Human Action Recognition using Scaled Convolutional Neural Network

Aditi Jahagirdar, Manoj Nagmode

Abstract*: *Deep learning is current buzz word in domain of computer vision. In this work, a method for human action recognition based on a variation of General Convolutional Neural Network (GCNN), called Scaled CNN (SCNN) is proposed. In GCNN, weights of the network are updated in every epoch of training to minimize the classification error. In SCNN, the weighs are first computed using gradient descent algorithm as in GCNN, and then multiplied by scaling factor. Scaling factor is calculated using statistical measures, mean and standard deviation of the frames. Since statistical measures vary from video to video, scaling factor adapts to these changes. As the statistical information from the frames is directly used to alter the weights, it results in minimizing the error faster as compared to GCNN. Results of the proposed method prove that higher accuracy can be achieved with less number of epochs if scaling is used.

Keywords*: *Convolutional neural network, Deep learning, Human Action recognition

I. INTRODUCTION

Human action recognition is very eminent research area in computer vision domain. With increased use of CCTV cameras for video surveillance, requirement of automatic detection and recognition of human action has gained a lot of importance. Human action recognition plays an important role in the extraction of specific activity clips from the long duration videos. Human actions are divided as gestures like palm movement, simple actions like walking, interaction like two people shaking hands and group activity like many people walking together [1]. Methods developed for all these action recognition tasks vary as per the application.

Main challenges in the action recognition task are inter and intra class variations present in the action classes. Intra class variation is high as different people can perform same action in different ways. For example, one person can move his hand faster than other in waving action. Inter class variation is when actions belonging to two different classes generate similar features. For example, jogging action of one person can match with walking or running action of other person [2]. Oher than this, illumination changes, camera jitter, cluttered background, occlusion, different camera view angles also play a very important role in correct recognition of human action. In last few years, a lot of work is carried out on gesture recognition and simple action recognition where most of the time single person is present in a frame and performs the action. Work is also carried out to detect human-human interaction [3]. Most of the work carried out in the field of human action recognition and classification in previous years, uses handcrafted features and a machine learning approach. In machine learning approach, video is first represented using some features and then a classifier is trained to classify the test cases. Researchers have proposed numerous algorithms and methods to extract global as well as local features from a video. A global feature represents the frame as a whole and generally represents it in single vector. Local feature looks at the frame as a set of multiple patches. Local features are more robust to occlusion and clutter but computationally more complex than global features [4]. Many researchers have suggested use of combination of local and global features for increasing recognition accuracy [5]. Several classification algorithms like K-Nearest Neighbor, Support Vector Machine, Neural network etc. are used for classification purpose. Main drawback of using hand crafted features is the choice of features and selection of important features. In a video, spatial as well as temporal information is important for accurate recognition of action. So deciding which feature will be best suited for given video is difficult task. It is seen that, the type of feature to be used many times depend on the type of application [6]. Also the features extracted from a video are very large in number and have huge redundancy as very small changes occur in the background pixels. Selecting subset of features having maximum relevance is another tedious task in hand crafted feature method. Dimensionality reduction techniques like principal component analysis, Linear Discriminant Analysis, Generalized Discriminant Analysis, Independent Component Analysis etc. are used to remove redundant features and reduce computational cost of method. To overcome all these disadvantages, deep learning methods have been explored for human action recognition [7]. Use of deep learning methods for computer vision applications is a current hot area of research. Main advantage of deep learning is that it removes the tedious task of feature extraction and feature selection. After AlexNet, which implemented Convolution Neural Network (CNN), won the ImageNet contest in 2012 [8], CNN became most explored algorithm in deep learning field. Various forms and versions of CNN have been proposed by researchers for image classification. CNN architectures are further explored and applied to video data for application of human action recognition.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

Aditi Jahagirdar*, Department of IT, MIT College of Engineering, Pune, India. Email: aditi.jah@gmail.com

Manoj Nagmode, Department of E & TC, Government College of Engineering and Research, Avasari, India. Email: manoj.nagmode@gmail.com

In this work, a method using variation of convolutional neural network is proposed for human action recognition. Six layered CNN architecture is used for this task. CNN is configured as one input layer, two convolution layers, two pooling layers and one fully connected output layer. Frames of videos along with their action labels are given as input for training. Convolution layers use six filters of size 5 X 5. Two pooling layers use sampling rate of two to reduce the dimensionality. A fully connected output layer finally classifies the action.

In second step of work, a scaling factor is calculated from information contents of video frames and it is used to manipulate the weight terms calculated by back propagation algorithm. Mean and standard deviation of the frames of a video are used to compute the scaling factor. It is observed that, recognition accuracy increases and network reaches minimal error in less number of epochs as compared to GCNN with this method. For UT 1 interaction dataset, recognition accuracy increased from 89% with GCNN to 95.33% with SCNN.

The remaining paper is organized as follows. Section II gives related work. Background of CNN is given in section III. Proposed methodology is explained in section IV. Results are discussed in section V and paper concludes with section VI.

II. RELATED WORK

In many application of computer vision, classification of Human - human interaction and Human - object interaction is important for understanding events. Focus of this work is human –human interaction where most of the time one person moves and other responds.

In [9], Ijjina, Earnest Paul et.al. have proposed a method using genetic algorithm and CNN is proposed. Genetic algorithm is used to generate the weights used in CNN. Multiple iterations are used to reduce the classification accuracy.

Haodong Yang et al. in [10] propose attention mechanism based encoder-decoder framework. Long short-term memory (LSTM) method is used for attention mechanism. convolutional neural network(RACNN), which incorporates convolutional neural networks (CNNs),) and attention mechanism.

Sijie Song et al. in [11] propose spatio and temporal attention model based on recurrent neural network. The method uses long short term memory model which is trained to discriminate between joints in each frame. The model use regularized cross-entropy loss function to train the network.

Georgia Gkioxari et al. in [12] propose a method using person's pose, clothing, etc. to localize objects they are interacting with. Proposed method first jointly learns to detect people in frame. The method uses Recurrent CNN with a Feature Pyramid Network. In [13], Recurrent Neural Network based method is proposed which uses Long Short Term Memory concept and motion of human body joints. Contextual information is modeled using information from neighboring joints and previous frames. Stacked auto encoders are used in [14] to recognize human action. Segmentation is applied to extract region of interest from each frame. Harris detector is then applied to extract features from these regions. Histogram of region of interest is used as second feature. High accuracy is achieved by this method. In [15], 2D spatial convolution is applied to all the frames followed by 1D temporal convolution to achieve 3D convolution. Spatial information and motion information thus obtained is stacked. Multiple sampling of video is done to increase the sampling space. In [16], five RNNs are trained by considering five subdivisions of human skeleton. The outputs of the RNNs are then given to higher level layer using hierarchical structure. Single layer perceptron is used at final output layer.

III. BACKGROUND

CNN is extension of neural network with basic difference that, neural network works on a vector while CNN operates on volume. CNN is a multilayered network in which, basic layers are input layer, convolution layer, pooling layer and output layer. First layer of convolution layer can extract general features from the input. As detailed features are desired for classification application, dense CNNs are preferable. Dense CNNs are designed by using multiple convolution and pooling layers. Different values of kernel coefficients, sometimes called as weights, are used at each convolution layer for extracting different features. Parameters in convolution and fully connected output layer are trained using gradient descent algorithm. CNN used here has six layers. One input layer, two convolution layers, two pooling layers and one output layer. Fig. 1 shows diagrammatic representation of CNN used in this work.

A. Input Layer

Input layer is the first layer in CNN. No processing is done at this layer. Frames of the video are converted to gray scale and resized before giving as input to this layer. For this layers number of input maps and number of output maps is one. Input size is computed as in (1).

$$Inputmapsize = Height * Width * NumberofFrames \quad (1)$$

B. Convolution Layer

In convolution layer, input is convolved with filter coefficients to locate the features and store them as output map. Depth of the output map is equal to the number of filters or kernels used. Beauty of convolution layer is that, different filters can be applied to same input map in one convolution layer and different features can be found as different output maps. Every filter corresponds to particular feature of input map. For example, if three filters are applied to input map then, three different features will be available to represent same image as three output maps.

This property of convolution layer, makes it useful and different from hand crafted feature map. As kernels can't pass the edges of the frame, size of the output map is slightly less than the input map. Size of the output map is calculated as in (2).

$$outputmapsize = Inputmapsize - kernelsize + 1 \quad (2)$$

Retrieval Number: A1440109119/2019©BEIESP
DOI: 10.35940/ijeat.A1440.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

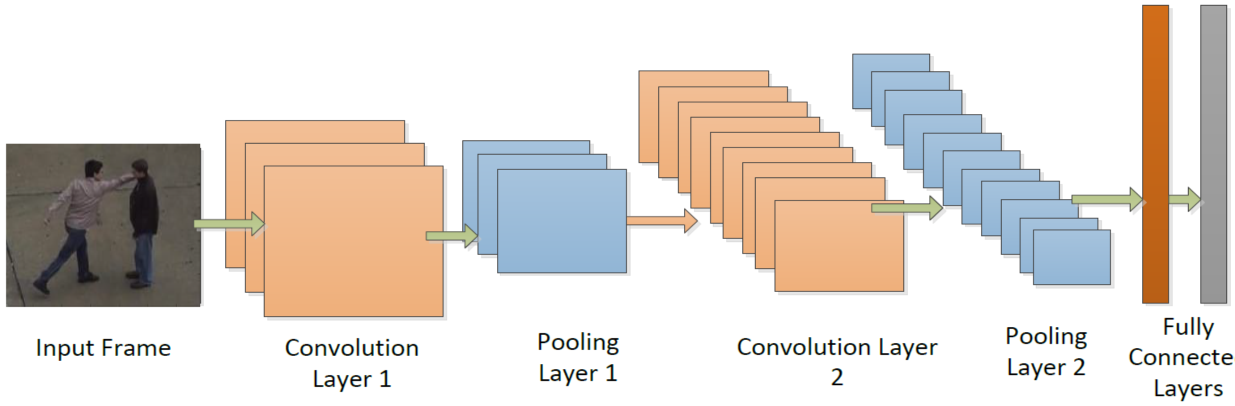

Fig. 1 Architecture of GCNN

In the first epoch, kernels values are randomly initialized. At every pass, kernel values are modified. After convolution operation, output map is passed through a nonlinear activation function. Logistic sigmoid and hyperbolic tangent functions are used after convolution in many applications. Recently, Rectified Linear units (Re-Lu) have become popular which combines nonlinearity and rectification in one function. Re-Lu function is easy to implement. It converts all negative values to zero which increases sparsity and robustness against small changes. Re-Lu function also reduces the likelihood of vanishing gradient problem in deep learning.

C. Pooling Layer

Pooling layer is also called as sub sampling layer. Output feature maps are sensitive to location of features in input map. To reduce the spatial dependency of the features, output maps of convolution layer are down sampled in pooling layer. This makes the feature robust against local translational variance. Pooling layer summarizes the contents of the patches of the feature map and represents it with one value. Minimum pooling, Maximum pooling and Average pooling are the most used pooling techniques. In minimum poling, smallest value present in the patch is selected to represent the patch. In maximum pooling, largest value is selected for representing the patch. In average pooling, average of feature values in particular patch is used to represent that patch. Minimum and average pooling can result in losing some of the distinct features identified by convolution layer. Maximum pooling preserves the distinct patterns identified by the convolution layer as features.

D. Fully connected layer

After passing the input through multiple convolution and pooling layers, it is given to fully connected output layer for classification. Fully connected output layer is nothing but neural network layer. Even if the size of the frame goes on decreasing after every pooling layer, number of output maps increase because of use of multiple kernels at every convolutional layer. Generally, depth of the output maps of the last pooling layer is considerably high. So output of last pooling layer is flattened before applying to first fully connected layer. Flattening is a process in which all the pooled output maps are converted to a single vector. There can be multiple fully connected layers in a network. Number of neurons in the last fully connected layer, which is also called as output layer, is equal to number of classes.

In this work, Softmax function is used in output layer to assigns one of the class to input image depending on highest score. Cross entropy loss is used as a loss function to calculate classification error.

Once the feed forward process is complete, error is calculated and back propagated to previous layers. Weights of the filters are updated in each iteration to minimize the classification error.

IV. METHODOLOGY

A. General CNN (GCNN)

In this work six layered CNN is implemented for human action classification. Number of video frames extracted from each video is kept constant to 60. Each frame is resized 56 * 56 pixels. Therefore size of input map for input layer is 56 *56 *60. In the first convolution layer, three filters or kernels of size 5 x 5 are applied. Each kernel convolves with the input map generating an output map. At this layer, output map size is 52 x 52 x 3. For first convolution layer, depth of the output map i.e. number of output maps is three. A pooling layer is used as a second layer to reduce the dimensionality. Max pooling method with stride of two is used in this work. Pooling is applied for every feature map generated by previous layer separately. Size of pooling filter here is 2 x 2 with stride of 2. Size of resulting output map is 26 x 26 x 3. In second convolution layer, six filters of size 5 x 5 are used. Each filter convolves with each of the 3thrre input maps generating 18 output maps. Size of the output map is 22 x 22 x 18. Second pooling layer with stride 2 and window size of 2 x 2 is applied to all 18 input maps. It generates output maps of size 11 x 11 x 18. One fully connected output layer is used with number of neurons equal to number of classes. After flattening of out of second pooling layer, vector of size 2178 is obtained which is used as input to fully connected layer. Multiple epochs are used to train the CNN for reducing the classification error. Parameters in convolution and fully connected output layer are trained using gradient descent algorithm.

B. Scaled CNN (SCNN)

In the proposed, SCNN approach, same six layered structure of GCNN, having one input layer, two convolution layers, two pooling layers and one output layer is implemented. In GCNN, weights and bias terms are upgraded in every epoch using back propagation algorithm.

In proposed adaptive SCNN, the weight terms are first computed as in GCNN using back propagation method using gradient descent algorithm to minimize the classification error. The weights computed for each level are then multiplied by the scaling factor. The scaling factor is computed from average mean and standard deviation of all the frames of a video. The scaling factor is given as in (3). Values of *k1* and *k2* are computed empirically.

$$s = k1 * Mean_{Avg} + k2 * STD_{Avg} \qquad (3)$$

Fig. 2 shows the block schematic of the proposed method used to calculate the scaling factor. The process of calculating scaling factor can be given in detail as:

1. Input a video
2. Convert the frames to gray scale
3. Find the mean of each frame
4. Find the standard deviation of each frame.
5. Take the average of means of all the frames, "*Mn*".
6. Take the average of standard deviations of all the frames, "*STD*".
7. Compute the scaling factor *s = k1 * Mn + k2 * STD*, where *k1* and *k2* are calculated empirically.
8. Multiply the weights calculated in back propagation algorithm by *s* in next epoch.

Mean value of each video is computed by taking the average of means of all the frames. As scaling factor is computed from intensity values of the frame, it is different for every video. As GCNN uses randomly initialized filter coefficients, it takes more time to find exact features which can give good recognition accuracy. In scaled CNN, in first epoch, filter coefficients are selected randomly. In every subsequent epoch, scaling factor computed from contents of the video is used to upgrade the weights. This helps the neural network to train in less number of epochs resulting in fast training of the network.

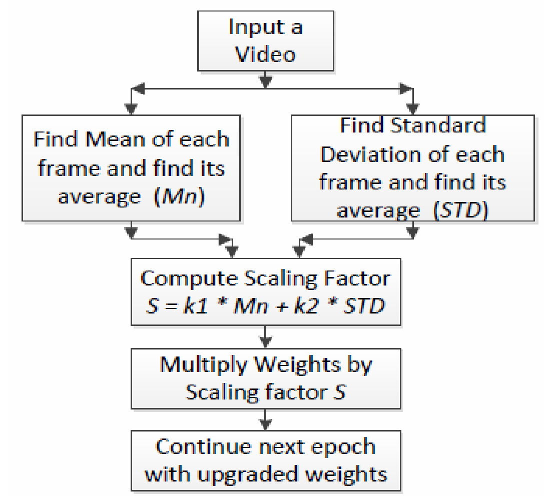
Input a Video
Find Mean of each frame and find its average (Mn)
Find Standard Deviation of each frame and find its average (STD)
Compute Scaling Factor S = k1 * Mn + k2 * STD
Multiply Weights by Scaling factor S
Continue next epoch with upgraded weights

Fig. 2 Block schematic of Calculation of Scaling factor

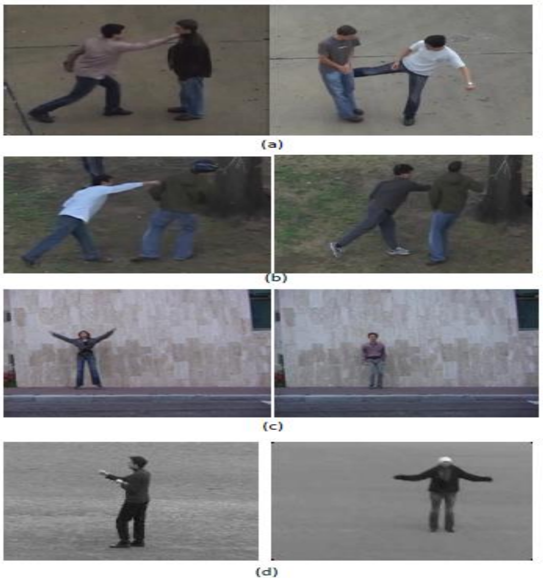

Fig.3 Sample frames from (a) UT1, (b) UT 2, (c) Weizmann and (d) KTH datasets

C. Datasets

As focus of this work is human action recognition, UT interaction dataset, Weizmann dataset and KTH dataset are used for evaluating the proposed method. Each of these dataset is having unique characteristics.

The UT-Interaction dataset contains videos of continuous executions of 6 classes of human-human interactions: shake-hands, point, hug, push, kick and punch. Several participants with more than 15 different clothing conditions appear in the videos. The videos are taken with the resolution of 720*480, 30fps, and the height of a person in the video is about 200 pixels. UT 1 interaction dataset is recorded in a parking lot where background is mostly static with very little camera jitter and only two actors appear in every frame. UT 2 interaction dataset is more complex. It is having cluttered background and recording is done with different illuminations. Also in many frames, pedestrians are also present along with the actors. Fig. 3 shows sample frames from all datasets.

Weizmann and KTH datasets are less complex datasets. Both these datasets have low resolution. Weizmann dataset is having 10 simple actions like bending, walking, running, jogging etc. performed by 9 different actors. It is recorded in controlled environment. Only one actor is present in each frame. Background used in the videos is uncluttered and remains almost constant. Camera angle and illumination is also constant for all the videos.

KTH dataset is much larger than other datasets used here. It is having approximately 600 videos. All the videos are recorded n controlled environment with one actor performing the action. Six simple actions like, boxing, walking, waving etc. are performed by 25 different actors in 4 scenarios. The actions are recorded indoor, outdoor, with changed view angle and changed illumination, increasing the complexity of the dataset.

Retrieval Number: A1440109119/2019©BEIESP
DOI: 10.35940/ijeat.A1440.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

D. Performance measures

As the ultimate objective of the proposed method is to recognize the input action class, recognition accuracy is used as a performance measure to compare the results obtained with the proposed method. Time required for training the network is another important performance parameter in deep learning methods. Time required to train a CNN depends mainly on number of iterations required to reach minimum classification error that is highest accuracy. To address this point, number of epochs required to reach maximum accuracy is used as other performance parameter.

For finding the recognition accuracy, tenfold validation is used. Stratified random sampling is used to avoid domination of any one class. Care is taken to keep equal number of samples of each target class in training and testing data.

VI. RESULTS AND DISCUSSION

The proposed SCNN method is evaluated using different scaling factors for all the used datasets. To calculate the scaling factor *s*, average mean and standard deviation values are computed from the input video. Optimal values of multiplying factors, *k1* and *k2*, are identified empirically. For evaluating the effect of scaling factor on recognition accuracy, *k1* value is changed from 0.1 to 1.5. *k2* is kept constant at 0.5. Graph in Fig. 4 shows effect of change in *k1* on recognition accuracy for all the datasets. It is observed that, for all the datasets, maximum recognition accuracy is achieved when *k1* value is in the range 0.7 to 0.8. Considering these results, 0.75 is used as the *k1* value in further experimentation.

To study the time required for training the network by the proposed SCNN method, recognition accuracy is calculated for different number of training epochs for all the datasets. More is the number of training epochs required for reaching the highest accuracy, more is the training time. Table 1 shows recognition accuracy obtained for UT1, UT 2, Weizmann and KTH datasets, using proposed SCNN method, when number of epochs are increased from 5 to 200. It is observed that, recognition accuracy increases fast when number of epochs are increased from 5 to 20. After 20 epochs, recognition accuracy decreases or remains almost same. This shows that using proposed SCNN method, high accuracy is achieved with less number of epochs thus reducing the time required.

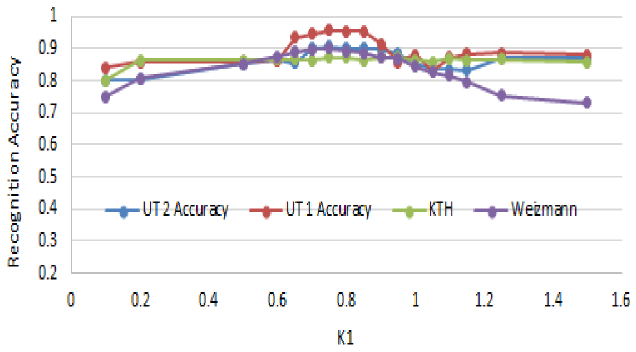

Fig. 4 Recognition accuracy obtained for various values of *k1*

Table 1: Recognition Accuracy for different number of epochs

No. of epochs	% Recognition Accuracy			
	UT1	UT2	Weizmann	KTH
5	94.17	90.84	78.06	84.86
10	95.33	91.45	84.44	85.28
20	94.92	90.63	86.94	81.74
50	92.83	90.08	90.00	81.74
100	88.08	88.28	89.17	81.24
200	86.42	86.06	88.82	80.64

It is observed that recognition accuracy achieved for Weizmann and KTH datasets is less as compared to that achieved for UT interaction dataset. As described earlier, CNN extracts the features based on spatial information of the video. At the beginning, the filter coefficients are initialized randomly and then tuned in every iteration to minimize the classification error. As the videos in Weizmann and KTH datasets are recorded with low resolution, less spatial information is available in these videos. This results in extraction of weak features which ultimately effect in high classification error. Handcrafted features have proved to give better accuracy for these datasets than deep learning methods.

Comparison of proposed SCNN method is done with the simple GCNN method on the basis of number of training epochs required for reaching maximum accuracy. Fig. 5 shows comparative graph of recognition accuracy achieved for different number of epochs with GCNN and SCNN on UT 1 interaction dataset. Similar results are obtained for other datasets.

The comparison in Fig. 5 shows that, higher recognition accuracy is achieved with less number of epochs when SCNN method is used. For GCNN, 200 training epochs are required for obtaining 91% recognition accuracy with training time of 1009.12 seconds. Whereas with SCNN, 95.33% recognition accuracy is achieved with training time of 49.51 seconds. It proves that weights get tuned faster with SCNN method as spatial information in the form of statistical parameters is directly used to manipulate them. Thus reducing the time required for training the CNN.

Comparison of GCNN and proposed SCNN methods, based on recognition accuracy achieved for all the used datasets is given in Table 2. Maximum accuracy achieved for each dataset is considered for the comparison.

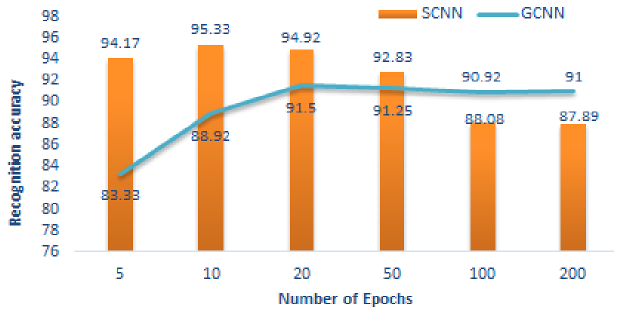

Fig. 5 Recognition Accuracy for GCNN and SCNN based on number of epochs required

Table 2: Comparison of Recognition accuracy with GCNN and SCNN

Dataset	% Recognition Accuracy	
	GCNN	SCNN
UT 1	91.50	95.33
UT 2	83.75	91.45
Weizmann	88.06	90.00
KTH	86.25	85.28

It is observed that higher recognition accuracy is achieved for all the datasets with proposed SCNN method.

V. CONCLUSION

In this work, a variation of CNN, called SCNN, is proposed for human action recognition application. Well known datasets, UT interaction, Weizmann and KTH are used for evaluating the new approach. Six layered convolution neural network is used for classifying the human actions.

In the first epoch of the CNN, filter weights are initialized randomly and tuned in every subsequent epoch to reduce the classification error. In proposed SCNN method, these weights are further upgraded by multiplying with a scaling factor. The scaling factor used is computed from average mean and standard deviation of the video frames. The hyper parameters *k1* and *k2*, used in the scaling factor calculation are computed empirically. Effect of change in *k1* value is studied for all four datasets to finalize its value.

It is seen that, less recognition accuracy is obtained for Weizmann and KTH datasets as compared to UT interaction dataset. As the videos in these datasets have low resolution, features extracted from these videos are not strong enough to classify the actions.

It is observed that, with SCNN, higher recognition accuracy is obtained with less number of training epochs, thus reducing the time required for training. Maximum accuracy of 95.33% is obtained for UT interaction dataset with just 10 epochs when SCNN is used.

REFERENCES

1. Arunnehru, J., G. Chamundeeswari, and S. Prasanna Bharathi. "Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos." Procedia computer science 133 (2018): 471-477.
2. Yuan, Yuan, Lei Qi, and Xiaoqiang Lu., "Action recognition by joint learning", Image and Vision Computing Vol. 55, Part 2, pp. 77-85, November 2016.
3. Nikzad, Saman, and Hossein Ebrahimnezhad. "Two-person interaction recognition from bilateral silhouette of key poses." Journal of Ambient Intelligence and Smart Environments 9, no. 4 (2017): 483-499.
4. Lisin, Dimitri A., Marwan A. Mattar, Matthew B. Blaschko, Erik G. Learned-Miller, and Mark C. Benfield. "Combining local and global image features for object class recognition." In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops, pp. 47-47. IEEE, 2005.
5. A. S. Jahagirdar, M. S. Nagmode, "Human Action Recognition using Ensemble of Shape, Texture and Motion features," International Journal of Pure and Applied Mathematics,Vol.119, No.12, pp 13025-13032, 2018.
6. Baccouche, Moez, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. "Sequential deep learning for human action recognition." In International workshop on human behavior understanding, pp. 29-39. Springer, Berlin, Heidelberg, 2011.
7. Kim, Ho-Joon, Joseph S. Lee, and Hyun-Seung Yang. "Human action recognition using a modified convolutional neural network." In International Symposium on Neural Networks, pp. 715-723. Springer, Berlin, Heidelberg, 2007.
8. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.
9. Ijjina, Earnest Paul, and Krishna Mohan Chalavadi. "Human action recognition using genetic algorithms and convolutional neural networks." Pattern recognition 59 (2016): 199-212.
10. Yang, H., Zhang, J., Li, S. and Luo, T., Bi-direction hierarchical LSTM with spatial-temporal attention for action recognition. Journal of Intelligent & Fuzzy Systems, (Preprint), pp.1-12.
11. Song, S., Lan, C., Xing, J., Zeng, W. and Liu, J., 2017, February. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. In AAAI (Vol. 1, No. 2, pp. 4263-4270).
12. Gkioxari, G., Girshick, R., Dollár, P. and He, K., 2018, June. Detecting and recognizing human-object interactions. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8359-8367). IEEE.
13. Liu, Jun, Amir Shahroudy, Dong Xu, and Gang Wang. "Spatio-temporal LSTM with trust gates for 3D human action recognition." In European Conference on Computer Vision, pp. 816-833. Springer, Cham, 2016.
14. Berlin, S. Jeba, and Mala John. "Human interaction recognition through deep learning network." In 2016 IEEE International Carnahan Conference on Security Technology (ICCST), pp. 1-4. IEEE, 2016.
15. Berlin, S. Jeba, and Mala John. "Human interaction recognition through deep learning network." In 2016 IEEE International Carnahan Conference on Security Technology (ICCST), pp. 1-4. IEEE, 2016.
16. Du, Yong, Wei Wang, and Liang Wang. "Hierarchical recurrent neural network for skeleton based action recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1110-1118. 2015.

AUTHORS PROFILE

Ms. Aditi S. Jahagirdar, received her Bachelor's degree in 1993 in Electronics & Telecommunication engineering and Master's degree in Electronics Engineering in 1999from University of Pune, Maharashtra, India. She is currently working as Assistant Professor in Department of Information Technology at MIT College of Engineering, Pune. She is having total 21 years of teaching experience. She is currently working towards her Ph.D. degree in Electronics Engineering in Savitribai Phule University, Pune, India. She is life member of 'Computer Society of India' and 'Institution of Electronics and Telecommunication Engineers'. Her research interests include computer vision, Image processing, video processing and pattern recognition.

Dr. Manoj S. Nagmode, is currently working as Professor and HoD in Government College of Engineering and Research, Avsari, Pune, India, in the Department of Electronics and Telecommunication Engineering. He is having teaching experience of more than 22 years. He has published more than 45 research papers in various reputed journals and conferences. He is an author of lambert and Taylor and Francis series book based on research methodology. He has worked in different committees for the evaluation of research papers. His areas of interest include Image processing, Signal processing and Embedded systems. He was awarded Ph.D. degree by University of Pune, India in 2009 in Electronics and Telecommunication Engineering in the domain of Video Processing.