Human Action Recognition

[1]M.Muthu Lakshmi, [2]Dr.P. Arockia Jansi Rani
[1]Research scholar, [2]Associate Professor
Computer Science and Engineering Department,
Manonmaniam Sundaranar University, Tirunelveli, India

***Abstract* - Human Action Recognition plays significant role in various computer vision applications such as Face Recognition, Speech Recognition, Optical Character Recognition, Traffic Sign Recognition, Finger Print Recognition Etc. The Human Action Recognition problem consists of two stages like Feature Extraction and Classification. The objective of this work is to create a test dataset for various human actions and also to validate system for real time videos. System is proposed to analyze the role of ITL, CNN, and KNN for the extraction of both spatial and temporal features. The system's response has been validated using the real time videos.**

***Index Terms* - Action Recognition, Spatial Feature, Temporal Feature, Internal Transfer Learning, Convolutional Neural Network, K-Means nearest Neighbor Algorithm.**

I. INTRODUCTION

Human Action Recognition refers to the classification of human actions that is present in video. The action detection involves locating actions in space. Classifiers are used to identify the action class and their Spatial, Temporal locations. The Human Action Recognition process consists of two stages like, Extracting Features then Classifying the extracted Features. Human Action Recognition is a model of deep learning technique. **Deep learning** algorithms are used to learn the Image Recognition problem and to classify input Videos or images into appropriate categories. Every process in Image Processing starts with preprocessing step. Preprocessing is an improvement of image that removes unwanted details or enhances some image features for further processing. After preprocessing a video the features are extracted from a video. The Spatial & Temporal Features are extracted for further processing. **Spatial features** are consists of x, y coordinate values. **Temporal features** are stores data related to past, present, future time. Convolutional Neural Network (CNN) as a feature extractor for training an image classifier. **ITL** is a combination of Transfer Learning and sub data classification methods. Transfer learning is used to train the data. The ITL Algorithm is fed to KNN and CNN for the classification purpose.

II. METHODOLOGY

After the Frames extraction and preprocessing a video, the Feature extraction process begins. Feature extraction is the process of variable selection. It is the selection of attributes from the data. Feature Extraction process include extraction of Spatial and Temporal Features. For the extraction of spatial feature the optical flow method is used. For the extraction of temporal feature the gradient method is used. **Optical flow** or **optic flow** is the pattern of motion in objects, surfaces, and edges caused by the relative motion between an observer and a scene. The optical flow methods try to calculate the motion between two image frames which are taken at t times and voxel position. This voxel position is placed on the approximations of the image signal. Optical Flow contains two coordinates Vx & Vy. The functions of Vx & Vy assigned to images, alpha & iterations of the optical flow method. For the optical flow determination we can used the Horn-Schunck method. Temporal gradient filter is used with Lucas-Kanade algorithm for extracting temporal features. This is for to perform the Gaussian derivation. The temporal gradient filter used by the Lucas-Kanade algorithm. The extracted features are fed into ITL for training process. ITL is used with N Class, the classification process divided the class into several ones. The class of KNN Classify method consists of sample, training and group. The sample consists of those matrixes whose rows will be classified into groups. The number of columns of sample is equal to the number of columns of training. The rows of matrix are grouped in the sample class. Training also has the same number columns as sample. The rows of training are grouped with the group vector value. The optional value k is the nearest neighbors used in the classification. CNN requires a large amount of labeled training data to be effective. A transfer learning method of training a CNN with available labeled source data and then extracting the CNN internal layers to a target CNN learner. This method is referred to as the transfer convolutional neural network (TCNN). To correct for any further distribution differences between the source and the target domains, an adaptation layer is added to the target CNN learner, which is trained from the limited labeled target data.
The experiments are run on the application of object image classification where average precision is measured as the performance metric. Train a 6c-2s-12c-2s Convolutional neural network which has six convolutional layers, two sampling layers, twelve convolutional layers, and two sub sampling layers.

III. RESULTS AND DISCUSSION

The human action recognition table consists of following values for input video, Frame ID (Order of frames from 1 to 60), Width (Width of the each frame), Height (Height of the each frame), Weight (Weight of the each frame), Mean (Mean value of each frame), STD (Standard deviation value of each frame), Average (Average value of each frame), Moving direction of the frames from starting to end of the frames of the video. Frame Based Action Performance for STD, Mean, and Weight values as follows:

Table 1: Frame Based STD Value

Frame Based Action Performance (Std Value)						
Frame Id	Bend	Jump	Wave	Walk	Run	Jump With Run
Frame1	46.268	33.933	32.674	31.065	35.836	40.429
Frame2	45.284	35.430	31.662	32.043	33.417	41.714
Frame3	45.708	37.704	31.641	33.417	36.435	41.688
Frame4	46.803	37.734	33.377	31.505	37.848	40.311
Frame5	45.650	35.248	32.665	32.347	33.484	40.828
Frame6	48.941	36.268	33.573	35.464	29.917	41.044
Frame7	46.082	32.121	34.293	36.435	30.048	42.274
Frame8	44.118	30.913	33.405	33.058	32.043	40.643
Frame9	44.372	28.665	34.317	37.048	31.085	40.065
Frame10	45.986	33.767	38.261	38.612	32.347	39.807

Table 2: Frame Based Weight Value

Frame Based Action Performance (Weight)						
Frame Id	Bend	Jump	Wave	Walk	Run	Jump With Run
Frame1	109	114	256	413	932	725
Frame2	126	124	260	418	935	730
Frame3	130	128	277	450	947	731
Frame4	136	139	278	485	956	774
Frame5	144	142	282	559	1043	782
Frame6	148	145	281	602	1058	776
Frame7	151	148	289	665	1106	780
Frame8	175	152	301	668	1156	818
Frame9	192	248	305	678	1352	845
Frame10	209	248	305	694	1620	868

Table 3: Frame Based Mean Value

Frame Based Action Performance (Mean Value)						
Frame Id	Bend	Jump	Wave	Walk	Run	Jump With Run
Frame1	0.722	0.291	0.250	0.345	0.352	0.458
Frame2	0.727	0.311	0.234	0.309	0.272	0.481
Frame3	0.712	0.344	0.237	0.345	0.345	0.468
Frame4	0.777	0.343	0.283	0.352	0.361	0.444
Frame5	0.765	0.315	0.258	0.305	0.305	0.460
Frame6	0.838	0.334	0.285	0.361	0.236	0.461
Frame7	0.802	0.273	0.277	0.352	0.231	0.497
Frame8	0.711	0.261	0.288	0.368	0.255	0.462
Frame9	0.788	0.223	0.275	0.313	0.247	0.442
Frame10	0.819	0.283	0.380	0.321	0.298	0.429

Overall performance				
	KNN Accuracy %		KNN Time_Taken	
Action Process	Real_Time	Data set	Real_Time	Data set
Bend	71.745	92.259	0.414	0.672
Jump	90.560	93.757	0.424	0.689
Wave	90.990	95.339	0.406	0.321
walk	84.102	84.072	0.424	0.636
Run	87.223	85.405	0.392	0.724
Jump With	84.134	88.555	0.395	0.628

Overall performance				
	CNN Accuracy %		CNN Time_Taken	
Action Process	Real_Time	Data set	Real_Time	Data set
Bend	92.502	95.701	0.151	0.196
Jump	95.345	97.431	0.148	0.207
Wave	96.331	98.273	0.145	0.147
walk	85.050	86.973	0.152	0.196
Run	89.264	86.120	0.156	0.160
Jump With Run	92.169	88.812	0.147	0.171

Run				

Figure : Over All Performance for CNN & KNN

IV. CONCLUSION

In this paper, I focused on the human action recognition problem. I utilize the Convolutional Neural Network to automatically extract both spatial and temporal features. To avoid the difficulty of training data I utilize the Internal Transfer Learning (ITL) algorithm. My method achieves better results for CNN compared with KNN Classifications.

V. REFERENCES

[1] Sreeman Ananth Sadanand & Jason J. Corso , Computer Science & Engineering, SUNY at Buffalo, *" A High Level Representation Of Activity In Video", 2012.*

[2] Shuiwang Ji , Weixu, Ming Yang, Member IEEE and Kaiyu Member IEEE, *"3DConvolutional Neural Network for Human Action Recognition" ,2013.*

[3] Karinne Ramirez- Amaro , Eun Sol Kim, Jiseob Kim, Byoung Tak Zhang, Michael Beetz & Gordon Cheng , *"Enhancing Human Action Recognition Through Spatio-Temporal Feature Learning", 2013.*

[4] Xue Wen Chen , *"Big Data Deep Learning, Challenges & Perspectives",2014*

[5] Linsun, Kuijia, Dit yan yeung University of Hong Kong, *"Human Action Recognition Using Factorized Spatio-Temporal Convolutional Network ,2015.*

[6] Ling Shao, Fan Zhu*, "Transfer Learning For Visual Categorization, A Survey", 2015.*

[7] Rajendra Kumar *"Human Action Recognition, A Survey", 2012.*

[8] Jubel, Sonia *"Applications and challenges of Human Action Recognition using sensors", 2015.*

[9] Rana, Tanya Jha, Rashmi Shetty *"Machine Learning Techniques in Human Activity Recognition", 2015* [10]Deepika, Sowmya, Soman *"Image Classification Using Convolutional Neural Network"*,2014.