Real-Time Object Recognition using Region based Convolution Neural Network and Recursive Neural Network

R. Priyatharshini, Aswath Ram. A.S, R. Shyam Sundar, G. Nethaji Nirmal

Abstract*: The recognition of real-world objects demands the recognition and characterization of digital image samples. Automated methods for the detection and recognition of entity types have many significant commercial and industrial applications. While deep convolution neural networks (CNN) and machine learning (ML) concepts have contributed to the classification of globe items, they cannot fully scale the reliance of powerful GPUs to classify the key attributes of images. By using a Recurrent Neural Network (RNN) we tend to resolve the issue arisen in the previous systems. In particular, a hybrid approach using R-CNN and RNN has been proposed that improve the accuracy of object recognition and learn structured image attributes and begin image analysis. Specifically, we applied the transfer learning approach to pass the load parameters which were pre-trained on the Image web dataset to the RNN portion and follow a custom loss feature for the model to train and test more rapidly with precise weight parameters. Experimental results show that in comparison to CNN models like Resent, origin V3, etc., our proposed model achieved improved accuracy in categorizing universe pictures.

Keywords*: Real time object recognition, *Convolution Neural Network, Recurrent Neural Network, Transfer Learning

I. INTRODUCTION

One of the fundamental problems faced by computer vision is Object detection. It provides information for comprehension of images and videos and has gained a lot of attention in recent times. Object identification is responsible for the prudent task of localizing objects in an image. The sliding windows have been used for generating object hypotheses, which are classified using a model like the "Deformable Parts Model" The meteoric rise of convolution neural networks for computer vision has prompted analysis into applying CNNs for objection detection. Recently, Faster R-CNN is used for the accurate detection of objects in a picture with heightened speed. It achieved a 73.2 mean average precision (map) on the PASCAL VOC 2007 dataset. By a combination of both the RCNN and the RNN classifier into one network, it is able to learn automatically good feature representation for the task a lot faster than previous systems. With the release of the Image Net VID dataset, we apply the Faster R-CNN model to the video clips and also real-time images. We find that RCNN has better performance and untenable for video (e.g. treating each frame independently). By using both the neural networks we are able to improve the performance of the system significantly. Recent progress in object detection was motivated because of the popularity of regional solution methods and region-based convolution neural networks (R-CNNs) for greater precision. Even though R-CNNs were computationally expensive as freshly developed, their price decreased drastically because of the sharing of convolutions across systems. The latest development, Fast R-CNN, clocked close to a period of time rates victimization terribly deep networks, once ignoring the time spent on region proposals. Selective Search (SS), one of the most well-known methods, blends superpixels based on low-level attributes. Currently, Edge Boxes provides the best quality and speed. Recently clocked at 0.2s per image. It is duly noted that the runtime comparisons are rendered negligible when fast RCNN is applied to the CPU which works off the GPU to their full extent. Though the method might be an efficient solution it lacks computer sharing possibilities and is ineffective during re-implementation.

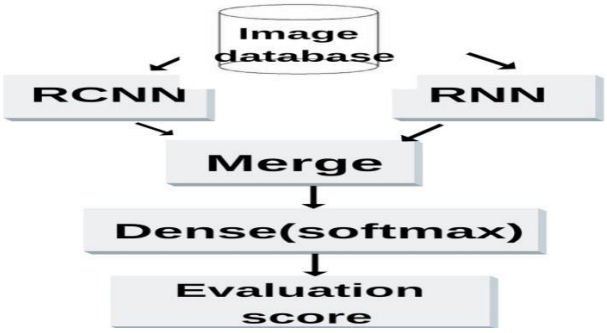

Fig. 1: Overview of the proposed method using the CNN-RNN framework and transfer learning for categorizing images

Revised Manuscript Received on November 15, 2019

Dr.R. Priyatharshini *, Department of Information Technology, Easwari Engineering College, Ramapuram, Chennai 600 089 India, Email: Priya.sneham@gmail.com

Aswath Ram. A.S, Department of Electrical and Electronics, Easwari Engineering College,Ramapuram, Chennai 600 089.India, Email: aswathram3@gmail.com

R. Shyam Sundar, Department of Information Technology, Easwari Engineering College,Ramapuram, Chennai 600 089.India Email: nethaginirmal13@gmail.com

G. Nethaji Nirmal, Department of Information Technology, Easwari Engineering College,Ramapuram, Chennai 600 089.India Email: shyamamshy9941@gmail.com

II. LITERATURE SURVEY

Real-time object recognition is achieved using Image Net in 2012, Krizhevsky et al. [1] reinvigorated of the image characterization and classification. Alex Net was being used before in the previous models. Deep learning performs well in image categorization tasks [2-5] which can be noted from its recent improvements. The proposed method uses the RCNN-RNN algorithm and tensor learning for categorizing real-world object images. We have put forward a system that combines both RCNN and RNN [6-7]. It combines the local attributes from the RCNN layer with the features from the RNN [8] to classify the image. The pre-trained CNN [9] model on the Image Net dataset and maintain its weight parameters are used.[2] The RCNN is frozen at the beginning and the RNN layer is loaded with the pre-trained data set.[2] The attributes from the RNN are extracted and by merging the RCNN and RNN we get the final image classification.[2] The parameters of the training data are constantly adjusted in order to improve the precision of RNN. In the end, all the layers are removed and the data is used as feedback for the CNN and RNN model. Then the results of the classification are shown with a Soft Ax [10]. We also retrain the CNN-RNN model to improve the accuracy and eventually achieve better results of the classification. The ease of extraction of relevant features and online recognitions yields better results than the previous methods [11]. The most important distinction is the detection of individual letters and the identification of the whole text. The latter is more difficult and perfect results for character recognition that have been obtained [12], [13] have never gone hand in hand with full lines of texts. Handwriting recognition is so diversified that it has multiple constraints. A hybrid technique of cluster generative statistical time warping is seen with a better dynamic time integration with HMM's and merging, clustering and statistical time modeling into a single function space [2]. This could be done by pre-segmenting words into characters also referred to as the paradox of Sayre [20]. The existing system runs in GPU and fetches information from the main server every time for providing results. Our proposed system runs on a mobile GPU and fetches information from the trained data set which provides faster results.

III. METHODOLOGY

Neural networks have become the trite concept nowadays because of the improvement in biotechnology. Neurons generally are like an individual perception of the entire input. They are organized together to achieve an apprehension of the object in the given field of view as done in [2]. RCNN extracts both the local and hidden attributes of the image that is sent in as input. Using Recurrent Neural Network for the processing of the sequence data can be beneficial. A normal neural network cannot handle digital image kind of sequential data because there is always a link between the input layer to the hidden layers to the final layer the serves as the output. A convolution-recursive deep model in 3D objects classification and categorization combines the Region convolution and recurrent neural networks (RCNN and RNN) into a single system. The RCNN layer of the system is used to learn low-level translation-invariant features in the given input image. This is then used as input for the multiple RNNs of the system to formulate complex attributes of the image. RNN can be viewed as one of the most efficient systems that find combined working stages of CNN and highway networks. The output of the RNN is used as input for a recurrent LSTM (Long-short Term memory). Combining the RCNN and the Recursive Neural network system that is tuned based on the digital image dataset with critical weight parameters and attributes passed from one more RCNN that is already turned on the Image Net dataset. Both these models obtain comparatively better results that the pre-ordained or previously used system. Inspired from these results, we combined the RCNN and RNN model for classification and categorization of Real-world object images. We have hence named the model in this script as "RCNN-RNN". This method incorporates both the training and testing phase. The object image dataset undergoes the data pre-processing techniques which are after the RCNN model was pre-trained with the Image Net dataset. Following this step, a transmission learning method is applied to the system to boot up the RCNN layer. Succeeding after this all the CNN models are frozen and the training for RNN is progressed. In the meantime, Neural attention mechanisms are used to combine RCNN and RNN properties and attributes. The pre-processed sample images are sent as an output to the fine-tuned RCNN-RNN system during the testing phase, thereby obtaining the outcomes of the classification through a Soft Ax layer.

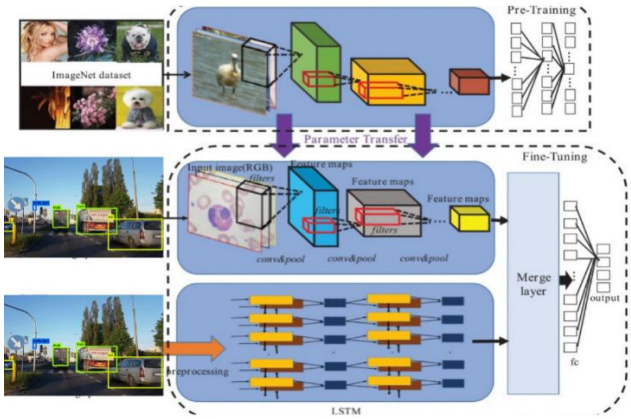

Fig. 2: CNN model proposed for the for fine-tuning on given image dataset

A. Data preprocessing:

For refining the accuracy of the network model, the image dataset has to be refined; this can also reduce the over-fitting [2]. Matrix transformations are used to increase the number of images in the dataset rotation matrix algorithm is applied over the images in the dataset. However, rotating object images may reduce a trivial amount of their high-frequency attributes and shouldn't bring about any changes in the abnormality/normality of a maximum number of the images. Equalization of samples improves the speed and accuracy of the training model which can overcome the prescient disadvantages.

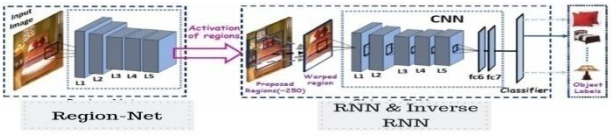

Fig. 3: Process flow of the proposed system with Region-NET for providing RNN & Inverse RNN

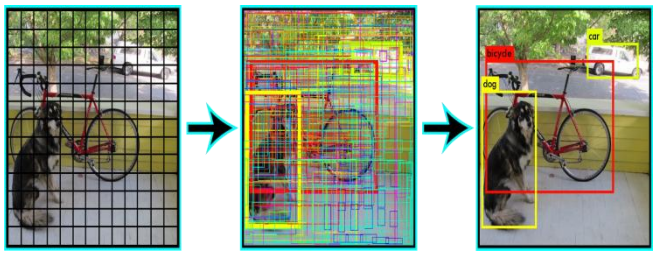

Fig. 4: Object recognition of the system through the proposed system

B. Model development:

The proposed model comprises of the following parts: an RCNN layer, RNN layers, and completely connected Soft Ax output layer. The end framework of the network model after all the design parameters are taken into consideration is given as follows.

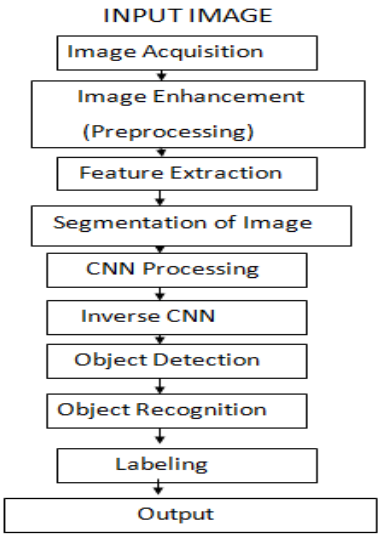

Fig. 5: flow chart of object recognition

1) PRE-TRAINEDCONVOLUTIONAL NEURAL NETWORK LAYER

The Region convolution neural network comprises of two layers, a pooling layer, and a convolution layer. The weighted parameters obtained from the pre-trained image dataset are used as the preliminary weights for the RCNN model. The initialization weights are obtained from the pre-processed image attributes.

2) REGION *CONVOLUTIONAL* LAYER

The crucial part of the CNN, the cardinal method used to calculate is by using the technique called convolution window with varying sizes so that we can perform regional convolution operation with the feature and attributes mapping of the previous computation layers. The Convolution window of various sizes usually slides in sequential order to the attribute map of the previous layers. The weight of the window generally used is 3×3 or 5×5, and the number of weighted parameters of the convolution part also revised correspondingly. Each of the neurons present in the RCNN layer is processed through their convolution windows, which then leads to end result using the excitation function situated in the neural layer. The output is refined well-sequenced after processing in the RCNN layer.

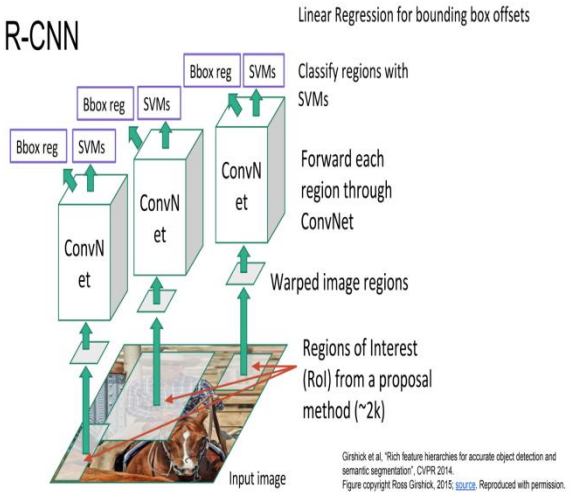

Fig.6: Image layer classification for R-CNN

3) POOL LAYER

The calculation procedure is more likely similar to the RCNN layer. A dominant difference seen is nothing but the sliding window present in the lower-lying layers is 2 × 2, and the sliding step for each of the layers is 2. The feature map gets halved than their predecessor because of the condition due to which convolution weights and the speed of the Neural can be reduced to a great extent [3], the more the number of training data presents the better is the speed of the process overall. Meanwhile, this also enables the model to be more adaptive and susceptible to be trained to scale the image changes. Three input doors are placed in the cell, a forgotten gate, and an output gate as in [2]. Messages passed through the LTSM are easily judged by using rules which is advantageous for us. Finally, the inconsistent data will be thrown out by the Oblivion Gate.

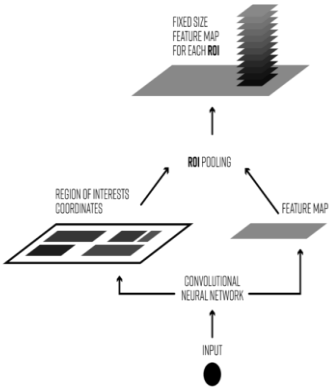

Fig. 7: Roi pooling representation

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

4) *RNN LAYER*

The RNN and CNN layers contain 3 sub-layers each: an input layer, a hidden layer and then an output layer. The connections between the layers play an important role. The input and the hidden layer are connected through the network and the hidden layer output is used as the input for the upcoming output layer. To turn the process dynamic, the output layer is again fed back into the adjacent hidden layers. This model closely mimics the biological nervous system and so the RNNs are cyclic networks with recurrent nature. In the following paper, we use the LTSM recurrent neural model(Long term short Term) which is depicted in the picture below,

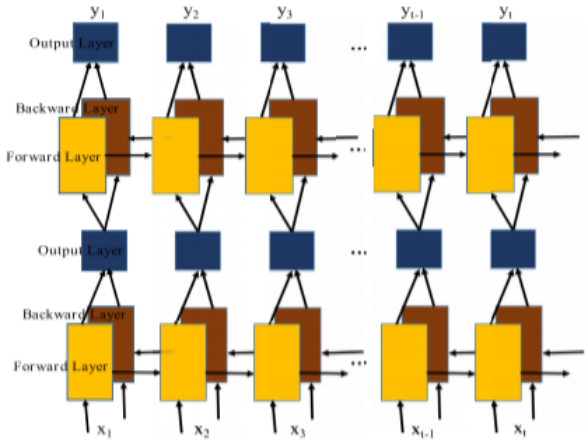

Fig. 8: ***bidirectional LSTM structure shown in the preceding figure (X indicates input data node, Y indicates the output node), X and Y combining forward and backward to form bi-directional LSTM.***

HYBRID (RCNN_RNN) ALGORITHM FOR REAL WORLD OBJECT DETECTION

STEP1: Recognition of image.

STEP2: Image acquisition and preprocessing occurs.

$$FOV_x = 2.\tan^{-1}\left(\frac{width\ of\ sensor/2}{f}\right) \quad (1)$$

$$FOV_Y = 2.\tan^{-1}\left(\frac{width\ of\ sensor/2}{f}\right) \quad (2)$$

FOV= field-of-view
FOV_x = field of view in horizontal direction
FOV_y =field of view in a vertical direction
f = focal length

STEP3: Feature extraction and pooling of images occur.

Layer = averagePooling2dLayer (pool Size)
Layer= averagePooling2dLayer (pool Size, Name, Value)

STEP4: Convolution processing and Inverse Convolution neural network processing occurs

$$R_{Learn} = \arg\min_{R_\theta = \theta \in \phi} \sum_{n=1}^{N} f\left(x_n, R\theta\left(y_n\right)\right) + g(\theta) \quad (3)$$

f = cost function and regularization
arg min= optimization
x_n, y_n = training set
$R_\theta = \theta \in \phi = network\ architecture$

STEP5: Detection, Recognition, and Labeling of an object occur.
Detection:

$$s = \log\left(\frac{OH(R,G,B)}{NH(R,G,B)}\right) \quad (4)$$

OH- Object histogram
NH- Non- object histogram

STEP6: Recognizing image and the labeled image are displayed on the screen.

IV. EXPERIMENT AND RESULTS

The experiments were carried out with this model to get the exact figures and accurate results. The detailed experimental procedure and the model descriptions are given as follows:
The model comprises of 2 main part: The RCNN part that uses exception model and transfers learning methodology; and then the RNN section that employs the bidirectional LSTM. This part is trained using both the sections: First where all of the RCNN layers are frozen and the classification layers and RNN networks are schooled. The whole scheme of the model is to detect, recognize and categorize real-world images. Exception LSTM is run in the Tensor flow framework by employing an NVIDIA Tesla k40s GPU with 12GB of memory. In order to show our model's superiority in image classifications, we used the following models for conducting the experiment and compare experimental results with our dataset.
ResNet50, Exception, InceptionV3, ResNet50- LSTM, Exception-LSTM, and InceptionV3-LSTM

With this, the experimental setup is built and the environment is set up for the training of the RCNN-RNN.
Image Dataset is derived from multiple sources and is collectively put under a single unit. The unit is constantly refreshed with a new set of images and the attributes of the images are changed periodically to test the flexibility of the model.

A. Experimental setup:

The raw data is retrieved from the Image net dataset and some publicly available datasets. We get a new dataset consisting of 2, 00,000 images of objects (JPEG), which is then divided as training and test images, respectively. In the dataset there a wide range of object images which include airplane, dog, cat, table, spoon, etc.

Fig. 9: data sample

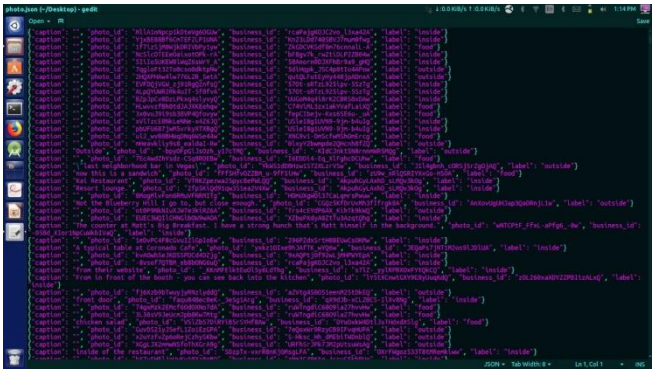

Fig. 10: attributes of the data sample

B. Experiment Results:

Fig.11: Detection and labeling of image

C. performance measures:

Table1: the above performance measure table is calculated by the equation from eq.5- eq.10

%	BBS	W4	SGM	MGM	RCNN & RNN
Perfect Detections	84.2	81.7	86.7	84.9	91.19
Failures in detection	12	9.61	11.3	13.2	8.3
Splits in detection	2.91	5.41	0.21	1.92	0.31
Merges in detection	0.3	1.0	0.3	0.5	0.9
Matching Area	64.7	50.4	61.9	61.3	78.8

The performance measure is calculated by the equation from eq.5-eq.1

Basic Background Subtractions (BBS), W4, Single Gaussian Model (SGM), Multiple Gaussian Models (MGM), Region convolution neural network (RCNN)& Recursive Neural network(RNN)

1) Correct Detection (CD): regions detected are mapped to one and only region.

$$precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

2) Detection Failure (DF):

RPN = Severity X Occurrence X Detection (7)

3) Merge Region (M): many of the detected regions are associated with the ground truth region.

Img1=Image1;
Img2=Image2;
Img3=Combined Image;
Img3 = [Img1 Img2]; %# Concatenate horizontally (8)

4) Split Region (S): ground truth regions are linked with the number of detected regions.

mask=poly2mask (x, y) (9)

5) Matching Area (SM): when all conditions specified in 4,5 are satisfied simultaneously

$$M(i) = \frac{R_i \cap R_j}{R_i \cup R_j} \quad (10)$$

M = area of overlap
M(i) = image sequence
j= index of detected region
R_i, R_j = detected region

Fig.12: the above graph shows the correct detections of the algorithm

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

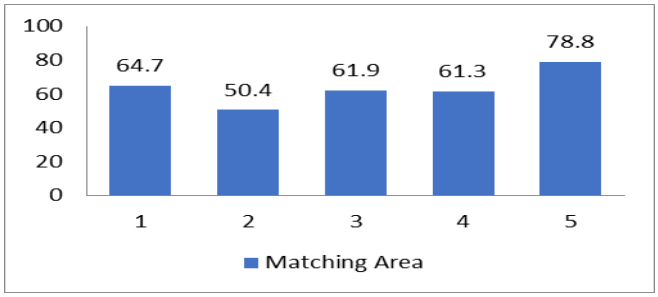

Fig. 13: the above graph shows the matching area of the algorithm

V. CONCLUSION

A Hybrid neural network approach which combines the attributes of convolution neural networks and recursive neural networks has been developed for recognizing the real time objects. This model retains the spatial and temporal information of the processed image and can easily learn structured information of the image attributes. Compared with the previously existing methods, our proposed technique was able to achieve high performance in terms of categorization of real time objects and its related dataset. The proposed approach can be extended to other pattern recognition applications I order to improve the prediction accuracy .

ACKNOWLEDGMENT

This work was supported by DST-FIST Programme No.SR/FST/College-110/2017, Government of India

REFERENCES

1. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘‘ImageNet classification with deep convolution neural networks,’’ in Proc. Int. Conf. Neural Inf. Process. Syst., vol. 60, 2012, pp. 1097–1105.
2. Garbo Liang, Huichao Hong, Weifang Xie, Lixin Zheng. "Combining Convolutional Neural Network with Recursive Neural Network for Blood Cell Image Classification", IEEE Access,2018
3. K. He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Compute. Vis. Pattern Recognition., Jun. 2016, pp. 770–778.
4. K. Simonyan and A. Zisserman, ‘‘Very deep convolution networks for large-scale image recognition,’’ in Proc. ICLR, 2015, pp. 1–14.
5. C. Szegedy et al., ‘‘Going deeper with convolutions,’’ in Proc. IEEE Conf. Compute. Vis. Pattern Recognition. , Jun. 2015, pp. 1–9.
6. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ‘‘Rethinking the inception architecture for computer vision,’’ in Proc.
7. IEEE Conf. Compute. Vis. Pattern Recognition., Jun. 2016, pp. 2818–2826.
8. A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, ‘‘A novel connectionist system for unconstrained handwriting recognition,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 855–868, May 2009.
9. B. Su and S. Lu, ‘‘Accurate scene text recognition based on recurrent neural network,’’ in Proc. ACCV, vol. 9003, 2014, pp. 35–48.
10. A. Graves, A.-R. Mohamed, and G. Hinton, ‘‘Speech recognition with deep recurrent neural networks,’’ in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., Vancouver, BC, Canada, vol. 38, May 2013, pp. 6645–6649.
11. Y. LeCun, L. Bottom, Y. Bengio, and P. Haffner, ‘‘Gradient-based learning applied to document recognition,’’ Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
12. Y. Sun, Y. Chen, X. Wang, and X. Tang, ‘‘Deep learning face representation by joint identification-verification,’’ in Proc. Conf. Workshop Neural Inf. Process. Syst., vol. 2, 2014, pp. 1988–1996.
13. R. Seiler, M. Schenkel, and F. Eggimann, “Off-line cursive handwriting recognition compared with on-line recognition,” in ICPR ’96: Proceedings of the International Conference on Pattern Recognition (ICPR ’96) Volume IV-Volume 7472. Washington, DC, USA: IEEE Computer Society, 1996, p. 505.
14. R. Plamondon and S. N. Srihari, “On-line and off-line handwriting recognition: A comprehensive survey,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 1, pp. 63–84, 2000.
15. A. Vinciarelli, “A survey on off-line cursive script recognition,” Pattern Recognition, vol. 35, no. 7, pp. 1433–1446, 2002.
16. H. Bunke, “Recognition of cursive roman handwriting - past present and future,” in Proc. 7th Int. Conf. on Document Analysis and Recognition, vol. 1, 2003, pp. 448–459.
17. I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, “Unipen project of on-line data exchange and recognizer benchmarks,” in Proc. 12th Int. Conf. on Pattern Recognition, 1994, pp. 29–33.
18. J. Hu, S. Lim, and M. Brown, “Writer independent on-line handwriting recognition using an HMM approach,” Pattern Recognition, vol. 33, no. 1, pp. 133–147, January 2000.
19. C. Bahlmann and H. Burkhardt, “The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time are warping,” IEEE Trans. Pattern Anal. And Mach. Intell., vol. 26, no. 3, pp. 299–310, Mar. 2004.
20. C. Bahaman, B. Haddon, and H. Burckhardt, “Online handwriting recognition with support vector machines - a kernel approach,” in Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition, 2002, pp. 49–54.
21. G. Wolfing, F. Sinned, and L. Ruedisueli, “On-line recognition of handwritten symbols,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 9, pp. 935–940, 1996.

AUTHORS PROFILE

Dr.R.Priyatharshini received her BE degree from University of Madras in 2001 and ME degree from Anna University in 2008. She has completed her PhD degree from Anna University in 2017. She has 16 years of teaching experience and currently working as an associate professor in the Department of Information Technology, Easwari Engineering College, Chennai. She has published more than 20 research papers in International Conferences and Journals. Area of interest towards Data Mining, Medical Image Processing and Big Data Analytics. She is a life time member of Institution of Engineering and Technology.

Aswath Ram. A.S pursuing his BE degree in Easwari Engineering College, has a take towards combining state of art machine learning and neural networks to the electronics side of the world. Aspirations to become a robotics engineer found the niche field of combining computing and hardware into one field.

R. Shyam Sundar completed his BE degree in Information technology and pursuing a career in data mining and Soft computing, has an inclination towards the machine learning field.

G. Nethaji Nirmal finished his BE degree in Information technology at Easwari engineering college, is pursuing a career at a reputed firm. He has ambitions of becoming data scientist or a machine learning enthusiast one day in the near future and working towards it.