Grenze International Journal of Engineering and Technology, Jan Issue

Cartoon Classification using Recurrent Multiscale Convolutional Neural Network

Prem Singh M[1] and Sharath Kumar Y H[2]
[1]Department of Computer Science, Government College (Autonomous) Mandya
[2]Department of ISE, Maharaja Institute of Technology Mysore
Email: premmsingh1973@gmail.com, sharathyhk@gmail.com

***Abstract*— A type of artificial neural network called recurrent neural networks (RNN) has gained popularity recently. Unlike feedforward networks, the RNN is a unique type of network with recurrent connections. The main advantage is that the network can now refer to previous states and process arbitrary input sequences thanks to these connections. CNN normally has a feed-forward architecture, whereas the visual system has a lot of recurrent connections. This is a key distinction between CNN and RNN. Recurrent connections are incorporated into each convolutional layer of the Recurrent Multiscale Convolutional Neural Network (RMCNN), a combination of Multiscale convolutional and recurrent neural networks, which is inspired by this fact, and is proposed for the categorization of Cartoons. Even though the input is static and each unit is affected by its neighbours, the network can nevertheless change over time with the help of these connections. This attribute incorporates an image's context data, which is crucial for categorizing images. According to this method, network parameters can be learned from data by training a multiscale CNN that can handle images with RNN to grasp the context of the images by sending the output of one training step to the input of the next training steps. Either a fully universal time-varying weighting or temporal pooling is learned by CNN models. A model using a combination of Multiscale CNN and RNN is proposed in accordance with existing CNN-based methods. By utilising straightforward neural gates that resemble memory cells, recurrent neural networks with long-range learning enhance hidden state with nonlinear processing methods. The contributions in this work are: First, RNNs are used to extract contextual information. Second, RMCNN is proposed for logo classification, which combines the advantages of both MCNNs and RNNs.**

***Index Terms*— CNN, RNN, RMCNN, Classification, Retrieval**

I. INTRODUCTION

Cartoon is one of the most popular artificial arts in the history. It is well-liked by everyone for its art and the comical characters. Cartoons are one of the most effective means of communication which can convey the message more quickly than a written notice. The cartoon images can be used powerfully in the advertising industries. The skill of the effective cartoonist is not just the skill to draw well, it involves the ability to distill an idea in the form of images. This communication shortcutting makes the cartoon image to carry message effectively. The cartoon images as secondary products are now all over the internet and hence the customers have a strong demand to find the cartoon image by retrieval. Due to the huge growth in the amount of cartoon images the necessity for cartoon

Grenze ID: 01.GIJET.10.1.173

images retrieval is increased. It would be impossible to cope with the extension of cartoon images unless those data could be retrieved effectively and efficiently. Unfortunately, most cartoon image retrieval systems are text based methods. The text based retrieval methods could be retrieved if the images are well annoted. In other words, cartoon images without annotation make them incapable of being retrieved. Hence content based cartoon image retrieval, retrieves with content based instead text based, plays an important role in multimedia system. The cartoon based image retrieval systems can be defined as: For a given query cartoon image, finding similar cartoon images stored in database. The search process relies on how faster the images can be retrieved from the large database. To achieve a fast retrieval speed and to make the retrieval system truly scalable to the large size of the image collections, an effective indexing structure is a paramount part of the whole system.

II. Related Works

Haseyama and Matsumura [1] have proposed a cartoon image retrieval system, in which the partial features are referred as Regions and Aspects [1]. These partial features were used to compute cartoon image similarities. The cartoon images in the database that are most similar to the query image are returned as results. This system proposed can only be used to retrieve the face of the cartoon images. It limits the application of this system to the character design reuse rather than the cartoon sequence synthesis which is a setback. Yang et al., [2] proposed a Retrieval-based cartoon Clip Synthesis (RCCS) method that is used to combine edge and motion direction to represent a cartoon character. It uses an unsupervised distance metric learning algorithm for the cartoon image retrieval. The algorithm is formulated by a trace ratio maximization problem, which can be optimized by the iterative approach [2]. Sykora et al., [3] proposed a counter algorithm for cartoon image detection called Curve structure extraction for cartoon images. In their work, a novel counter detection algorithm is used to get counters with a very good connectivity based on the assumption that foreground parts of cartoons are surrounded by bold dark contours. Curve structure extraction for cartoon images was not suitable for most modern cartoons as the assumption were too strict. Tiejun Zhang et al., [4] proposed feature that uses the Harris-Laplace corner to localize all the key points and corresponding scale in the cartoon image. Then, the local shape was described by the shape context. The feature point matching is achieved by a weighted bipartite graph matching algorithm and the similarity between both the query and the indexing image is presented by the match cost. The curves in the cartoon images, and described as the main content and introduces a local feature named Scalable Shape Context (SSC) based on the SC feature. Jun Yu et al., [5] proposed a boundary curve extraction methods for all the general images: A summary of boundary curve extraction methods [5], it is also called edge detection. These methods define curves as sharp changing image region and detect them by finding curve points with a maximal gradient magnitude along the curve profile. Though it works well for boundary curves, it is not suitable for decorative curves. Two parallel curves on either side of decorative curve will be produced. This brings difficulties to further processing like vectoring and editing. H. Kang et al., [6] proposed decorative curve extraction methods for general images: Decorative curve extraction methods define curves as lines with small but with a finite width [6]. These algorithms analyze curves by modelling the curves and their surroundings as well. However, their models of curve profiles are suitable only for decorative curves. Significant bias will arise for boundary curve. Moreover, as not introducing regular feature of cartoons, noise in orientation information can't be reduced well enough especially in weak curve area. Belongie [7] proposed a feature called Shape Context (SC), which allows measuring the shape similarity between all the curvilinear structures, and it is used in recognition of digits, silhouette similarity-based retrieval. The basic idea of SC is to selext a pixel and model the distribution to other curve pixels. Shape Context will not describe the object path for a more common scene with background and when scale of the object is not predefined. K bozas et al., [8] this method each image is divided into pool of patches. For each patch Histogram of Oriented Gradients (HOG) is applied to find the object in that particular patch. Later binarization is applied for the extracted feature. Then Min-hash tuples are calculated and it is the final patch representation. For each tuples created a lookup in hash table is performed. The sketch provided from the user also under goes the same process and finally the hash tables of the sketch and the hash in database are compared on the voting basis. The database consists of 31 user drawn queries outlining objects and sceneries. Each sketch query is associated with 40 photos assigned with a value between 1 and 7. The final benchmark score is the average correlation value across 31 queries. The reader can verify the spatial coherence between query and the acquired photos achieved with patch voting scheme. it consumes more memory for storing the hash keys in the hash table and comparing the hash keys in hash table consumes more time if the number of hash keys increases. R Zhou et al., [9] took advantage of the two types of regions that can be found in an image ie. Main region: defined by weighted centre image feature, with this feature no one can retrieve objects in images regardless of their size and positions. Saavedra et al., [10] proposed a novel

method based on the edge orientation which gets a global representation in the name of both sketch and the test image. This method is focuses on estimating local edge orientations and forming the global descriptor name HELO (Histogram of Edge Local Orientation). This has two stages in which the first stage performs the preprocessing tasks which gives an abstract representation of both the query and test images in second stage the histogram is made. To get the rotation invariance this method uses the fourier transform after the feature extraction. In this method the query images need to be drawn in continuous strokes. Juan and Bodenheimer [11] developed Graph based cartoon clip synthesis (GCCS) that combines similar characters into user-directed sequence based on dissimilarity measure in edges. GCCS builds a graph and generates a new cartoon sequence by finding the shortest path between them. It performs well on the simple cartoon characters whereas for the complex colour and gesture pattern will not generate a smooth pattern because the edges can encode neither the colour information nor the gesture pattern and the synthesis result is fixed by the initial specification. R Hu et al., [12] proposed that each image is represented by a bag of regions derived from a region tree. The contours are extracted by region maps containing various level of details and capture local structure in contour map using the Gradient Field Histogram Oriented Gradients (GF-HOG) descriptor. Houssem et al., [13] proposes two content based image retrieval algorithms. The first algorithm extracts the shape features by using support region and the second algorithm uses the shape context descriptor to make it a scale invariant and enhances its performance in the presence of the noise. In [21], they propose a novel technique called facial emotion recognition using convolutional neural networks (FERC). The FERC is based on two-part convolutional neural network (CNN): The first part removes the background from the picture, and the second part concentrates on the facial feature vector extraction. In [22] the cartoon faces in the wild (IIIT-CFW) database and associated problems. This database contains 8,928 annotated images of cartoon faces of 100 public figures. It will be useful in conducting research on spectrum of problems associated with cartoon understanding. This database contains cartoon images of 100 international celebrities (politician, actor, singer, sports person, etc.). In [23], develop a system for recognizing cartoon caricatures of public figures. The proposed approach is based on the Deep Convolutional Neural Networks (DCNN) for extracting representations. In [24], propose a dynamic multi-task learning approach for cross-modal photo-caricature face recognition.

III. PROPOSED METHOD

We propose to develop a method to build an efficient cartoon image retrieval system which can retrieve the similar cartoon image for a given query image with some objectives which includes: Collection of dataset, Pre-Processing of cartoon images, Efficient feature extraction and dimensionality reduction technique for cartoon representation, Effective indexing method for efficient retrieval. Cartoon images from various sources are collected and a database is formed. The pre-processing task involves extraction of edge features for the cartoon images, for this various shape descriptors are extracted. Based on the similarities, the cartoon images similar to query images are identified and retrieved. This method makes use of indexing technique for more effective and accurate retrieval of the cartoon character.

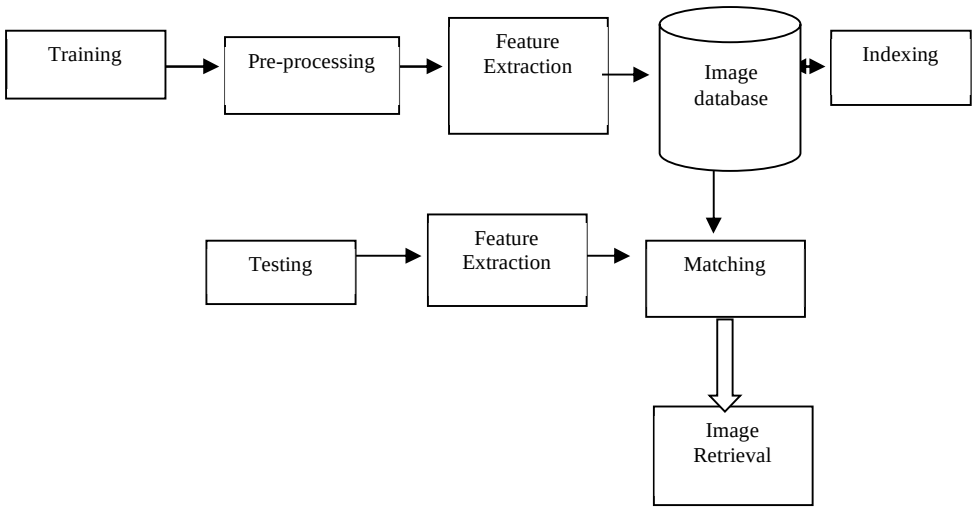


Fig 3: Block diagram of the proposed work

A. Recurrent Neural Networks

A recurrent neural network (RNN) is a type of artificial neural network that extends traditional feedforward neural networks with loops in their connections. Unlike feedforward neural networks, RNNs can process sequential inputs by iterating through hidden states where the activations at each step depend on the activations in the previous step. Hence, the network can exhibit dynamic time behavior. RNNs have shown great potential in time-series data processing, including NLP and speech recognition. An important property of time series is that there is usually a strong relationship between a given sample and previous samples. RNNs are designed to recognize patterns in data sequences such as text, genomes, handwriting, spoken language, or numeric time series data from sensors, stock markets, and government agencies. These methods consider time and order and have a temporal dimension.

B. RNN Structure

The RNN has at least one loopback connection so that the activations are cycled. A common type of RNN is the standard multi-layer perceptron (MLP) plus an additional loop. MLP's non-linear mapping capabilities help improve the performance of RNNs. In another form of RNN, each neuron has a stochastic activation function. If the activation function is deterministic, learning can be done using gradient descent, resulting in a backpropagation algorithm for the feedforward network. However, if the activation is stochastic, the simulated annealing approach is more appropriate. on figs. Figure 4 shows an RNN in its simplest form. This RNN has an MLP and the previous set of hidden module activations are returned to the network with inputs. Activations are updated at each discrete time step, and the delay block must hold activations until processed at the next discrete time step.

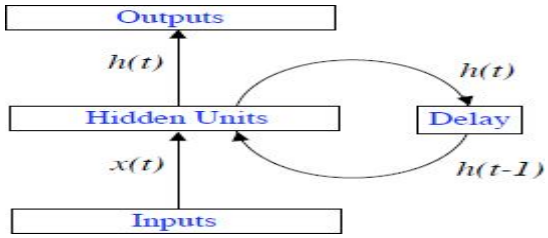


Figure 4: Simplified view of RNN

In RNNs, the output of the previous hidden states is connected to thecurrent states in the form of feedback loops to encode the contextual information of the image sequences.

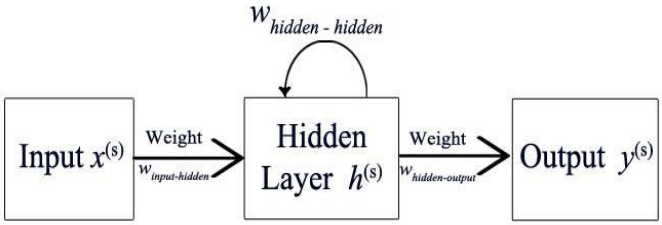


Figure 5: General RNN structure

A typical RNN (Zhen *et al.* [23]) is shown in Figure 5 with inputlayer of length S, hidden layer h(s), and the predicted output y(s) at the state $S \in [1, \ldots, S]$. Deep CNNs are typically used when the network has three or more convolutional layers combined with pooling layers, with a large number of nodes in each layer. Similarly, deep RNNs typically represent networks with multiple iterative layers. As a sequence model, RNNs assume that the current output of a sequence depends not only on the current input, but also on previous outputs.

i. Long Short-Term Memory

The Long Short-Term Memory (LSTM) unit developed by Schmidhuber & Hochreiter in 17, which enables information to be written to a cell, output from the cell, and stored in the cell, represents a significant advance for RNNs. An RNN architecture called LSTM was created for data with distant interdependencies. The input gate, forget gate, and output gate are three multiplicative gate units that regulate the activation of the "memory cells" that make up an LSTM layer. The gates give the cells access to long-range context by enabling them to store and retrieve information over time. Because each cell has a single recurrent link, which is activated by a single forget

gate, the basic LSTM formulation is explicitly one-dimensional. However, it can be extended to n dimensions by using n recursive connections with n forget gates, one for each cell's previous state in each dimension.

ii. Bidirectional Recurrent Neural Networks

In many cases it is useful to have access to the context of the past and future. Therefore, Shuster and Paliwal (1997) introduced bidirectional recurrent neural networks (BRNNs). In these networks, the input space is increased and there are forward and backward connections. A BRNN contains two separate hidden layers that process the input sequence forward and backward. Both hidden layers are connected to the same output layer, so the network has access to both past and future contexts.

c. Recurrent Multiscale Convolutional Neural Network

Multiscale convolutional and recurrent neural networks are used to create RMCNN. Recurrent and multiscale convolutional networks are both used in RMCNN. First, from input sequences, convolutional layers may effectively extract abstract, locally invariant intermediate-level features. Pooling layers minimise computation and aid in the management of overfitting. Second, from feature sequences produced by earlier convolutional layers, recursive layers extract contextual information. The dependencies between various bands in the hyperspectral sequence, which is more stable and valuable for classification, are captured by contextual information. Third, recurrent layers can handle input sequences of different lengths, albeit we are not utilizing this advantage in our work. Long-term dependency and vanishing gradient issues won't typically arise because the lengths of the input hyperspectral sequences are not particularly long (typically below 200) and the pooling layers further lower the length (typically below 50). The 2D images will initially be scanned in four separate scanning orientations since RNN is ideal for one-dimensional time sequences (left to right, right to left, top to bottom and bottom top). The RNN will then be given these four sequences to determine any spatial dependencies. Fully linked layers are used to process the RNN findings as a whole to represent the image globally. As seen in Figure 6. In the architecture of RMCNN, connectivity and shared weights in feed-forward and recurrent connections are local among different locations.

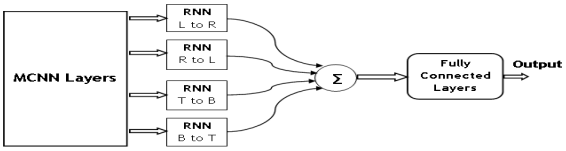


Figure 6: Block diagram of RMCNN Framework

d. Cartoon Retrieval System

In this proposed work, Recurrent Multiscale Convolution Neural Network (RMCNN) is taken with five convolution layers, one recurrent layer and two fully connected layers. Here, MCNN and RNN are to learn middle-level visual patterns and spatial dependencies between the middle level visual patterns respectively. In the last stage, fully connected layers are used to learn a global image representation. When the convolutional layers are more, abstract and robust patterns can be extracted more. With back propagation, the global representations will be transmitted back to RNN for improving the spatial dependency encoding. This will feed CNNs further to learn better middle-level and low-level features. Convolutional layers (including pooling layers) should be used before recurrent layers because they can first collect intermediate-level, locally invariant information from the input sequence. Since the computational complexity of recurrent layers increases linearly with respect to the length of the input sequence, shortening the sequence by pooling layers will speed up backpropagation through the layer. Recurrent layers can thus more effectively extract the spectral dependencies from the middle-level information that convolutional layers have provided. Another advantage of RNN is that it does not require a predetermined length for the input sequence. The method can still be developed to tackle situations where input sequences may have varying lengths even though each pixel in a hyperspectral image in the problem under consideration has the same length of spectrum. The proposed algorithm is shown in Fig. 7.

e. Training Process

Almost all networks are trained using the backpropagation method, but using iterative concatenation requires adaptation. This is done simply through network deployment. We show that the network consists of one cyclic layer and one direct layer. A distributed network can be trained in the same way as a forward propagation network with backpropagation, except that each epoch must go through each distributed layer. The learning rate used for this model is 0.01.

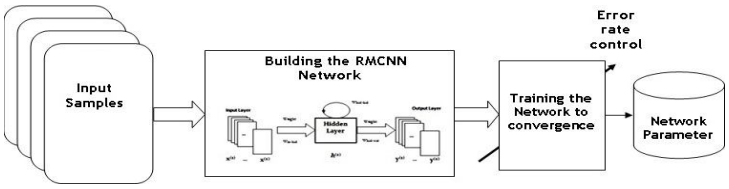
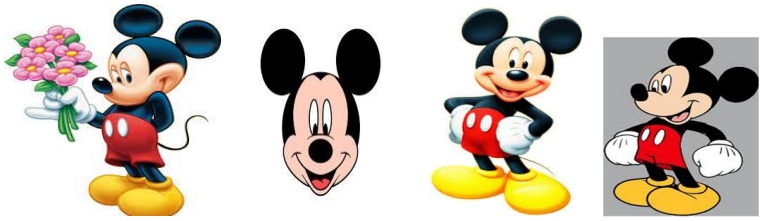


Figure 7: Proposed method of classification with RMCNN

IV. Experimental Results and Discussion

This section is devoted to illustrate the capabilities of the presented deep neural networks (RNN, RMCNN) for logo classification. A traditional method based on SVM with RBF kernel and CNN are also implemented for comparison. The database consists of thirty classes. Each class has twenty cartoon images in it. All the cartoon images are collected from the different cartoon series and internet. The images are present in a different background, position or angle. Also, we have made sure that no redundant images are present in the same class. All images are stored in JPEG format in a sample folder on a hard disk. The features of these cartoon images can be classified into two groups, Intra-features is useful for measuring the similarity between data from the same class and Inter features are used to measure the dissimilarity among the data from different classes. The large intra-class variability and the small inter-class variability make this dataset very challenging. The Figure 8 shows some of the images in image database we have created.

Mickey Mouse



Jerry mouse

Figure 8: Set of sample images in different classes with large intra-class variation

Network Configurations

To evaluate the performance of the proposed RMCNN, SVM, CNN, and RNN are taken as baseline methods. For SVM, RBF is utilized as kernel function. For the CNN, two convolutional layers, two max pooling layers, and one Fully-Connected (FC) layer are selected. Different model structures are implemented based on different images. The training process starts with the weights of all networks randomly initialized, and the initial learning rate is set to 0.0001. During training, the learning rate will be decreased by half every 500 epochs until loss reaches convergence.

Classification Results and Analysis

Here we conducted the experimentation by varying the training and testing sets for estimating the accuracy. The 2k cartoon Dataset size of 10 classes. Initially we the divided the dataset into three random Logo sets of different training and testing images shown in figure 8. Further we compare with other well-known methods. The experiments show that the proposed method outperforms in terms of Overall Accuracy (OA), Average Accuracy (AA) and Kappa Coefficient (k). This is more evident from Tables 1 to 2 and Figures 3 to 4. It is inferred from the Tables that the Average Accuracy is more in the case of Random Set 1 and Random Set 2 whereas Overall Accuracy is more in the case 2. This is also clearly visible from Figures 5 to 6. The classification maps of Logo dataset Random Set 2 are shown in Figure 7 for all classification methods at the maximum overall accuracy. Almost all the classes are marked clearly in proposed RMCNN and many classes are mismarked and with noise in other

previous methods. Still it is challenging to discriminate between two classes with very similar spectral features. The classification maps of Logo dataset Random Set 3 are shown in Figure 8 for all classification methods at the maximum overall accuracy. Almost all the class are marked clearly in proposed RMCNNand many class are mismarked and with noise in other previous methods.

TABLE I: ACCURACY OF CLASSIFICATION FOR RANDOM SET 1

Class No	SVM	CNN	RNN	RMCNN
1	97.18	97.83	97.22	99.32
2	97.28	96.79	96.39	99.09
3	75.34	83.73	85.12	92.73
4	84.36	87.28	88.63	96.30
5	95.27	97.85	97.18	99.13
6	79.65	81.70	84.21	95.18
7	75.37	83.11	84.95	93.93
8	80.49	83.77	84.34	89.47
9	99.42	99.380	98.40	99.46
10	89.72	91.21	91.20	94.8921

TABLE II: VARIOUS ACCURACY MEASURES OF THE RANDOM SET 1 IN %

Type	SVM	CNN	RNN	RMCNN
OA	83.42	85.22	86.39	93.42
AA	84.67	87.77	88.51	94.30
Kappa	76.98	82.12	83.80	92.23

Figure 9: Accuracy of classification for Random

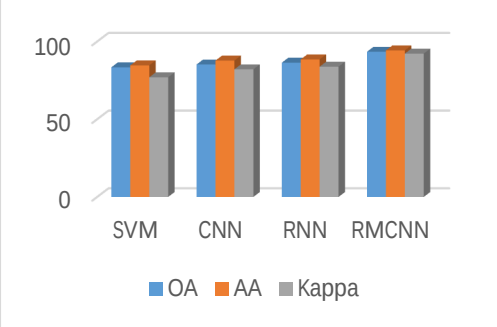


Figure 10: Comparison of various measures of proposed RMCNN method with SVM, CNN and

TABLE III: ACCURACY OF CLASSIFICATION FOR RANDOM SET 2

Class No	SVM	CNN	RNN	RMCNN
1	97.52	96.53	96.12	98.78
2	95.77	94.78	94.83	98.84
3	65.57	68.93	73.26	89.22
4	71.27	76.62	80.32	94.70
5	95.50	98.50	97.74	99.42
6	59.51	63.62	69.67	90.65
7	52.10	66.87	71.54	88.47
8	84.27	83.54	85.11	93.27
9	99.92	99.65	98.36	98.49
10	99.92	99.65	98.36	98.49

TABLE IV: VARIOUS ACCURACY MEASURES OF THE RANDOM SET 2 IN %

Type	SVM	CNN	RNN	RMCNN
OA	82.12	84.45	86.51	96.20
AA	80.16	83.2	85.22	94.65
Kappa	76.98	79.79	82.73	94.91

Figure 11: Accuracy of classification for Random Set 1

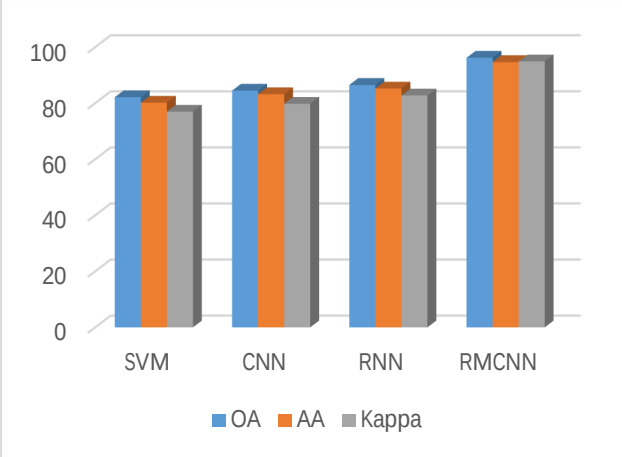


Figure 12: Comparison of various measures of proposed RMCNN method with SVM, CNN and RNN for Random Logo Set 2 data set

TABLE V: ACCURACY OF CLASSIFICATION FOR RANDOM SET 3

ClassNo	SVM	CNN	RNN	RMCNN
1	96.84	99.12	98.31	99.86
2	98.79	98.79	97.94	99.34
3	85.11	95.53	94.75	96.24
4	97.44	97.94	96.96	97.95
5	95.03	97.20	96.63	98.83
6	99.79	99.77	98.72	99.71
7	98.63	99.35	98.36	99.38
8	76.70	83.99	83.56	85.66
9	99.12	99.02	98.13	99.43
10	81.91	85.89	86.29	91.00

TABLE VI: VARIOUS ACCURACY MEASURES OF THE RANDOM SET 3 IN %

Type	SVM	CNN	RNN	RMCNN
OA	84.75	85.99	86.27	90.63
AA	89.18	92.31	91.79	93.95
Kappa	76.98	84.45	89.81	89.55

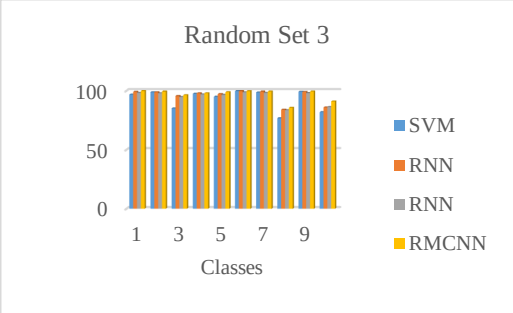


Figure 13: Accuracy of classification for Random Set 1

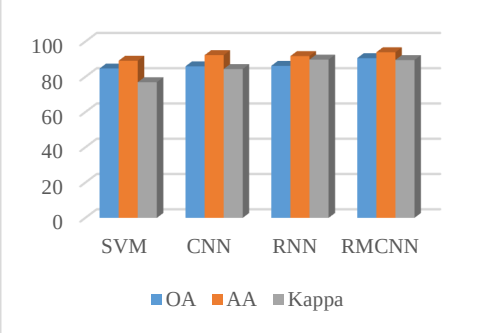


Figure 14: Comparison of various measures of proposed RMCNN method with SVM, CNN and RNN for Random Logo Set 3 data set

V. SUMMARY

A novel logo classification method using RecurrentMultiscale CNN is proposed in this work for classifying various class segmentsin order to derive useful information for further processing in the chosen application. The proposed RMCNN method outperforms compared to the existing methods SVM, CNN, RNN and MCNN-GK in terms of Overall Accuracy, Average Accuracy and Kappa value. Hence, it is inferred that the proposed method is performing better on Logo Random Set 1 dataset. Though it performs well on Random Set 2 dataset, the improvement is very small. For all three datasets, it is found that the improvement is more in Kappa value and

hence, the proposed method classifies better by agreement than by chance alone. For Logo Random Set 1, the improvement of OA is 2.84% more, AA is 3.14% more and Kappa is 3.79% more in RMCNN compared to MCNN-GK method. For Logo Random Set 1, the improvement of OA is 5.04% more, AA is 4.58% more and Kappa is 6.89% more in RMCNN compared to MCNN-GK method. For Logo Random Set 2, the improvement of OA is 0.75% more, AA is 0.64% more and Kappa is 0.76% more in RMCNN compared to other method. Hence, it is inferred that the proposed method is performing better on Random Set 1 dataset. Though it performs well on Random Set 2 dataset, the improvement is very small. For all three datasets, it is found that the improvement is more in Kappa value and hence, the proposed method classifies better by agreement than by chance alone.

REFERENCES

[1] M. Haseyama and A. Matsumura, "A trainable retrieval system for cartoon character images," in Proc. ICME, Jul. 2003, pp. 393–396.

[2] Y. Yang, Y. Zhuang, D. Xu, Y. Pan, D. Tao, and S. Maybank, "Retrieval based interactive cartoon synthesis via unsupervised bi-distance metric learning," in Proc. ACM Multimedia, 2009, pp. 311–320.

[3] D. Sykora, J. Burianek, J. Zara, Sketching cartoons by examples, In Proceedings of The 2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling, 27–34, 2005

[4] Tiejun Zhang, D. Tao, D. Xu, J. Yu, and J. Luo," Recognizing cartoon image gestures for retrieval and interactive cartoon clip synthesis," IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 12, pp. 1745–1756, Dec. 2010.

[5] Jun Yu, Dongquan Liu, Dacheng Tao and Hock Soon Seah "On Combining Multiple Features for Cartoon Character Retrieval and Clip Synthesis", IEEE Trans. Cybernetics, Vol. 42, no. 5, pp.1413-1427, Oct. 2012

[6] H. Kang, S. Lee, C. K. Chui, Coherent line drawing, In Proceedings of the 5th International Symposium on Non-photorealistic Animation and Rendering, 43–50, 2007

[7] S. Belongie, J. Puzicha and J. Malik Shape Matching and Object Recognition Using Shape Contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24 (24): 509-521

[8] K bozas, Qi Han, Handan Hou, Xiamu Niu " Local Invariant Shape Feature for Cartoon Image Retrieval", IEEE Trans. 2013 Second International Conference on Robot, Vision and Signal Processing

[9] R Zhou ,Y. Yang, S. Xiang, and C. Zhang, "Neighborhood MinMax projections," in Proc. IJCAI, 2007, pp. 993–998

[10] D. A. Forsyth, J. Ponce. Computer Vision: A Modern Approach, Prentice Hall, 2002

[11] C. D. Juan and B. Bodenheimer, "Cartoon textures," in Proc. Symp.Comput. Animation, 2004, pp. 267–276.

[12] R Hu and S. Lee, "Human action recognition using shape and CLG-motion flow from multi-view image sequences," Patt. Recog., vol. 41, no. 7, pp. 2237–2252, 2008.

[13] Y. Houssem, D. Zhang, G. Lu, and W. Ma, "A survey of content-based image retrieval with high-level semantics," Patt. Recog., vol. 40, no. 1, pp. 262–282, 2007.

[14] T.Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," IEEE Trans. Known. Data Eng., vol. 21, no. 9, pp. 1299–1313, Sep. 2009.

[15] Dalal, N., Triggs B., 2005. "Histogram of Oriented Gradients for human detection" in : CVPR. pp. 886-893

[16] Lowe, David G. (1999). " Object recognition from local Scale invariant features". Proceedings of the international conference on computer vision 2. pp .1150-1157

[17] E. Oja. Subspace methods of pattern recognition, volume 6 of Pattern recognition and image processing series. John Wiley & Sons, 1993.

[18] D.Gleich[Online].Available:http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=12422&objectType=FILE

[19] S. Balakrishnama, A. Ganapathiraju, Linear Discriminant Analysis.