479 - val_loss: 0.3218 - val_accuracy: 0.9758
Epoch 7/10
269/269 [==============================] - 67s 250ms/step - loss: 0.8047 - accuracy: 0.9567 - val_loss: 0.2337 - val_accuracy: 0.9853
Epoch 8/10
269/269 [==============================] - 67s 249ms/step - loss: 0.7284 - accuracy: 0.9795 - val_loss: 0.3187 - val_accuracy: 0.9756
Epoch 9/10
269/269 [==============================] - 69s 257ms/step - loss: 0.6536 - accuracy: 0.9840 - val_loss: 0.2151 - val_accuracy: 0.9851
Epoch 10/10
269/269 [==============================] - 68s 255ms/step - loss: 0.3414 - accuracy: 0.9940 - val_loss: 0.1600 - val_accuracy: 0.9903
```

Fig. 11. Training and validation loss, training and validation accuracy of inception V3 model with edge detection

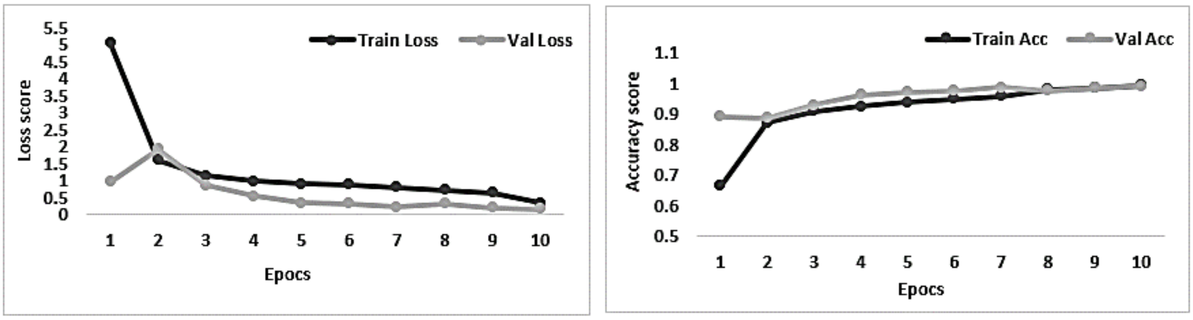


Fig. 12. Training and validation loss, training and validation accuracy of inception V3 model with edge detection

```
Epoch 1/10
269/269 [==============================] - 1341s 5s/step - loss: 5.0430 - accuracy: 0.6498 - test_loss: 1.4676 - test_accuracy: 0.8899
Epoch 2/10
269/269 [==============================] - 59s 220ms/step - loss: 1.3770 - accuracy: 0.8867 - test_loss: 0.6255 - test_accuracy: 0.9395
Epoch 3/10
269/269 [==============================] - 61s 226ms/step - loss: 1.2441 - accuracy: 0.9109 - test_loss: 0.3786 - test_accuracy: 0.9612
Epoch 4/10
269/269 [==============================] - 63s 233ms/step - loss: 0.9372 - accuracy: 0.9330 - test_loss: 0.3694 - test_accuracy: 0.9667
Epoch 5/10
269/269 [==============================] - 60s 223ms/step - loss: 1.0181 - accuracy: 0.9347 - test_loss: 0.3708 - test_accuracy: 0.9729
Epoch 6/10
269/269 [==============================] - 62s 232ms/step - loss: 0.8448 - accuracy: 0.9470 - test_loss: 0.3996 - test_accuracy: 0.9752
Epoch 7/10
269/269 [==============================] - 59s 220ms/step - loss: 0.6640 - accuracy: 0.9523 - test_loss: 0.4818 - test_accuracy: 0.9651
Epoch 8/10
269/269 [==============================] - 59s 220ms/step - loss: 0.5184 - accuracy: 0.9726 - test_loss: 0.2871 - test_accuracy: 0.9853
Epoch 9/10
269/269 [==============================] - 62s 230ms/step - loss: 0.3677 - accuracy: 0.9884 - test_loss: 0.2469 - test_accuracy: 0.9806
Epoch 10/10
269/269 [==============================] - 59s 219ms/step - loss: 0.2151 - accuracy: 0.9943 - test_loss: 0.1157 - test_accuracy: 0.9949
```

Fig. 13. Train and test loss, train test accuracy of inception V3 model with edge detection

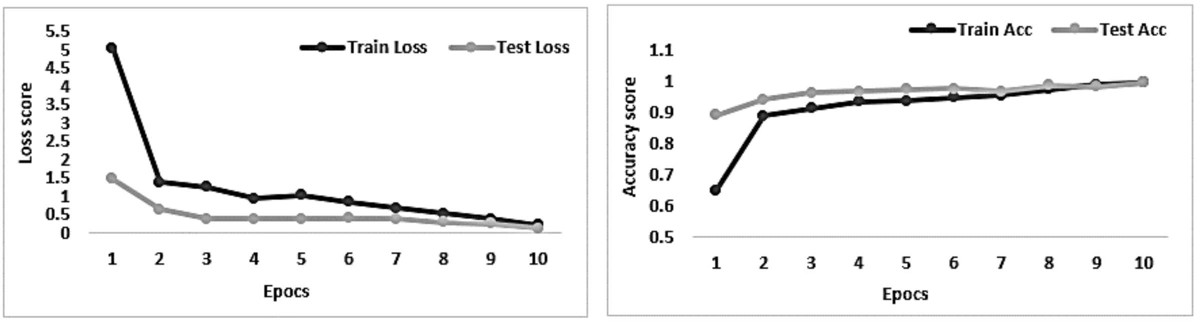


Fig. 14. Train and test loss, train accuracy and testing accuracy of inception V3 model with edge detection

5. PERFORMANCE EVALUATION

The performance efficiency of edge detection of the proposed Traffic Sign Edge Detection algorithm is evaluated on parameters like Signal to Noise Ratio, Peak Signal to Noise Ratio, and Mean Square Error.

5.1. Mean square error (*MSE*)

MSE is one of the popular quality assessment metrics used to compare the image edge detection quality. The smaller value of *MSE* indicates that the edge detection method performs better. It can be calculated by evaluating the difference between the original image and the edge detected image, as given in equation (20).

$$MSE = [f1(x,y)-f2(x,y)] \tag{20}$$

Where *f1* is the original image and *f2* is the edge detected image.

5.2. Root mean square error (*RMSE*)

RMSE is also a quality assessment metric similar to *MSE*. The following formula shown in equation (21) is used to calculate the value of *RMSE*.

$$RMSE = \sqrt{MSE} \tag{21}$$

5.3. Signal to noise ratio (*SNR*)

SNR is a quality metric used to calculate the false switching of pixels while estimating an edge. It is widely used to compare the performance of different methods of segmentation and edge detection. It represents the level of noise in detected edges [35, 36]. The higher *SNR* value indicates the sharpness of the edge. To evaluate the proposed method, *SNR* is calculated using equation (22).

$$SNR = \left[\frac{\sum_{x=1}^{M}\sum_{y=1}^{N} f2(x,y)^2}{\sum_{x=1}^{M}\sum_{y=1}^{N}(f2(x,y)-f1(x,y))^2}\right] \tag{22}$$

Where *f1* is the original image and *f2* is the edge detected image. The image size is $M \times N$. The rows and columns are given by x and y respectively.

5.4. Peak signal to noise ratio (*PSNR*)

PSNR describes how the noise affects the pixel quality. It calculates the ratio between the max value of a pixel and the noise. It is given on a logarithmic decibel scale. The higher value of *PSNR* generally indicates that the error is less but, in some cases, like edge detection, the lesser value of *PSNR* shows the accurate edge detection [36, 37]. Equation (23) gives the value of *PSNR*.

$$PSNR = 10 \log \left[\frac{255^2}{MSE}\right] \tag{23}$$

Table 3 to Table 6 shows the statistical comparison of the proposed method with existing traditional methods using the noise and edge detection quality assessment metrics. The proposed method is more accurate and has a good quality of edge detection as compared to traditional methods for the input image that contains scratches on the signboard (Table 3), skewed signboard images (Table 4, and Table 5), and the deformed signboard image (Table 6).

Tab. 3

Performance Comparison for Input Image in Fig. 1(a)

Criteria	Prewitt	Sobel	Robert	Canny	Proposed (TSED)
SNR	3.8874	3.2169	4.7832	8.3726	**9.0142**
PSNR	2.9054	2.9843	2.7987	2.4568	**2.9122**
MSE	3.8352	4.4786	3.3303	3.0215	**2.2113**
RMSE	1.9583	2.1162	1.8249	1.7382	**1.4870**

Tab. 4

Performance comparison for input image in Fig. 1(b)

Criteria	Prewitt	Sobel	Robert	Canny	Proposed (TSED)
SNR	3.3832	3.6432	4.8934	8.8735	**9.1823**
PSNR	2.9364	2.8954	2.9987	3.6943	**4.2342**
MSE	3.4023	3.9237	3.0012	2.7926	**2.4134**
RMSE	1.8445	1.9808	1.7323	1.6711	**1.5535**

Tab. 5

Performance comparison for input image in Fig. 1(c)

Criteria	Prewitt	Sobel	Robert	Canny	Proposed (TSED)
SNR	3.2834	3.5792	4.4892	8.6253	**9.4844**
PSNR	2.8247	2.7962	2.8246	3.4956	**4.0122**
MSE	3.0624	3.4642	3.3389	4.0062	**1.9834**
RMSE	1.7499	1.8612	1.8272	2.0015	**1.4083**

Tab. 6

Performance comparison for input image in Fig. 1(d)

Criteria	Prewitt	Sobel	Robert	Canny	Proposed (TSED)
SNR	3.7832	3.7653	4.6946	8.7673	**9.3472**
PSNR	3.0313	3.2372	3.1034	3.5532	**4.3032**
MSE	3.1023	4.9145	3.4312	4.0216	**2.7134**
RMSE	1.7613	2.2168	1.8523	2.0053	**1.6472**

As shown in Table 3 to Table 6 for an input image, the parameter values help determine the edge's quality detected in the image. The proposed traffic sign edge detection algorithm achieved a higher quality of edge detection than the other algorithms. By this comparative analysis of the result, it can be concluded that the proposed traffic sign edge detection algorithm is the best and most optimal algorithm concerning all other existing algorithms.

Impact of proposed novel edge detection technique on VGG-16 and Inception V3 model's performance are compared using Precision Rate, Recall Rate, and F1 Score metrics.

5.5. Precision rate

Precision is used to compute the model's classification accuracy in terms of positive samples classified either correctly or incorrectly. It is the ratio of the number of samples classified as True positive to the total number of positively classified samples, as given in equation (24). Higher the precision, better the model performance. Ideal model would have a perfect score 1.0

$$P = TP/(TP+FP) \tag{24}$$

5.6. Recall rate

The model's ability to detect positive samples is called the recall rate. The higher recall rate indicates that the model can detect more positive samples. It is given as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. Equation (25) is used to calculate the recall rate.

$$R = TP/(TP+FN) \tag{25}$$

5.7. F1 score

The F1 Score combines the precision and recall rate in one single metric. It is given by the equation (26). F1 score is the harmonic mean of precision and recall. F1 score 1 represents that model perfectly classified all observations correctly.

$$F1\ Score = 2 * \{ (P*R) / (P+R) \} \tag{26}$$

Figure 15-17 show that the models perform better when the input image is edge detected. It is evident that for the classes 16th, 17th, 22nd, 29th, and 30th, the Recall rate is apparent, and the Precision rate is higher.

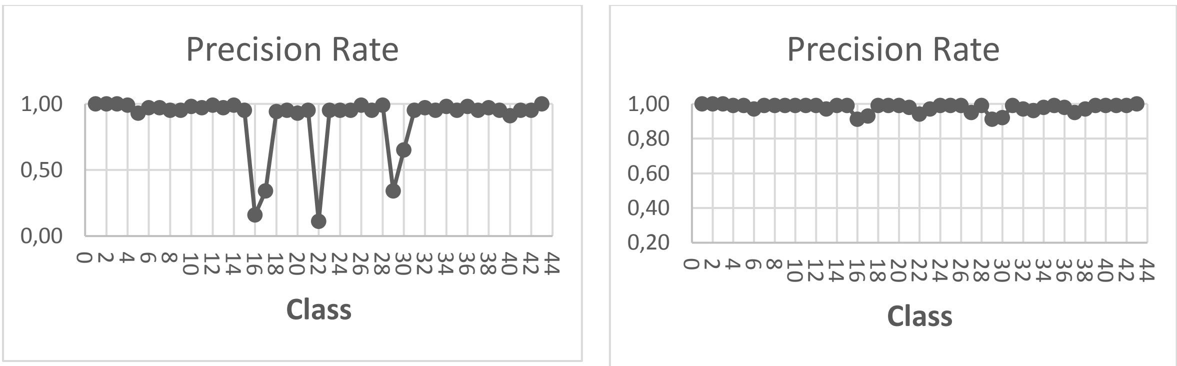


Fig. 15. Precision rate without edge detection (a); precision rate with edge detection (b)

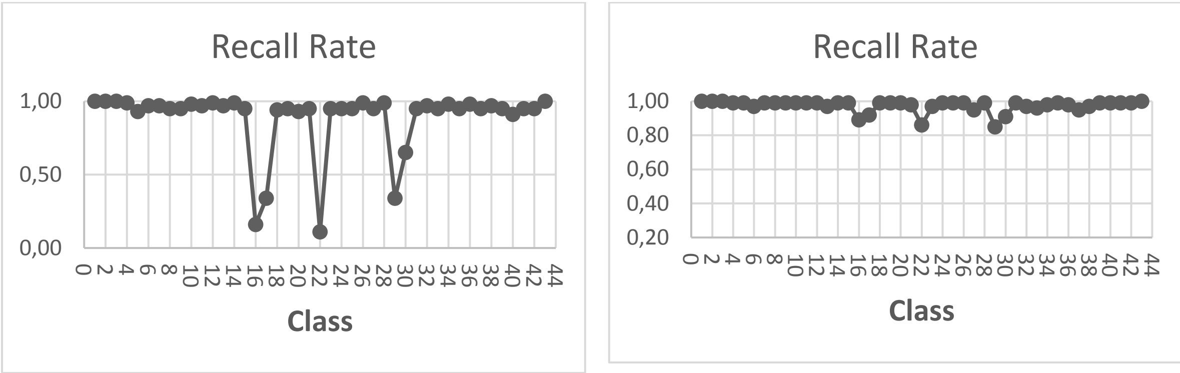


Fig. 16. Recall rate without edge detection (a); recall rate with edge detection (b)

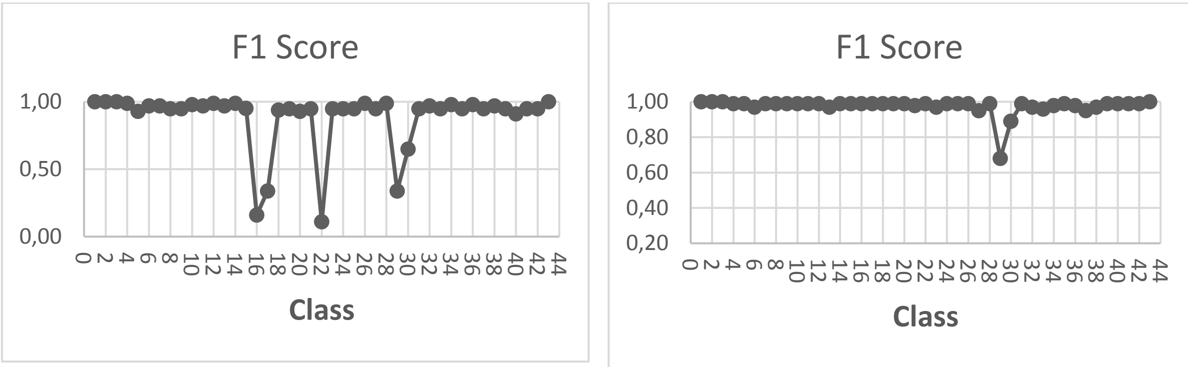


Fig. 17. F1 Score without edge detection (a); F1 score with edge detection (b)

6. COMPARATIVE ANALYSIS

The proposed approach combines edge detection with transfer learning for traffic sign recognition and identification. Table 7 below summarizes some similar works. It is observed that the proposed edge detection algorithm helps to improve the accuracy of traffic sign recognition.

Tab. 7

Performance comparison of previous similar studies

Reference paper	Dataset	Classes	Edge detection	Model	Training set (%)	Validation set (%)	Testing set (%)
[31]	GTSRB	43	No	InceptionV3	-	70.74	-
[42]	GTSRB	43	No	InceptionV3	-	-	96.03
[31]	GTSRB	43	No	VGG-16	-	30.61	-
[43]	GTSRB	43	No	VGG-16	-	-	74.5
[44]	GTSRB	43	No	IVGG-16	-	-	99.00
This study	GTSRB	43	No	VGG-16	93.04	93.53	91.78
This study	GTSRB	43	No	InceptionV3	96.81	98.81	98.67
This study	GTSRB	43	Yes	VGG-16	96.36	97.21	94.32
This study	GTSRB	43	Yes	InceptionV3	99.40	99.03	99.49

7. CONCLUSION AND FUTURE SCOPE

This paper discussed why traffic sign recognition and classification are crucial today. A novel edge detection technique is implemented, and the proposed algorithm results are presented. The comparative analysis of the performance metric parameters shows that the proposed traffic sign edge detection algorithm provides promising results for edge detection in challenging conditions. Thresholding is the critical factor for the extraction of visual features in images. The proposed method defines three levels of thresholding: fixed, floating, and ideal. In fixed and ideal thresholding conditions, the Canny edge detector performs well compared to Robert, Prewitt, and Sobel edge detection algorithms. However, the proposed traffic sign edge detection algorithm, derived from canny edge detection, produces more promising results with a floating thresholding range from 100 pixels to 600 pixels. The results clearly show that the proposed algorithm considers the actual images for the symbol. When integrated with pre-trained CNN models, the proposed traffic sign edge detection algorithm gives better training, validation, and testing accuracy. Two pre-trained models, VGG-16 and Inception V3, were used in the study, out of which Inception V3 is a top performer with 99.49% test accuracy. The results show that the proposed algorithm achieves optimal *MSE* and *RMSE* error rates and has a better *SNR* and *PSNR* ratio than the traditional edge detection algorithms. There is a scope to address a few challenges discussed in section two in the future. In the future, the researcher will aim to implement the proposed algorithm to detect and recognize the traffic signs from skewed, scratched, and deformed traffic symbols in real-time using the other CNN models.

References

1. Liu Chunsheng, Shuang Li, Faliang Chang, Yinhai Wang. 2019. „Machine vision based traffic sign detection methods: Review, analyses and perspectives." *IEEE Access* 7: 86578-86596. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2924947.
2. Alam Altaf, Zainul Abdin Jaffery. 2020. „Indian traffic sign detection and recognition. *International Journal of Intelligent Transportation Systems Research* 18(1): 98-112. DOI: https://doi.org/10.1007/s13177-019-00178-1.
3. Winkle Thomas. 2022. "Findings from Traffic Accident Analysis". *Product Development within Artificial Intelligence, Ethics and Legal Risk*. Springer Vieweg, Wiesbaden. P. 7-43. ISBN: 978-3-658-34293-7. DOI: https://doi.org/10.1007/978-3-658-34293-7
4. Temel Dogancan, Min-Hung Chen, Ghassan AlRegib. 2019. „Traffic sign detection under challenging conditions: A deeper look into performance variations and spectral characteristics." *IEEE Transactions on Intelligent Transportation Systems* 21(9): 3663-3673. DOI: https://doi.org/10.1109/TITS.2019.2931429.
5. Barodi Anass, Abdrrahim Bajit, Mohammed Benbrahim, Ahmed Tamtaoui. 2020. „An Enhanced Approach in Detecting Object Applied to Automotive Traffic Roads Signs." In: *2020 IEEE 6th International Conference on Optimization and Applications (ICOA).* P. 1-6. IEEE 2020. 20-21 April 2020. Beni Mellal, Morocco. ISBN: 978-1-7281-6655-1. DOI: https://doi.org/10.1109/ICOA49421.2020.9094457.
6. Sundararajan Duraisamy. 2017. *Digital image processing: a signal processing and algorithmic approach*. Springer International Publishing. ISBN: 978-981-10-6113-4.
7. Ghosh Chinmoy, Suman Majumder, Sangram Ray, Shrayasi Datta, Satyendra Nath Mandal. 2020. „Different EDGE detection techniques: a review." *Electronic Systems and Intelligent Computing*: 885-898. DOI: https://doi.org/10.1007/978-981-15-7031-5_84.

8. Amer Ghassan Mahmoud Husien, Ahmed Mohamed Abushaala. 2015. „Edge detection methods.” In: *2015 2nd World Symposium on Web Applications and Networking (WSWAN).* P. 1-7. IEEE, 2015. 21-23 March 2015. Sousse, Tunisia. ISBN: 978-1-4799-8172-4. DOI: https://doi.org/10.1109/WSWAN.2015.7210349.
9. Yousaf Rehan Mehmood, Hafiz Adnan Habib, Hussain Dawood, and Sidra Shafiq. 2018. „A comparative study of various edge detection methods.” In: *2018 14th International Conference on Computational Intelligence and Security (CIS).* P. 96-99. IEEE, 2018. 16-19 November 2018. Hangzhou, China. ISBN: 978-1-7281-0170-5. DOI: https://doi.org/10.1109/CIS2018.2018.00029.
10. Shah Bickey Kumar, Vansh Kedia, Rohan Raut, Sakil Ansari, Anshul Shroff. 2020. „Evaluation and Comparative Study of Edge Detection Techniques.” *IOSR Journal of Computer Engineering* 22(5): 6-15. DOI: https://doi.org/10.9790/0661-2205030615. ISSN: 2278-8727.
11. Deshpande Abhinav V., Ramani Kannan, M. Monica Subashini. 2018. „Study of Various Image De-Noising Methods Used for the Purpose of Traffic Sign Board Recognition in an Intelligent Advanced Driver Assistance System.” In: *2018 International Conference on Intelligent and Advanced System (ICIAS).* P. 1-6. IEEE, 2018. 13-14 August 2018. Kuala Lumpur, Malaysia. ISBN: 978-1-5386-7270-9. DOI: https://doi.org/10.1109/ICIAS.2018.8540630.
12. Kumar Arvind, Sartaj Singh Sodhi. 2020. „Comparative analysis of gaussian filter, median filter and denoise autoenocoder.” In: *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)*. P. 45-51. IEEE, 2020. 12-14 March 2020. New Delhi, India. DOI: 10.23919/INDIACom49435.2020.9083712.
13. Kumar Sunil, Amit Kumar Upadhyay, Preeti Dubey, and Sudeep Varshney. 2021. “Comparative analysis for Edge Detection Techniques.” In: *2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*. P. 675-681. IEEE, 2021. 19-20 February 2021. Greater Noida, India. ISBN: 978-1-7281-8530-9. DOI: 10.1109/ICCCIS51004.2021.9397225.
14. Tariq Nazish, Rostam Affendi Hamzah, Theam Foo Ng, Shir Li Wang, Haidi Ibrahim. 2021 „Quality assessment methods to evaluate the performance of edge detection algorithms for digital image: A systematic literature review.” *IEEE Access*. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3089210.
15. Yu Zhiding, Chen Feng, Ming-Yu Liu, Srikumar Ramalingam. 2017 „Casenet: Deep category-aware semantic edge detection.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR)*. P. 5964-5973. July 2017.
16. Liu Yun, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. 2017. „Richer convolutional features for edge detection.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR)*. P. 3000-3009. July 2017.
17. Lakhani Kanika, Bhawna Minocha, Neeraj Gugnani. 2016. „Analyzing edge detection techniques for feature extraction in dental radiographs.” *Perspectives in Science* 8: 395-398. DOI: https://doi.org/10.1016/j.pisc.2016.04.087.
18. Srujana P., J. Priyanka, Vyss Sudir Patnaikuni, Nancharaiah Vejendla. 2021. „Edge Detection with different Parameters in Digital Image Processing using GUI.” In: *2021 5th International Conference on Computing Methodologies and Communication (ICCMC).* P. 795-802. IEEE, 2021. 08-10 April 2021. Erode, India. ISBN: 978-1-6654-4775-1. DOI: 10.1109/ICCMC51019.2021.9418327.

19. Halder Amiya, Pritam Bhattacharya, Aritra Kundu. 2019. „Edge detection method using Richardson's extrapolation formula." In: *Soft Computing in Data Analytics, Advances in Intelligent Systems and Computing* 758. 727-733. Springer, Singapore. ISBN: 978-981-13-0513-9. DOI: 10.1007/978-981-13-0514-6_69.
20. Srinithyee S.K., E. Srivarsha, R. Priyadharsini, A. Beulah. 2021. „Optimized Image Edge Detection Approach using Fractional Order Calculus." In: *2021 6th International Conference on Communication and Electronics Systems (ICCES)*. P. 1179-1183. IEEE. 08-10 July 2021. Coimbatre, India. ISBN: 978-1-6654-1182-0. DOI: 10.1109/ICCES51350.2021.9489066.
21. Mineo Carmelo, Stephen Gareth Pierce, Rahul Summan. 2019 „Novel algorithms for 3D surface point cloud boundary detection and edge reconstruction." *Journal of Computational Design and Engineering* 6(1): 81-91. DOI: https://doi.org/10.1016/j.jcde.2018.02.001.
22. Zheng Zhen, Bingting Zha, Hailu Yuan, Youshi Xuchen, Yanliang Gao, He Zhang. 2020. „Adaptive edge detection algorithm based on improved grey prediction model." *IEEE Access* 8: 102165-102176. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2999071.
23. Shi Junfei, Haiyan Jin, Zhaolin Xiao. 2020. „A novel hybrid edge detection method for polarimetric SAR images." *IEEE Access* 8: 8974-8991. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2963989
24. Xizhen Shen, Zeng Wei, Guo Yiling, Yin Shengyang. 2021. „Edge detection algorithm of plant leaf image based on improved Canny." In: *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*. P. 342-345. IEEE. 09-11 April 2021. Xi'an, China. ISBN: 978-1-6654-4730-0. DOI: 10.1109/ICSP51882.2021.9408929.
25. Bausys Romualdas, Giruta Kazakeviciute-Januskeviciene, Fausto Cavallaro, Ana Usovaite. 2020. „Algorithm selection for edge detection in satellite images by neutrosophic WASPAS method." *Sustainability* 12(2): 548. DOI: https://doi.org/10.3390/su12020548.
26. Dhillon Deepak, Rajlaxmi Chouhan. 2022. „Enhanced Edge Detection Using SR-Guided Threshold Maneuvering and Window Mapping: Handling Broken Edges and Noisy Structures in Canny Edges." *IEEE Access* 10: 11191-11205. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3145428.
27. Fawwaz I., M. Zarlis, R.F. Rahmat. 2018. „The edge detection enhancement on satellite image using bilateral filter." In: *IOP Conference Series: Materials Science and Engineering* 308(1): 012052. IOP Publishing. 24-25 August 2017, Banda Aceh, Indonesia.
28. Ai Jiaqiu, Ruiming Liu, Bo Tang, Lu Jia, Jinling Zhao, Fang Zhou. 2019. „A refined bilateral filtering algorithm based on adaptively-trimmed-statistics for speckle reduction in SAR imagery." *IEEE Access* 7: 103443-103455. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2931572.
29. Galetto Fernando J., Guang Deng, Mukhalad Al-Nasrawi, Waseem Waheed. "Edge-aware filter based on adaptive patch variance weighted average. 2021." *IEEE Access* 9: 118291-118306. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3106907.
30. How Dickson Neoh Tze, Khairul Salleh Mohamed Sahari, Yew Cheong Hou, Omar Gumaan Saleh Basubeit. 2019. „Recognizing Malaysia traffic signs with pre-trained deep convolutional neural networks." In: *2019 4th International Conference on Control, Robotics and Cybernetics (CRC)*. P. 109-113. IEEE. 27-30 September 2019. Tokyo, Japan. ISBN: 978-1-7281-4621-8. DOI: 10.1109/CRC.2019.00030.

31. Palavanchu Sowjanya. 2021. „Transfer Learning Models for Traffic Sign Recognition System.” *Annals of the Romanian Society for Cell Biology*: 3477-3489.
32. Rajagopal Balaji Ganesh. 2020. „Intelligent traffic analysis system for Indian road conditions.” *International Journal of Information Technology*: 1-13. DOI: https://doi.org/10.1007/s41870-020-00447-3.
33. Ghedia Navneet S., C. H. Vithalani. 2021. „Outdoor object detection for surveillance based on modified GMM and Adaptive Thresholding.” *International Journal of Information Technology* 13(1): 185-193. DOI: https://doi.org/10.1007/s41870-020-00522-9.
34. Pamuji Fandi Yulian, Moch Arief Soeleman. 2020. „Improved Number Detection For Low Resolution Image Using the Canny Algorithm.” In: *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*. P. 638-642. IEEE. 9-20 September 2020. Semarang, Indonesia. ISBN: 978-1-7281-9069-3. DOI: 10.1109/iSemantic50169.2020.9234190.
35. Nadipally Manasa. 2019. „Optimization of methods for image-texture segmentation using ant colony optimization.” In: *Intelligent data analysis for biomedical applications*. P. 21-47. Academic Press. ISBN: 9780128155530. DOI: https://doi.org/10.1016/B978-0-12-815553-0.00002-1.
36. Malarvizhi C., P. Balamurugan. 2019. „Qualitative Analysis Of Various Edge Detection Techniques Applied On Cervical Herniated Spine Images.” *ICTACT Journal of Image and Video Processing* 9(04): 1986-1991. DOI: 10.21917/ijivp.2019.0282.
37. Paul Eben Sophia, J. Anitha. 2019. „Analysis of Transform-Based Compression Techniques for MRI and CT Images.” In: *Intelligent Data Analysis for Biomedical Applications*. P. 103-120. Academic Press. ISBN: 9780128155530. DOI: https://doi.org/10.1016/B978-0-12-815553-0.00005-7.
38. Fitriyah Hurriyatul, Edita Rosana Widasari, Gembong Edhi Setyawan. 2017. „Traffic sign recognition using edge detection and eigen-face: Comparison between with and without color pre-classification based on Hue.” In: *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*. P. 155-158. IEEE. 24-25 November 2017. Malang, Indonesia. ISBN: 978-1-5386-2183-7. DOI: 10.1109/SIET.2017.8304127.
39. Zhu Yanzhao, Wei Qi Yan. 2022. „Traffic sign recognition based on deep learning.” *Multimedia Tools and Applications* 81(13): 17779-17791. DOI: https://doi.org/10.1007/s11042-022-12163-0.
40. Wan Haifeng, Lei Gao, Manman Su, Qinglong You, Hui Qu, Qirun Sun. 2021. „A novel neural network model for traffic sign detection and recognition under extreme conditions.” *Journal of Sensors*. DOI: https://doi.org/10.1155/2021/9984787.
41. Lin Chunmian, Lin Li, Wenting Luo, Kelvin C.P. Wang, Jiangang Guo. 2019. „Transfer learning based traffic sign recognition using inception-v3 model.” *Periodica Polytechnica Transportation Engineering* 47(3): 242-250. DOI: https://doi.org/10.3311/PPtr.11480.
42. Atif Muhammad, Tommaso Zoppi, Mohamad Gharib, Andrea Bondavalli. 2022. „Towards enhancing traffic sign recognition through sliding windows.” *Sensors* 22(7): 2683. DOI: https://doi.org/10.3390/s22072683.
43. Persson Siri. 2018. „Application of the German Traffic Sign Recognition Benchmark on the VGG16 network using transfer learning and bottleneck features in Keras.” *Thesis*. Uppsala universitet, Datalogi. Available at: http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-344672.

44. Kheder Mohammed Qader. 2022. „Improved Traffic Sign Recognition System (Itsrs) for Autonomous Vehicle Based on Deep Convolutional Neural Network." *SSRN*. DOI: http://dx.doi.org/10.2139/ssrn.4135313. Available at: https://ssrn.com/abstract=4135313.

Received 07.12.2022; accepted in revised form 29.03.2023