Multimedia Tools and Applications (2023) 82:13837–13853
https://doi.org/10.1007/s11042-022-14018-0

Number plate recognition from enhanced super-resolution using generative adversarial network

Anwesh Kabiraj[1,2] · Debojyoti Pal[1,3] · Debayan Ganguly[3] · Kingshuk Chatterjee[4] · Sudipta Roy[1]

Received: 15 March 2022 / Revised: 4 July 2022 / Accepted: 23 September 2022 /
Published online: 28 September 2022

Abstract
Identification and recognition of number plate is very difficult from low resolution images due to poor boundary and contrast. Our goal is to identify the digits from a low-quality number plate image correctly, but correct detection was exceedingly difficult in some cases due to the low-resolution image. Another goal of this paper was to upscale the image from a very low resolution to high resolution to recover helpful information to improve the accuracy of number plate detection and recognition. We have used Enhanced- Super-Resolution with Generative Adversarial Network (SRGAN). We modified native Dense Blocks of the Generative Adversarial Network with a Residual in Residual Dense Block model. In addition to Convolutional Neural Networks for thresholding. We also used a Rectified Linear Unit (ReLU) activation layer. The plate image is then used for segmentation using the OCR model for detection and recognizing the characters in the number plates. The Optical character recognition (OCR) model reaches an average accuracy of 84% for high resolution, whereas the accuracy is 4% - 7% for low resolution. The model's accuracy increases with the resolution enhancement of the plate images. ESRGAN provides better enhancement of low-resolution images than SRGAN and Pro-SRGAN, which the OCR model validates. The accuracy significantly increased digit/alphabet detection in the number plate than the original low-resolution image when converted to a high-resolution image using ESRGAN.

✉ Sudipta Roy
sudipta1.roy@jioinstitute.edu.in

Anwesh Kabiraj
anwesh.kabiraj@jioinstitute.edu.in

Debojyoti Pal
debojyoti.pal@jioinstitute.edu.in

Debayan Ganguly
debayan3737@gmail.com

Kingshuk Chatterjee
kingshukchaterjee@gmail.com

Extended author information available on the last page of the article

Keywords Number plate detection · Super-resolution · Optical character recognition · Deep learning · Residual dense block · Structural similarity of images

1 Introduction

Super Resolution (SR) has had an increasing attraction in Industries and research communities for Computer Vision and Imaging. Earlier low-resolution images were converted into high-resolution images using various Linear and Bicubic optimization by pioneers in the Image Industry like Adobe and Sony. With the recent leap in AI and AI-based technologies, Deep Neural Networks have been applied and have brought prosperous results [36]. An increasingly new number of network architecture design and training techniques have continuously improved the results. Enhancing otherwise useless images can be of huge value considering the information that the image carries with it. There have been a lot of works in super-resolution over the past years with the application of Generative Adversarial Networks (GAN) and Neural Networks. However, in our model, we have replaced the batch normalization layer with a residual in a residual dense block, which provides a higher capacity for the adversarial loss used in training the SRGAN model.

A standard method is to train a model on different objects. It automatically applies learned information to enhance it when it finds a low-resolution image of a matching class. However, this method fails in improving the unique features of an image as we have an endless number of objects in our surroundings, and one cannot train a model on every object that possibly exists. Also, keep in mind that such a model adds general information learned from other images. Hence the enhancement done may not be suitable. Researchers extensively used neural networks to enhance each aspect of the image [4], which is not needed as information within the image might be limited to only specific parts and not the whole image. In such cases, a lot of time and computation power is wasted. Rafique et al. use Sobel filter techniques to detect the edges of number plates [28] accurately. Filtering techniques like Maximum Average Correlation Height filter and Log r-theta Mapping methods were used by [20] to classify various categories of automobiles. Optical Character Recognition methods were used to convert characters in number plate images into machine-coded texts [33]. The enhancements done by Amit et al. [1] use OCR software with various pre-processing techniques, increasing the chances of successfully recognizing characters. We used multilevel feed-forward networks like ANN, which classify characters and finally convert them into text format. The algorithm proposed in this paper combines pre-processing techniques like the Sobel filter technique and inverse binary thresholding [18], followed by an OCR model. Then we send it to the RNN + LSTM model for character recognition. We implement these techniques keeping in mind the main objective, i.e., attaining high accuracy in recognizing characters and extracting more information from the number plates. Figure 2 explains in detail the methodology implemented for Number Plate Recognition.

The first work in this field happened as early as 2014 [5]. The proposed SRCNN model performed better than any other existing work on super-resolution at that time. Later, numerous new Network architectures and improvements have made Deep Learning with residual networks, Residual Scaling and Residual Learning. Researchers used different network structures like Deep recursive back Projection [11], Laplacian Pyramid architecture [13] and Residual Dense Networks. Zhang et al. [39] proposed the EDSR model where they replace the batch normalization layer with Residual Blocks to benefit from Residual learning. Residual

Dense Block is very effective in removing Peak Signal to Noise Ratio (PSNR). The tesseract OCR is an open-source system developed by HP open-sourced in 2006. Over the past decade, OCR has helped convert handwritten documents [19] and digitize medieval manuscripts [12], making retrieving important information from handwritten documents and files easier. Researchers have worked on various machine learning approaches like Random Forests (RF) and Decision Tree (DT) [10]. They have tried to combine them with image processing and deep learning [3, 15, 31, 38] architectures for the digitization of handwritten documents and for increasing the accuracy of OCR systems, resulting in the adaptation of cluster computing and GPUs. Apart from the already mentioned applications, OCR techniques have been used in various real-world scenarios such as recognition of consumer service numbers in energy consumption bills [8], recognition of textual information from mobile captured images of receipts [34], developing character recognition systems of various languages [7, 9] and the development of automatic number plate recognition systems [32]. Recent works have focused on the usage of various pre-processing techniques using Haar cascade [27], and graph-based approaches [35] or have implemented Deep CNNs [17] to achieve results as good as 80%. However, these techniques either use specific use-cases as inputs or completely neglect low quality(resolution) images, which is a major obstacle in working with real-world data.

In this paper, we have used ESRGAN coupled with an optical character recognizer to detect alphanumeric characters from number plates of poor quality and recover helpful information. Lately, there has been a lot of work on Automatic Number plate detection. Most of them rely on high-speed cameras or other high spec tech and focus on faster real-time detection of the characters. However, our method is more of a post-process to recover information. We focus on a Deep Learning network to solve the super resolution problem.

The proposed method accurately predicts characters in low-quality (in terms of resolution and noises) number plate images, without using a huge number of training datasets or employing several post-processing techniques. The pre-existing OCR models have used a large training dataset. However, the existing models either exhibit erratic results for low-quality number plates or fail to produce accurate results on a different dataset. The proposed model aims to solve the drawbacks above by introducing a novel ESRGAN for image enhancement and using a novel OCR model for predictions.

2 Method

Dataset We have used a dataset containing 1000 high-resolution images of number plates. We have added 182 real-world images (collected with consent) for testing our model. The images were converted to low-resolution images by a factor of 64 using Bicubic transformation. Then we upscale the low-resolution images using the above framework and compare the upscaled images to original high-resolution images in terms of structural similarity and other metrics.

We additionally tested our model on a Kaggle car license plate dataset [2] containing 433 images and annotated in PASCAL VOC format (a format to store annotations for localizer or Object Detection datasets). We have followed the same method of downscaling the image first and then testing the model on it.

We have two phases for training the generative adversarial network, i.e., training the discriminator and training the generator. The main goal of the generator is to maximize the loss of the discriminator.

The Discriminator loss L_D is

$$L_D = -E_{xr}\left[log\left(D\left(x_r, x_f\right)\right)\right] - E_{xf}\left[log\left(1-D\left(x_f, x_r\right)\right)\right] \quad (1)$$

The Adversarial loss for the generator L_G is as follows

$$L_G = -E_{xr}\left[log\left(1-D\left(x_r, x_f\right)\right)\right] - E_{xf}\left[log\left(D\left(x_f, x_r\right)\right)\right] \quad (2)$$

x_r *and* x_f denotes real image and fake image, respectively, while E_{xr}*and* E_{xf} Represents the operation of taking an average of all real and fake images, respectively.

Summing up, we have

$$min_G max_D V(D,G) = E_{x\sim p_{data(x)}}[log(D(x))] + E_{z\sim p x_z(z)}[log(1-D(G(z)))] \quad (3)$$

Now we pass the LR image through successive Convolutional and ReLU layers. We pass it through the RRDB block to utilize the residual learning for training our adversarial loss function. The discriminator function helps in generating more real textures.

The procedure of the RRDB model begins with the Low-Resolution image(x) as input, where the Convolution layer extracts feature, which we use as input for the RRDB model.

$$F_0 = K_{CNN}(x) \quad (4)$$

F_0 denotes the feature extracted by the convolutional layer denoted by K_{CNN} on LR image (x).

Now, if we have *n* residual in residual dense blocks, then the output of the nth RRDB model can be written as Eq. [5]

$$F_N = K_{RRDB,n}(F_{rrdb-1}) \quad (5)$$

Where $K_{RRDB,\ n}$ = nth RRDB operation, which also includes the operation of the CNN and ReLU layers.

We have used 26 RRDB blocks and ReLU as an activation function in our model. The dense function of the CNN and RRDB blocks is formulated as Eq. [6]

$$F_{n,c} = \sigma\left(\mathrm{W}_{\mathrm{n,c}}\left[F_{n-1}, F_{n,1}, \ldots\ldots, F_{n,c-1}\right]\right) \quad (6)$$

Where σ denotes the activation function.

The Global Residual Learning is

$$F_{GRL} = \sum_1^{26} F_{n,c} \quad (7)$$

Hence the output from the first convolutional layer, in addition to the global residual learning, is

$$F_{total} = F_{GRL} + F_0 \quad (8)$$

A very dense connection between the CNN and RRDB block increases the network capacity used in this method, is shown in Fig. 1.

The LR image is passed through a 5 × 7 Convolution layer and then through the Residual in Residual Dense Block (RRDB) Model, which is much more densely connected and increases the performance output. Replacing the Batch Normalization (BN) layer with the RRDB model helps in the model's Generalization ability. A similar process of residual learning is proposed in [38]. We observe that increasing the number of layers drastically improves the output. We have used 24 RRDB blocks, which provide a much deeper and more

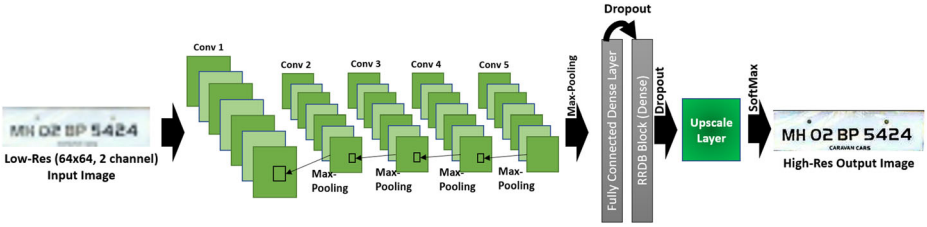

Fig. 1 The overall architecture of the convolutional neural network and residual in a residual dense block

complex structure, as shown in Fig. 1, and the network capacity becomes quite high due to the dense connections.

The RGB plate image is converted to grayscale to optimize the detection of characters by decreasing the number of primary and secondary colours available in the image drastically.

The grayscale image is then denoised by using Gaussian blur (sigma = 5). This pre-processing technique removes noises present in the images due to dust particles in the image scanner, sensor temperature, or other environmental factors impacting the scanner, as shown in Fig. 2. It also makes the edges of the image smoother and cleaner, making the characters more readable.

Inverse binary thresholding, which is the opposite of binary thresholding, is used to enhance the number plate images by obtaining white characters over a black background. This is done by assigning a pixel value less than the threshold value to a maximum value(white). The threshold value is determined using global thresholding method [24]. As shown [24], there are two options to find the threshold. The first is to minimize the within-class variance; the second is to maximize the between-class variance. The general algorithm's pipeline for the between-class variance maximization option can be represented in the following way:

1. calculating intensity level probabilities obtained from the image histogram
2. Initialising the values of $L_D = -E_{xr}[log(D(x_r, x_f))] - E_{xf}[log(1 - D(x_f, x_r))]$ as $\omega_i(0)$, $\mu_i(0)$

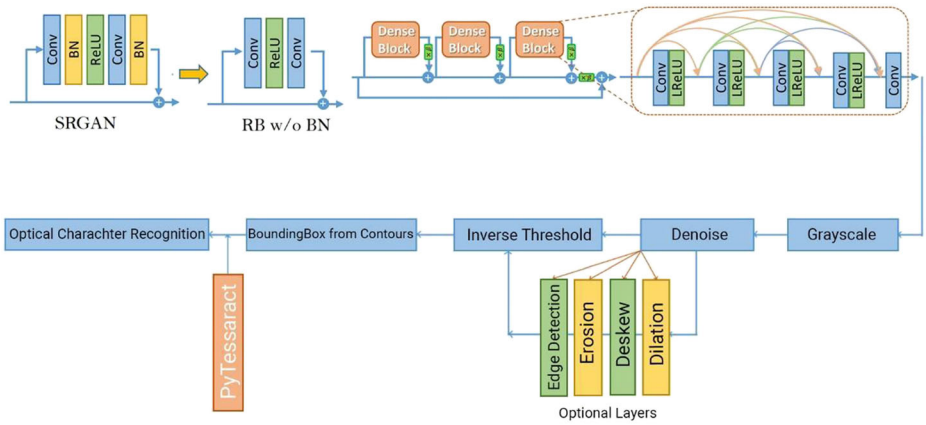

Fig. 2 The diagram of the GAN, residual blocks, image augmentation layers and the optical character recognition model

3. iterating over possible thresholds: t = 0….,max_intensity where max_intensity = 255

- updating the values of ω_i, μ_i, where ω_i is a probability and μ_i is a mean of class i
- calculating the between-class variance value $\sigma_b^2(t)$

4. the final threshold is obtained by maximizing the value of $\sigma_b^2(t)$

ω_i is the probability value of each pixel value of the bimodal clusters using the cluster probability functions expressed as:

$\omega_1(t) = \sum_{i=1}^{t} P$, μ is the mean of cluster probability, and σ is the standard deviation.

Image segmentation is used to locate objects in images. The image segmentation divides the entire plate image into segments that collectively cover the image. It then enables us to extract a set of contours from the image, which helps determine the bounding boxes for the character recognition phase.

The intermediate processes and a step-wise brief description are mentioned in Fig. 3(A). The alphabets refer to the intermediate steps (including pre-processing), and the italic sentences give a brief description of the steps involved in the workflow. Figure 3(B) simultaneously provides visualization for the steps involved in the workflow.

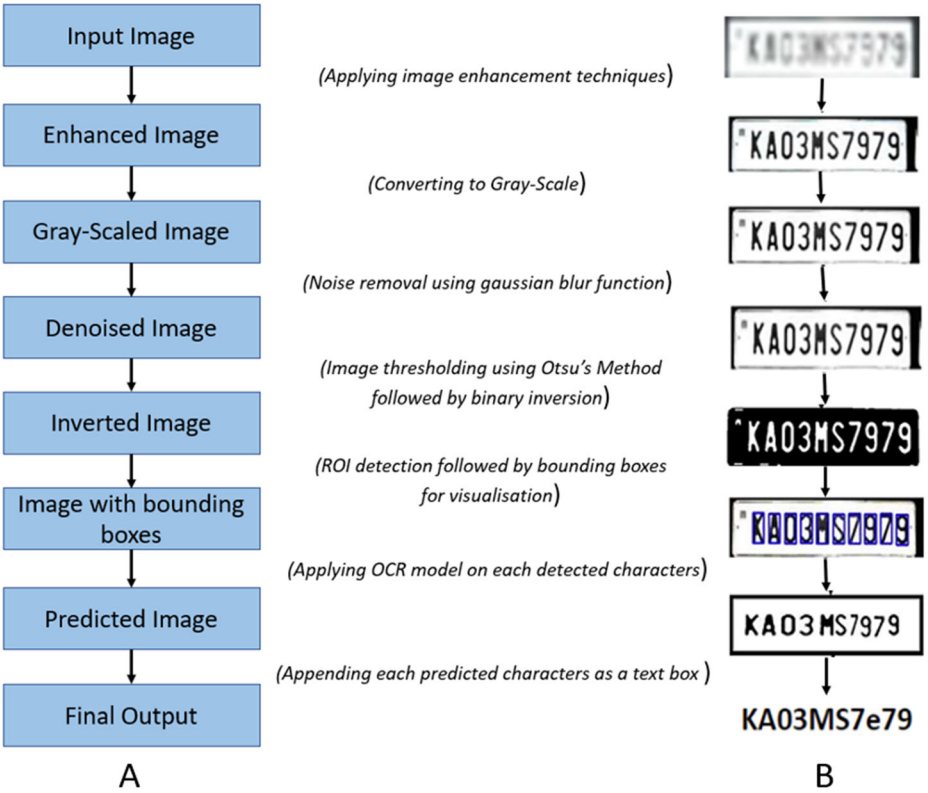

Fig. 3 Flow diagram of the proposed automatic number plate recognition model, A) represents the schematic workflow of the proposed model, B) represents the OCR based number plate recognition workflow results

2.1 OCR model

The proposed model uses an amalgamation of both text detection and text recognition models as an OCR framework, enabling the model to accurately detect and recognise alphanumeric characters in numberplate images. The pre-processed number plate images are passed as input. The boundary of each plate image are detected using edge-detection techniques which include canny-edge detection.The alphanumeric characters on the number plate are detected using contour detection operation (using findContour() function) followed by the formation of bounding boxes around the detected contours (using boundingRect() function). The text recognition operation starts by taking each individually segmented characters as input with the help of automatically generated bounding boxes. This is followed by a two-step process. The first step includes recognition which is performed using a static classifier.Each segmented text in bounding boxes is classified into text boxes (bounding boxes contatining segmented characters)..A dictionary search is performed on each text box to find an appropriate character or series of characters with high confidence in terms of the segmented texts. While the word result is unsatisfactory, the OCR model chops the text boxes with the worst confidence from the dictionary search. The chop points are determined using an algorithm which performs polygonal approximation of concave vertices of the character outline. After exhausting all the chopping possibilities, a best-first search is performed on the segmentation graph which comprises of all possible combinations of thechopped character outlines. At each step in the best-first search, any new combinations are classified and a dictionary search is performed.

Each satisfactory classified character is passed to an adaptive classifier as training data. The training data comprises of a 4-dimensional feature vector. Each 4-dimensional feature consists of x, y-position, direction, length which is derived from each element of the approximation algorithm and clustered to form feature vectors. In the recognition module, the elements of the polygon are chopped into shorter pieces having equal length.This enables the elimination of length dimension from the feature vector.

To reduce the time complexity, a class pruner creates a shortlist of 1–10-character and the 26 letters of alphabets as classes that the unknown might match using a Locality Sensitive Hashing (LSH) approach. In the second stage, the weighted distance d of each feature is calculated from its nearest prototype. The equation for the mentioned calculation is as follows:

$$d = d_{initial}^2 + \omega\theta^2 \tag{9}$$

Where d is the Euclidean distance of the feature coordinates from the prototype line and θ is difference of the angle from the prototype. The feature distance is converted to feature evidence E_f using the following equation:

$$E_f = \frac{1}{kd_f^2} \tag{10}$$

The constant k controls the rate at which the evidence decays with distance. As features are matched to prototypes, the feature evidence E_f, is copied to the prototypes E_p. The sums are

normalized by the number of features and sum of prototype lengths L_p, and the result is converted back into a distance:

$$d_{final} = 1 - \frac{\sum_f E_f + \sum_p E_p}{N_f + \sum_p L_p} \tag{11}$$

The pre-processing steps include denoising and inverse binary thresholding. The pre-processing is done earlier as an attempt to add a new LSTM model. The predicted characters are appended and the final output is generated in text(.txt) format.

2.2 Image resolution enhancement

The batch Normalization layer is used to deal with unstable gradients that occur because of using Neural Networks. It normalizes the features with estimated mean and variance and use the same for the whole training dataset and one drawback of the same is introduction of unpleasant artefacts in the image. We have used an epsilon value of 0.002 (for avoiding getting divided by zero) and momentum value of 0.87. It is used for building the convolutional neural network model. We have used a 5 × 7 Convolutional Neural Network structure. The ConVPool2D has a Rectified Linear Unit as the activation layer. A rectified linear unit is used to perform thresholding operations in the image i.e., setting the value to either black or white for grey pixels. This layer is responsible for improving the texture of the image. Used for single image super resolution using Generative Adversarial Networks. A Perceptual Loss function is used which contains the adversarial loss and content loss. We used a loss function with $\lambda = 6 \times 10^{-3}$ and $\eta = 1 \times 10^{-2}$. This adversarial loss in turn is used to train a discriminator function which then predicts relative realness in place of the actual network. The RRDB model is used in place of batch normalization layer as it provides a higher capacity for training, also using Residual Scaling in place of Batch Normalization facilitates training in a very deep network. We have used 24 RRDB blocks in our model, whereas 16 RRDB blocks have a similar capacity like SRGAN hence ours is a much deeper model. The RRDB model improves Peak Signal to Noise Ratio (PSNR) much better than using the BN layer. We have used Adam optimizer with $\beta 1 = 0.95$, $\beta 2 = 0.99$.

3 Results and discussion

The model was trained on a core i9 processor with NVIDIA RTX A4000 graphics accelerator. Compared different images with different levels of enhancement gives us an idea of how well the model is performing, but how well the model enhances the image compared to its original form is a point of concern. There are many ways of comparing two images; however, Generative Adversarial Networks are only focused on enhancing certain important attributes of the image like sharpening, edge detection etc., rather than the whole image.

Hence all image comparison methods do not give an idea of how well the image has been enhanced to satisfy our particular need, i.e., reading numbers from a number plate. In Table 2, we can find Mean Squared Error, Normalized histogram, Structural Similarity Index [37], as well as recognition accuracy. Among the above-mentioned methods, only the Structural

similarity index and recognition accuracy give us an idea of how well the image has been enhanced to satisfy our aim.

$$SSIM(x,y) = \frac{(2\mu x\mu y + C1)(2\sigma xy + C2)}{\left(\mu_x^2 + \mu_y^2 + C1\right)\left(\sigma_x^2 + \sigma_y^2 + C2\right)} \tag{12}$$

We find structural similarity using Eq. 7. To find out the effectiveness of the proposed model in terms of image enhancement, we calculate accuracy scores using two different methods:

$$Accuracy\_Present = \frac{Number\ of\ characters\ correctly\ predicted}{Number\ of\ characters\ \boldsymbol{present}\ intheimage} \tag{13}$$

$$Accuracy\_Identified = \frac{Number\ of\ characters\ correctly predicted}{Number\ o\ fcharacters\ \boldsymbol{identified}\ intheimage} \tag{14}$$

Since character detection using region of interests (ROIs) detection plays an important part in character recognition by OCR models, the accuracy score obtained by (Eq. 14) gives us an estimate of how well the character recognition part is actually performing. The accuracy score obtained by (Eq. 13) gives us an overall estimate of the performance of the OCR model.

Qualitative evaluation We present a qualitative evaluation of the performance of the proposed OCR with respect to the different rates of enhancement of the number plate images and different enhancement techniques used.

As we observe from Fig. 4, the number of bounding boxes that the model can detect is quite low in case of the original LR-Image. It gradually increases with increase in enhancement. A side-by-side comparison of ProSR also shows that bounding boxes in an image enhanced with our model have much more numbers compared to ProSR.

4 Quantitative evaluation

To test the performance of the enhanced super resolution model we tested it on the car number plate dataset [2] and compared our model in terms of structural similarity metrics that our model achieves. We also plotted the corresponding mean squared error (see Fig. 5). We cannot

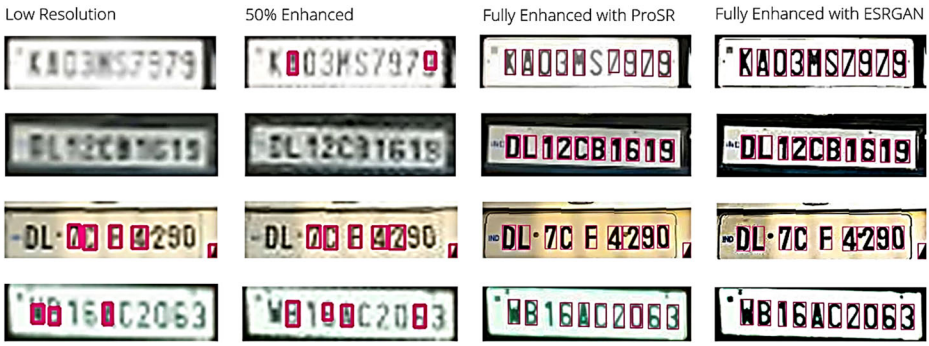

Fig. 4 Comparing the number of bounding boxes detected with respect to the change in rates of enhancement

visualize any relation between the two accuracy metrics however we can observe that structural similarity roughly lies between 76 and 92 (maximum – 92.47 minimum 76.91) while MSE value roughly lies between 74 and 88 (maximum - 88.23, minimum – 74.41).

We present a quantitative evaluation of how well the OCR model performs for bounding box detection and character recognition in terms of enhancement rate. Finally, a table comparing the accuracy rate of various state-of-the-art OCR techniques is provided to further the claim regarding the robustness of the proposed model.

The number of bounding boxes and characters recognized varies vastly from image to image. Some having no bounding boxes to some as high as 6 in low resolution state as we can see from Table 1. Most of the images have bounding boxes in the range of 1 to 3 with an average of 1 character correctly recognized in the low-resolution state. In a 50% enhanced state some images show drastic improvements, while some remain with little or no changes. However, the general trend is an increase in the number of characters recognized. This trend is further proved to be correct from Fig. 4.

In Fig. 6, we have plotted a graph of the percentage accuracy of the OCR model versus the percentage enhancement of the LR image. We have found average accuracy for a batch of images for every 20% increase in enhancement. We find that the accuracy increases over the enhancement of LR images and doesn't show any anomaly. The rise doesn't follow a straight line which further signifies that recognition accuracy rises with the enhancement of the number plate images. Table 2 shows three methods of comparing images, Mean Squared Error (MSE), structural similarity index measure (SSIM), and Normalized histogram. Our proposed framework also outperforms other state-of-the-art methods discussed in [6, 16] in Table 3.

The accuracy scores of the proposed model based on the aforementioned accuracy metrics are plotted as an unstacked bar plot in Fig. 7. The accuracy score mentioned in Eqs. (13 and 14) are represented as blue and green bars respectively. The accuracy score of the OCR model is calculated using the Accuracy_Present method and the accuracy of the detected characters from numberplate images is calculated using the Accuracy_Identified method. From Fig. 7, we can observe an increasing trend in accuracy percentage from 11% to 78% for low to high enhancement rate in images using the OCR model. The reduced accuracy score of OCR model is due to the failure of region of interest(ROI) detection algorithms in detecting characters in number plate images or inaccurate bounding boxes for the detected characters.. The accuracy

Fig. 5 Structural similarity and Mean squared error achieved for 433 images

Table 1 The number of bounding boxes and corresponding characters recognized for different images

Original Image	Bounding Box	Characters Recognized	Enhanced by 50%	Bounding Box	Characters Recognized
test 1	3	0	test 1	5	1
test 2	1	0	test 2	3	0
test 3	0	0	test 3	3	2
test 4	0	0	test 4	0	0
test 5	1	0	test 5	4	4
test 6	2	0	test 6	3	0
test 7	1	0	test 7	5	2
test 8	4	1	test 8	6	4
test 9	0	0	test 9	7	5
test 10	6	6	test 10	9	7

score calculated using Accuracy_Identified method, from Eq. (14) shows an increasing trend from 19% for low resolution images to 82% for high resolution images. This gives an idea of how efficient the proposed character recognition algorithm actually works.

In Fig. 8, the kernel density plot gives us an estimate of the probability distribution of bounding boxes in the original, mid resolution and high-resolution images. It is clearly visible that the high-resolution images have a higher detection rate of bounding boxes with a probability of 53% for 9–10 bounding boxes. The mid resolution images have the highest detection rate of bounding boxes in the range of 3 to 4 with a probability of 25%. The low-resolution images have the lowest bounding box detection rate with a peak around the range of 0–1 bounding boxes with a probability of 26%. Thus, with an improvement in resolution a rapid increase in the number of bounding boxes is observed, therefore an improvement in accuracy in terms of character detection is observed.

The area under the curve between two intervals gives information regarding the probability of a certain number of bounding boxes (between 0 to 10), which can be accurately recognized by the OCR model. The KDE plot along with the histogram plot (in Fig. 6) portrays the

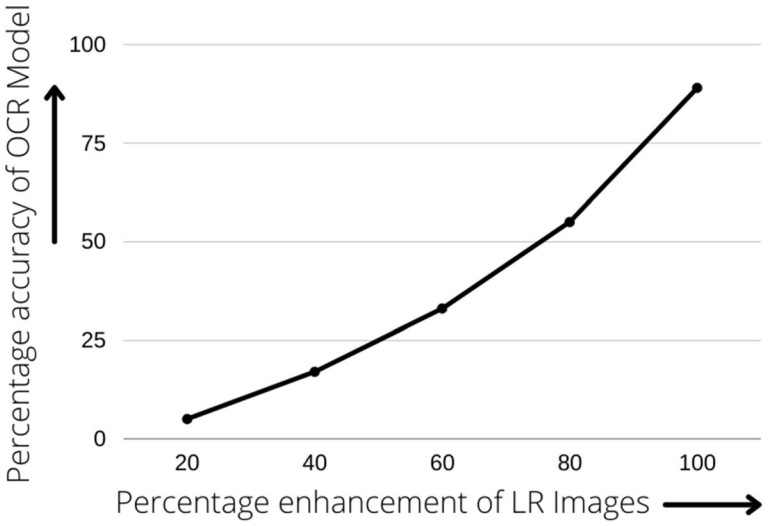

Fig. 6 Graph showing percentage accuracy achieved by the OCR model against different levels of enhancement

Table 2 Various accuracy matrices, like SSIM, MSE, normalized histogram and accuracy

Image	Mean Squared Error	Structural Similarity	Normalized Histogram	Characters recognized correctly	Recognition Accuracy (in %)
test 1	1220	88.94	71.23	9	90
test 2	1192	86.81	73.79	9	90
test 3	1166	85.58	76.70	8	80
test 4	1247	84.96	78.40	8	80
test 5	1121	87.46	73.95	10	100
test 6	1258	87.76	76.15	7	70
test 7	1208	82.23	72.69	8	80
test 8	1119	84.30	78.44	8	80
test 9	1332	88.99	72.94	9	90
test 10	1279	81.80	77.23	10	100

distribution of the number of bounding boxes generated by the model. From this plot we can establish that the OCR model can accurately identify 1 or 2 characters for low-resolution images,3 to 4 characters for mid-resolution images and 9 to 10 characters in a number plate for high-resolution images. This observation can be inferred from the peaks observed in the density plot in ranges 1–2, 3–4 and 9–10 for low, mid and high-resolution images respectively.

The OCR model exhibits the best performance for high resolution images, followed by mid resolution (50% enhanced) images and the original (low resolution) images with respect to model accuracy and bounding boxes. The OCR model reaches an accuracy as high as 84% for high resolution images whereas original and mid resolution images have an accuracy of 18% and 50% respectively. This is due to the increase in the average detection of bounding boxes from 0 to 2 for low resolution original images to 9 or 10 for high resolution images.

A comparative study on the various methods used for Number plate recognition in the recent years have been provided in Table 4. The table compares three widely used methods with the proposed method in terms of the size of the dataset in which the models have been trained as well as the accuracy obtained on the testing dataset.

The proposed OCR model exceeds in average accuracy rates with respect to [14, 21–23, 30] by 3% and 4% respectively as shown in Table 4. Although [29] used comparable method for number plate detection over a training dataset of 900 images, it has an even lower accuracy rate of 73% in comparison to ESRGAN enhanced OCR Model. This proves the effectiveness of the ESRGAN in number plate detection, which works great in tandem with the proposed OCR models in achieving greater accuracy over challenging datasets. Another observation from the table is the difference in training dataset size. Although the model proposed in the paper uses far less training dataset images and involves little to no pre-processing, the proposed model achieves the best accuracy rate of 84% among the comparedoptical character recognition model.

Table 3 Comparing other state of art methods to our method based on SSIM score

Method	SSIM
License Plate Image Reconstruction Based on Generative Adversarial Networks	72.0%
License Plate Image Analysis Empowered by Generative Adversarial Neural Networks (GANs)	80.7%
Our Model	85.8%

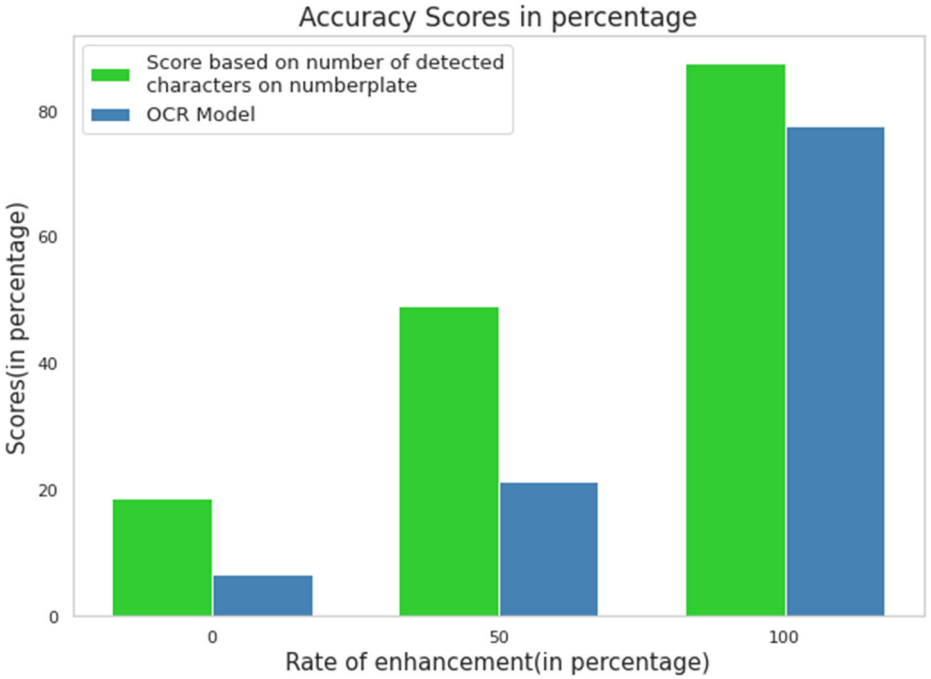

Fig. 7 Unstacked bar plot representing the accuracy scores of the OCR model and accuracy score based on number of characters detected on the number plate images

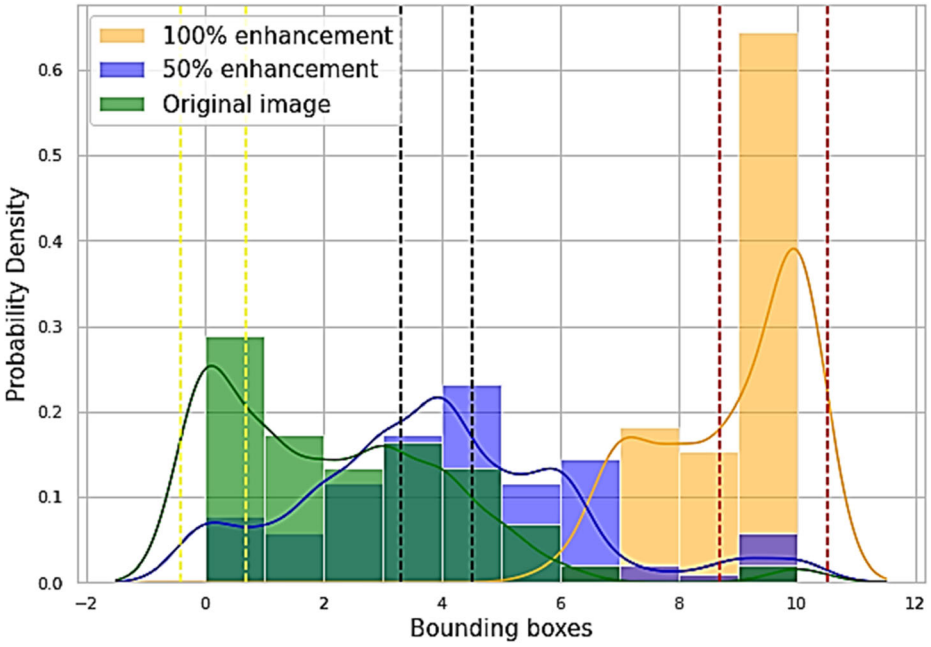

Fig. 8 Kernel density plot representing the probability density of the number of bounding boxes recognized by the OCR model based on the rate of image enhancement

Table 4 Average accuracy rates obtained by OCR based models in comparison with the proposed model in the paper with respect to the training dataset size

Approach	Average Accuracy (%)	Size of Training Datasets
ANPR using OD and OCR (2021) [32]	73	900
Fast OCR (Laroca et al.,2021a) [7]	80	772
VITSTR-Base (Atienza,2021) [9]	81	1811
Proposed ESRGAN enhanced OCR Model	84	500

In future we will apply on the different resolutions medical images to find out the robustness of different features in the medical images as reported in shoghi et al. [25, 26]. Non-uniform spacing may lead to overlapping of bounding boxes resulting in misreading of characters. The characters in the number plates need to have a specific area and length-breadth ratio for best possible results. Characters failing to maintain these parameters might lead to errors. The presence of noise, blur or non-uniform illumination in images can affect the accuracy.

5 Ablation study

Ablation study was performed on the proposed model in order to prove the effectiveness of the enhancement layer and the denoising layer. Three different versions of the proposed method namely i) the method without the super-resolution enhancement operation or 'Method – Enhancement', ii) the method without the gaussian removal (denoising) layer or 'Method – Denoising' and iii) the method having neither the enhancement nor denoising layers are compared with the proposed method. The ablation study was performed on the same dataset with respect to the total number of characters correctly recognised and the accuracy metrics mentioned in Eqs. (13) and (14). The data obtained from the study is tabulated in Table 5 in terms of the average, maximum and minimum scores along with the Standard Deviation.

The importance of the enhancement and the denoising layers is evident from the results of the ablation study presented in Table 5. The proposed method has an average accuracy_identified (mentioned in Eq. (14)) of 84.5 exceeding the method without enhancement layer, denoising layer and without both these layers by 65.8, 12.7 and 72.0 respectively.

Table 5 Ablation study on the proposed method showcasing its performance with and without the enhancement and denoising layers

Method	Number of Characters Correctly Recognised		Accuracy_Identified		Accuracy_Present	
	Avg±SD	Max - Min	Avg±SD	Max – Min	Avg±SD	Max - Min
Proposed Method	7.7 ± 1.4	10–4	84.5 ± 10.9	100–50	77.6 ± 14.1	100–40
Method – Enhancement	1.1 ± 1.0	6–0	18.7 ± 34.1	40–0	11.2 ± 10.9	60–0
Method – Denoising	6.5 ± 2.1	10–3	71.8 ± 12.6	100–40	65.8 ± 21.1	100–30
Method – (Denoising + Enhancement)	0.9 ± 1.2	4–0	12.5 ± 25.1	50–0	9.5 ± 12.5	40–0

*Avg = Average, SD = Standard Deviation, Max = Maximum and Min = Minimum values

The proposed method also has an average accuracy_present as mentioned in Eq. (13) of 77.6 exceeding the method without enhancement layer, denoising layer and both these layers by 66.4, 11.8 and 68.1 respectively. The proposed OCR model also has the best character recognition rate with an average of 7.7 characters out of 10 characters in a numberplate, exceeding the method without enhancement layer by 6.6, without the denoising layer by 1.2 and without both these layers by 6.8. The inclusion of the enhancement layer using the ESRGAN model has drastically improved the results for low-quality plate images as is evident in Table 5. The denoising operation by gaussian blur function also plays a crucial part. The combination of both these layers have proven to give excellent results in comparision to other state of the art OCR model as is evendent in Fig. 4.

6 Conclusion

The OCR model gives a better result averaging at around 85% accuracy rate after undergoing resolution enhancement by the ESRGAN model, which exceeds the initial accuracy rate of 15% for original low-resolution plate images in terms of number of characters recognized and number of bounding boxes generated. Such a high rate of accuracy is achieved with an astonishingly low training dataset and little post-processing, thereby reducing the computational power. The aforementioned model also greatly diminishes the search space for detecting number plates from plate datasets. The enhancement of the original low-resolution images performed by ESRGAN (Avg. Recognition Accuracy – 87%) edges slightly in comparison to enhancement performed by PROSR (Avg. Recognition Accuracy – 76%). Finally the ablation study is performed to establish the effectiveness of the enhancement and denoising layers in the proposed model.We can finally conclude that the enhancement techniques give us better results in comparison to the original unenhanced (low-resolution) images and ESRGAN outperforms ProSR as an enhancement technique as structural similarity achieved for ESRGAN swells around 85% while that of ProSR is below 70%. The upward trend in terms of accuracy and the number of bounding boxes with the improvement of images prove the fact that with better enhancement techniques and better pre-processing techniques, a low-resolution image can be drastically improved and an accuracy nearing 100% can be achieved.

Source code and data The complete source code and data link is available at: https://github.com/bad-eastwind/Super-Resolution-GAN-

Declarations

Conflict of interest Authors have no conflict of interest to declare.

References

1. Amit Y, Geman D, Fan X (2004) A coarse-to-fine strategy formulticlass shape detection. IEEE Trans Pattern Anal Mach Intell 26:1606–1621
2. Car License Plates Dataset (https://makeml.app/datasets/cars-license-plates visited on 3rd July, 2022)
3. Chatterjee S, Dutta RK, Ganguly D, Chatterjee K, Roy S (2020) Bengali handwritten character classification using transfer learning on deep convolutional network. In: Tiwary U, Chaudhury S (eds) Intelligent human

computer interaction. IHCI 2019. Lecture notes in computer science(), vol 11886. Springer, Cham https://doi.org/10.1007/978-3-030-44689-5_13
4. Chen H, He X, Qing L, Wu Y, Ren C, Sheriff RE, Zhu C (2022) Real-world single image super-resolution: a brief review. Inf Fusion 79:124–145
5. Dong C, Loy CC, He K, Tang X (2014) Learning a deep convolutional network for image super-resolution. In: ECCV
6. El-Shal IH, Fahmy OM, Elattar MA (2022) License plate image analysis empowered by generative adversarial neural networks (GANs). IEEE Access 10:30846–30857. https://doi.org/10.1109/ACCESS.2022.3157714
7. Girdher H, Sharma H, Gupta A (2022) Comprehensive survey on devanagari OCR. Available at SSRN 4033489
8. Karthick K et al (2019) Consumer service number recognition using template matching algorithm for improvements in ocr based energy consumption billing. ICIC Exp Lett Part B: Appl ICIC Int 10.10
9. Keipour A et al (2022) Omnifont persian OCR system using primitives. arXiv preprint arXiv:2202.06371
10. Khan RA, Meyer A, Konik H, Bouakaz S (2019) Saliency-based framework for facial expression recognition. Front Comput Sci 13(1):183–198
11. Kim J, Kwon Lee J, Mu Lee K (2016) Deeply-recursive convolutional network for image super-resolution. In: CVPR
12. Kumar M, Jindal SR, Jindal MK, Lehal GS (2018) Improved recognition results of medieval handwritten Gurmukhi manuscripts using boosting and bagging methodologies. Neural Process Lett 50:43–56
13. Lai WS, Huang JB, Ahuja N, Yang MH (2017) Deep laplacian pyramid networks for fast and accurate super-resolution. In: CVPR
14. Laroca R et al (2022) On the cross-dataset generalization for license plate recognition. arXiv preprint arXiv:2201.00267
15. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
16. Lin M, Liu L, Wang F, Li J, Pan J (2021) License plate image reconstruction based on generative adversarial networks. Remote Sens 13(15):3018. https://doi.org/10.3390/rs13153018
17. Onim MD et al (2022) BLPnet: A new DNN model and Bengali OCR engine for automatic license plate recognition. arXiv preprint arXiv:2202.12250
18. Qiao C, Li D, Guo Y, Liu C, Jiang T, Dai Q, Li D (2021) Evaluation and development of deep neural networks for image super-resolution in optical microscopy. Nat Methods 18(2):194–202
19. Radwan MA, Khalil MI, Abbas HM (2018) Neural networks pipeline for offline machine printed Arabic OCR. Neural Process Lett 48(2):769–787
20. Rahati S, Morvejian R, Kazemi EM, Kazem FM (2008) Vehicle recognition using contourlet transform and SVM. Proceedings of the Fifth International Conference on Information Technology
21. Rostami M, Berahmand K, Nasiri E, Forouzandeh S (2021) Review of swarm intelligence-based feature selection methods. Eng Appl Artif Intell 100:104210
22. Rostami M, Forouzandeh S, Berahmand K, Soltani M (2020) Integration of multi-objective PSO based feature selection and node centrality for medical datasets. Genomics 112(6):4370–4384
23. Rostami M, Forouzandeh S, Berahmand K, Soltani M, Shahsavari M, Oussala M (2022) Gene selection for microarray data classification via multi-objective graph theoretic-based method. Artif Intell Med 123:102228
24. Roy S, Bhattacharyya D, Bandyopadhyay SK, Kim TH (2017) An improved brain MR image binarization method as a preprocessing for abnormality detection and features extraction. Front Comput Sci 11:717–727. https://doi.org/10.1007/s11704-016-5129-y
25. Roy S, Whitehead TD, Li S, Ademuyiwa FO, Wahl RL, Dehdashti F, Shoghi KI (2022) Co-clinical FDG-PET radiomic signature in predicting response to neoadjuvant chemotherapy in triple-negative breast cancer. Eur J Nucl Med Mol Imaging 49(2):550–562
26. Roy S, Whitehead TD, Quirk JD, Salter A, Ademuyiwa FO, Li S, An H, Shoghi KI (2020) Optimal co-clinical radiomics: sensitivity of radiomics features to tumor volume, image noise and resolution in co-clinical T1-weighted and T2-weighted magnetic resonance imaging. EBioMedicine, The Lancet, volume 59, 102963
27. Sabóia CMG (2022) Brazilian mercosur license plate detection and recognition using haar cascade and tesseract OCR on synthetic imagery. International Conference on Intelligent Systems Design and Applications. Springer, Cham
28. SaimaRafique, Iqbal M, Habib HA (2009) Space invariant vehicle recognition for toll plaza monitoring and auditing system. Multitopic Conference. INMIC 2009, IEEE 13th International, pp 1–6
29. Salma MS, Rauf ur R, Khan MG, Zulfiqar A, Bhatti MT (2021) Development of ANPR framework for Pakistani vehicle number plates using object detection and OCR. Complexity 2021, Article ID 5597337, 14 pages. https://doi.org/10.1155/2021/5597337

30. Silva SM, Jung CR (2020) Real-time license plate detection and recognition using deep convolutional neural networks. J Vis Commun Image Represent 102773. https://doi.org/10.1016/j.jvcir.2020.102773
31. Singh S, Sharma A, Chauhan VK (2021) Online handwritten Gurmukhi word recognition using fine-tuned Deep Convolutional Neural Network on offline features. Mach Learn Applic 5:100037
32. Srilekha B, Kiran KVD, Padyala VVP (2022) Detection of license plate numbers and identification of non-helmet riders using yolo v2 and OCR method. 2022 International Conference on Electronics and Renewable Systems (ICEARS). IEEE
33. Su H, Tang L, Wu Y, Tretter D, Zhou J (2011) Spatially adaptive block-based super-resolution. IEEE Trans Image Process 21(3):1031–1045
34. Vu X-S et al (2021) Mc-ocr challenge: Mobile-captured image document recognition for vietnamese receipts. 2021 RIVF International Conference on Computing and Communication Technologies (RIVF). IEEE
35. Wang R, Fujii Y, Popat AC (2022) Post-ocr paragraph recognition by graph convolutional networks. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision
36. Wang X, Xie L, Dong C, Shan Y (2021) Real-esrgan: training real-world blind super-resolution with pure synthetic data. In proceedings of the IEEE/CVF international conference on computer vision (pp 1905–1914)
37. Zhai X, Bensaali F, Sotudeh R (2012) OCR-based neural network for ANPR. In IEEE, Pp1
38. Zhang K, Sun M, Han X, Yuan X, Guo L, Liu T (2017) Residual networks of residual networks: multilevel residual networks. IEEE Trans Circ Syst Video Technol
39. Zhang Y, Tian Y, Kong Y, Zhong B, Fu Y (2018) Residual dense network for image super-resolution. In: CVPR

Affiliations

Anwesh Kabiraj[1,2] · Debojyoti Pal[1,3] · Debayan Ganguly[3] · Kingshuk Chatterjee[4] · Sudipta Roy[1]

[1] Artificial Intelligence & Data Science, Jio Institute, Navi Mumbai, Maharastra 410206, India

[2] Information Technology, Government College of Engineering and Leather Technology, Sector-III, Bidhannagar, Kolkata, West Bengal 7000983, India

[3] Computer Science and Engineering, Government College of Engineering and Leather Technology, Sector-III, Bidhannagar, Kolkata, West Bengal 7000983, India

[4] Computer Science and Engineering, Government College of Engineering and Ceramic Technology, Beleghata, Kolkata, West Bengal 700010, India

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com