A Video Based Human Detection and Activity Recognition – A Deep Learning Approach

Moloy Dhar[1], Bidesh Chakraborty[2]

[1]Dept. of CSE, Guru Nanak Institute of Technology, India
[2]Dept. of CSE, Haldia Institute of Technology, India

ABSTRACT

Humanactiondetection and identificationhasawiderangeofapplications,suchasvideostorageandretrieval, intelligentvideosurveillance andenvironmentalhomemonitoring,intelligenthuman–machine interfaces and identity recognition which targets many research topics in computer perception, including human detection in video, human pose estimation, human tracking, and analysis and understanding of time series data.The Human Activity Recognition System (HARS) tries to classify activities based on a series of observations of many subjects' actions and a diversity of environmental variables.The purpose of this research work is to first investigate and compare the accuracy of various HARS for different human actions shown in videos, and then to offer a superior solution. In this paper 6 categories of human activities (jogging, hand waving, walking, running, and handclapping, and boxing) have been recognized with a mean average precision (mAP) of 79.33% at the frame-based and 84.4% at the image-based measurement on the HAR datasets.Extensive experiments on dataset shows that the suggested approach outperforms the current state-of-the-art in action recognition.

Keywords - CNN algorithm,Deep Learning, HAR datasets, Feature Learning

I. INTRODUCTION

Recognition of human activity is important in human-to-human interaction and interpersonal relationships. It is very tough to extract since it contains information about a person's identity, personality, and psychological condition. One of the key objects of study in the scientific fields of computer vision and machine learning is the human ability to recognize another person's activity. Many applications, including as video surveillance systems, human-computer interaction, and robotics for human behaviour characterization, now require a multiple activity recognition system as a result of this research. The human activity methods can be classified as unimodal and multimodal activity methods according to the nature of sensor data they employ.

Unimodal methods, which represent human activities from data of a single modality, such as images, can be further categorized as: (i) space-time, (ii) stochastic, (iii) rule-based, and (iv) shape-based methods.

Space-time methods represent human activities as a set of spatiotemporal features [1] or trajectories [2, 3]. Stochastic methods acknowledgehuman activities or actions by applying different statistical (e.g., hidden Markov models) [4]. Rule-based methods consists of a set of rules to define human activities [5, 6]. Shape-based methods efficiently represent activities with high-level reasoning by modeling the motion of human body parts [7, 11].

Multimodal methods, on the other hand combinesdifferent features collected from different sources [9] and are classified into three categories: (i) affective, (ii) behavioural, and (iii) social networking methods.

In affective methods the experience of feeling the underlying emotional state of a person and the emotional communications are the key parameters to represent human activities [10]. Behavioural methods targets to extractsbehavioural attributes, non-verbal multimodal cues, such as gestures, facial expressions, and auditory cues from the images and using all the extracted parameters it recognise human activity [3]. Finally, social networking methods model the characteristics and the behaviour of humans in several layers of human-to-human interactions in social events from gestures, body motion, and speech [10].

Usually, the terms "activity" and "behaviour" are used interchangeably in the literature [12]. We differentiate between these two terms in the sense that the term "activity" is used to describe a sequence of actions that correspond to specific body motion. On the other hand, the term "behaviour" is used to characterize both activities and events that are associated with gestures, emotional states, facial expressions, and auditory cues of a single person.

In this paper, a model has been proposed to identify the human actions, as the actions are classified into different types like running, jogging, walking, handclapping, hand waving, etc.The recognition of human actionsdatabase consist of six types of human actions (running, jogging, walking, hand clapping, hand waving, and boxing) implemented assorted times by 25 subjects in four distinct schemes: outdoors s1, outdoors with scale variation s2, outdoors with different clothes s3 and indoors s4 as illustrated in Fig. 1. Presently, the database accommodates 2391 arrangements. Here, all arrangements were captured over similar circumstances by a motionless camera with 25fps frame rate. The arrangements were down sampled to the dimensional arrangement of 256 x 256 pixels and have a duration of four seconds in standard.

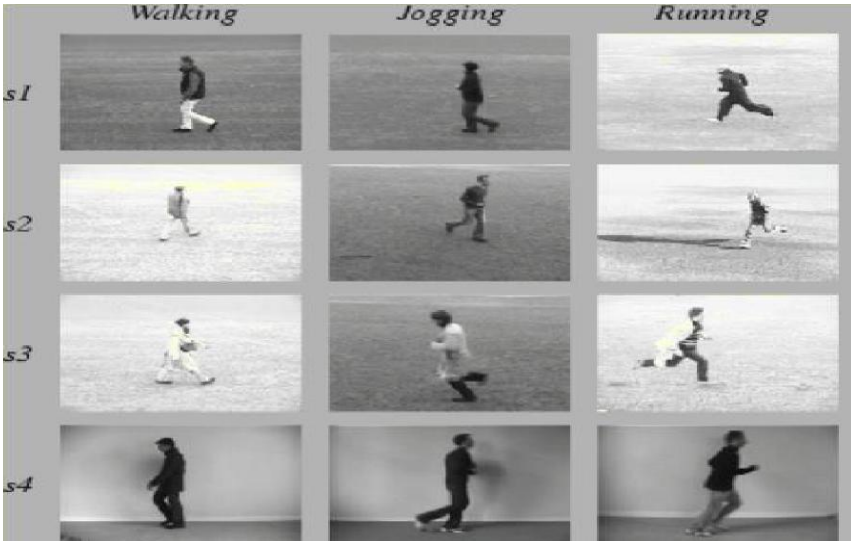


Figure 1. Training Sample RE-LU matrix

II. METHODOLOGY

The tasks involved are the following:

✓ Downloading, extracting and pre-processing a video dataset

✓ Divide the dataset into training and testing data

✓Create a neural network and train it on the training data

✓ Using test data for test the model

✓ Compare the performance of the model with some pre-existing models

II.I Convolutional Neural Networks (CNN)

A convolutional neural network (CNN) is a form of artificial neural network that is specifically intended to process pixels as inputs and is used in image recognition and processing. The underlying architecture of CNN is shown in Fig. 2.

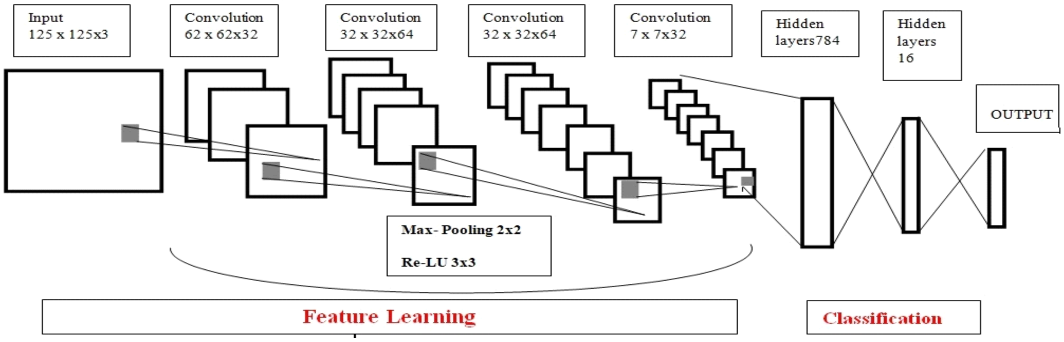
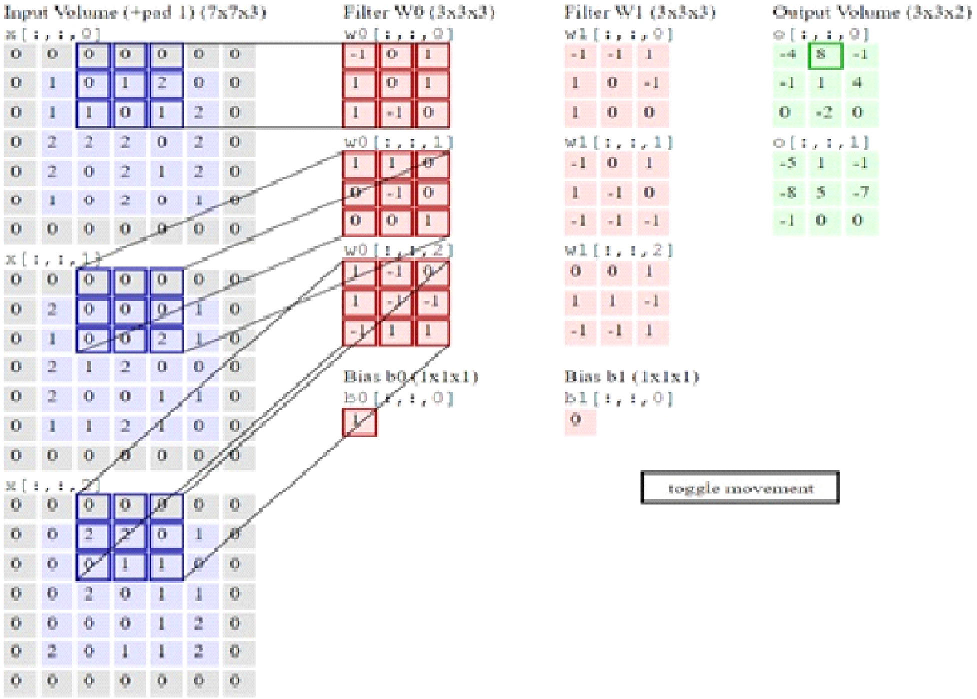


Figure 2. Analyzing image in forms of matrix

Fig. 2 is detached into domains, and each domain is then appointed to distinct unseen nodes. Each unseen node discovers arrangement in only one of the domains in the image. This domain is resolute by a kernel i.e. filter/window. The filter is convoluted in addition to the pair of x-axis and y-axis. Numerous filters are recycled in form to fetch various arrangements from the image. Output of one filter when convoluted during the whole of the unified image produces a 2-D zone of neurons i.e. called feature map. Each of the filter is accountable for one features map [14, 15].

These feature maps can be pushed into a 3-D array, that can be used as the input to the zones. It is accomplished by the zones noted as Convolutional layer in a CNN. These zones are pursue by the Pooling layers, that decrease the spatial aspects of the turnout (gained from the convolution layers) i.e., a window is drift in both the axes and the maximum value in that filter/window is taken (Max-Pooling layer). Repeatedly, average pooling layer is also used where the only dissimilarity is to endure the average value inside the window alternatively of the maximum value. Therefore, the convolutional layers boost the depth of the input image, when in fact the pooling layers declines the spatial aspects (peak and span). Significance of such design is that it encrypts the ease of an image that can be leveled into a 1-D array.


Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	1	0	1	2	0	0
0	1	1	0	1	2	0
0	2	2	2	0	2	0
0	2	0	2	1	2	0
0	1	0	2	0	1	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	2	0	0	0	1	0
0	1	0	0	2	1	0
0	2	1	2	0	0	0
0	2	0	0	1	1	0
0	1	1	2	1	0	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	0	2	2	0	1	0
0	0	0	1	1	0	0
0	0	2	0	1	1	0
0	0	0	0	1	2	0
0	2	0	1	1	2	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	0	1
1	0	1
1	-1	0

w0[:,:,1]

1	1	0
0	-1	0
0	0	1

w0[:,:,2]

1	-1	0
1	-1	-1
-1	1	1

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

-1	-1	1
1	0	-1
1	0	0

w1[:,:,1]

-1	0	1
1	-1	0
-1	-1	-1

w1[:,:,2]

0	0	1
1	1	-1
-1	-1	1

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

o[:,:,0]

-4	8	-1
-1	1	4
0	-2	0

o[:,:,1]

-5	1	-1
-8	5	-7
-1	0	0

Figure 3. Image analysis in form of matrix by adding bias to the actual image

II.I Model Parameters

In each convolutional layer, (Fig. 3) are configured with the following parameters:

- Filters - Number of feature maps needed as the output of that convolutional layer.
- kernel_size - Size of the window that will get convoluted on entirely the axes of the input data to generate a single feature map.
- strides - Number of pixels by which the convolutional window will drift by.
- padding - To determine what appears on regression models.

One of the most important part of this paper was to load the video dataset and perform the necessary pre-processing steps. So, we developed a class (Videos) that had a function called (read_videos()) that can be used to read and process videos. Creating this was very challenging as we concentrated on generalizing this function for any kind of videos. We have used NumPy (wherever) for storage and processing of the videos (much faster than in-built python lists with a ton of extra functionalities). The neural network was implemented using Keras.

III. Data Exploration

The dataset can be obtained from Recognition of Human Actions dataset.The video dataset contains six types of human activities (handclapping, hand waving, boxing, jogging, running and walking) performed several times by 25 different subjects in 4 different scenarios - outdoors , outdoors with scale variation , outdoors with different clothes and indoors. The model will be constructed irrespective of these scenarios. The videos were captures at a frame rate of 25fps and each frame was down-sampled to the resolution of 160x120 pixels where the dataset contains 300 videos – 100 videos for each of the 3 categories (Jogging, Running, and Walking).Fig. 4 illustratesfew sample frames for some videos from the dataset.

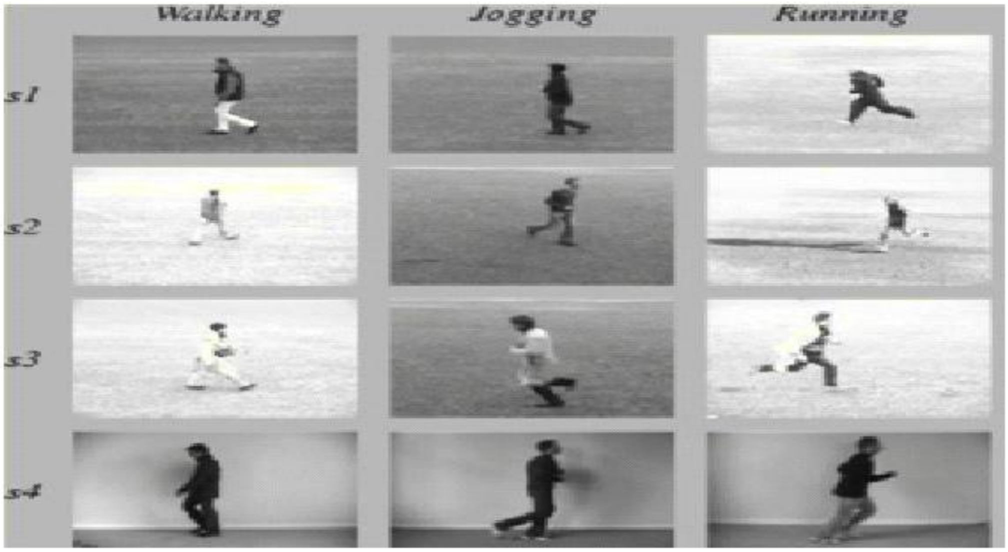


Figure 4. Training Sample RE-LU matrix

III. Algorithm and Techniques

The algorithm contains 4 steps. The dataset has been split into train and test dataset in step 1. In step 2, we have built the CNN model. Model compilation has been done in Step 3 and in step 4 the model can be tested with different inputs and accuracy will be measured.

```
Algorithm:
    Step 1: Preparing train and test dataset:-
        1. Count = -1, train_image=[ ], train_labels=[ ]
        2. For file in dataset:
           Randomly dividing the whole data into training (66.67%) and testing (33.33%) data
```

```
                    a.  train_files, test_files, train_targets, test_targets = train_test_split(files, targets,
                        test_size=1/3, random_state=191)
            3.  The categorical labels are converted into integers.:
                    a.  for label in zip(range(6), raw_data['target_names']
            4.  Load a sample of the training data as it is :
                    a.  sample_files = train_files[:1]
                        reader = Videos(target_size=None, to_gray=False)
                    b.   Loading the sample videos, in their original format
                        sample = reader.read_videos(sample_files)
                    c.  The shape of the tensor obtained is (1, 515, 120, 160, 3).
                            i.  Select the same number of frames from each video.
                                    Extract the N frames from the first ofeach video:
                                        Videos(target_size=(128, 128),
                                        to_gray=True, required_fps=5,
                                        max_frames=200, extract_frames='first')
                    d.  Pixel Normalisation(to ensure each input parameter;pixels, in this casehas a similar
                        data distribution, this makes convergence faster while training the network.)
                            i.  range[-1,1]=[0,…,225]
                           ii.  Videos(target_size=(128, 128),to_gray=True,
                                max_frames=40,required_fps=5, normalize_pixels=(-1, 1))
                    e.  The shape of the training data is (300, 200, 128, 128, 1)
        Step-2: Building the Cnn model:
            model = Sequential()
        # Adding convolutional and pooling layers
Step-3:- Model Compilation
            1.  40 epochs on the training data and saving the weights of the model that performed the best on
                the validation data.
                    model.compile(loss='categorical_crossentropy', optimizer='nadam', metrics=['accuracy']).
            2.  Using 'NADAM' optimizer.
Step-4:- Select the videos of 9 persons (randomly) performing each of these actions and predicting the outcomes of
those frames by our model to check.
```

IV. Proposed Model

From each video, 200 contiguous frames (8 seconds) were being extracted and given as input to the model. As, we know that the human body performs these activities (running, boxing etc.) with a certain speedso, within one second, the human body does not make much of a movement. Therefore, we do not need to collect every frame for each second of video that we are capturing. A different approach could be used, where only a certain number of frames are extracted for each second.

Now, we would be extracting only 5 frames per second (first 5 frames for each second). So, suppose we have a video of 10 seconds, we will get 10 x 5 = 50 frames. There is also a maximum limit on the number of frames that should be extracted from each video. We have set this value to 40. So, these 40 frames will be selected from the front of the extracted frames.

The range of normalized pixels has also been changed from [0, 1] to [-1, 1]. This is because the mean of the pixels would then be 0, which (Fig. 3.) would help the model converge faster.

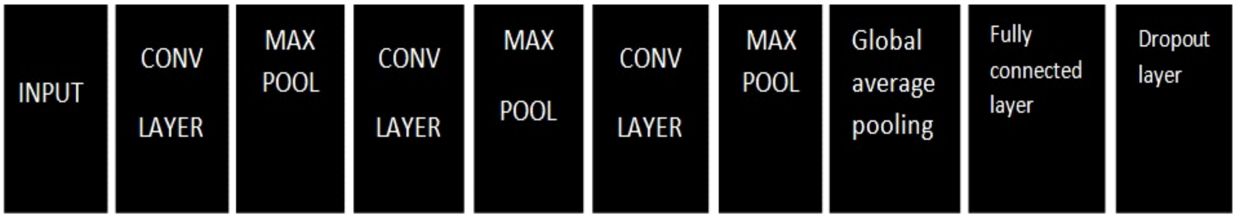


Figure 5. Model Architecture

The model (Fig. 5.) was trained on the training data for 40 epochs. The weights of the model which provided the best performance on the validation data were loaded. The model was then tested on the test data.The model gave an accuracy of 64.5% on the test data. This model gave a higher accuracy than the previous models, despite using 5 times lesser data for training.

We have used NADAM as the optimizer (instead of ADAM). In Keras, the default learning rate for ADAM optimizer is set to 0.001 whereas for NADAM, the default value of learning rate is 0.002 and there is a scheduled decay of learning rate.Using NADAM as the optimizer, the model produced better results than ADAM. Also, at the end of 40 epochs, the model did not overfit when the optimizer was NADAM, but in case of ADAM, the model showed some signs of over fitting.

V. Benchmark Model and Results

The model proposed in [13] gave the highest accuracy on the test data (74.5%). Fig. 6 is the learning curve of the model over 40 epochs. Here we chose the model weights that performed on the validation set and provides best accuracy on the test data.

Here we select [13, 14, 15] as benchmark model and compare our proposed model with the notion of local features in space-time to acquire and define local actions in a video. The general idea is to describe such events to define several types of image descriptors over local spatiotemporal neighbourhoods and evaluate these descriptors in the context of recognizing human activities. These points have stable locations in space-time and provide a potential basis for part-based representations of complex motions in video.

The benchmark model (Fig. 7.) was able to achieve an overall recognition rate of 80-85%. But, in order to compare the benchmark model with the proposed model, the confusion matrix of the benchmark model will be analyzed with the confusion matrix of the proposed model.

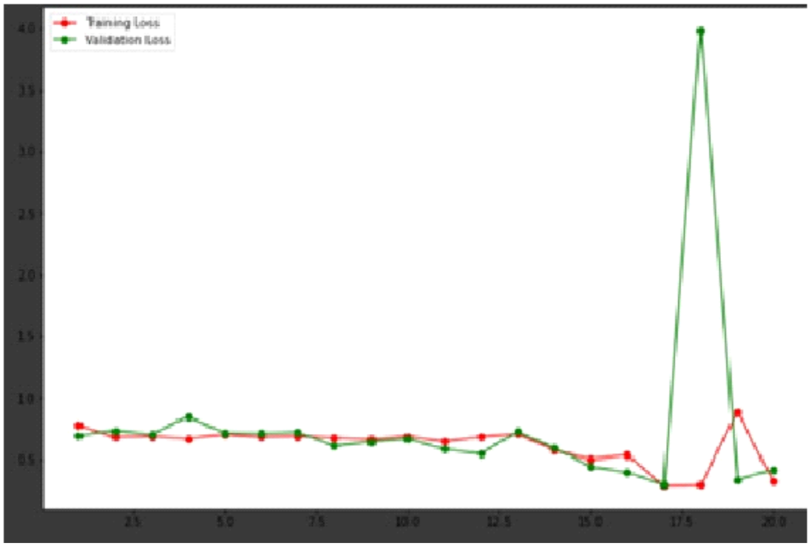


Figure 6. Training – validation graph over 40 epoch cycles

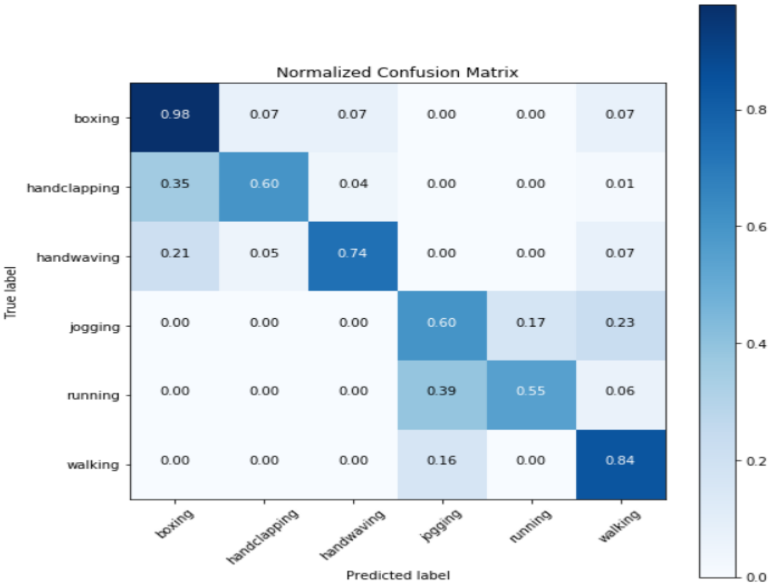


Figure 7. Confusion matrix of benchmark model [1]

The confusion matrix of the benchmark model [13] as well as the proposed model have been converted in the same format. Also, the confusion matrix has been normalized. The confusion matrix of the proposed model is shown in Fig. 8.The comparison results are addressed in Table 1, where a mean average precision (mAP) of our model is 79.33% whereas mean average precision of benchmark model is 65%.Fig. 9 shows how our model outperformed the benchmark model [13].1 and 2 from each Fig. 9 represents the accuracy from the benchmark model and our model respectively with different activities. It has been observed that our model got a very good accuracies in walking, jogging, and running activities. Handwaving, Handclapping, boxing activities are predicted equally as predicted in model [13].

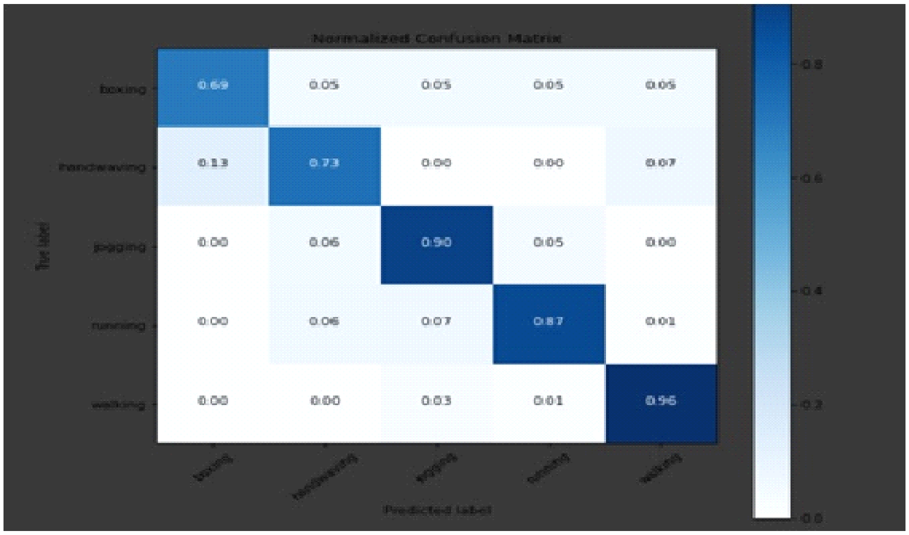


Figure 8. Confusion matrix of the Proposed Model

ACTIVITY CLASS	BENCHMARK ACCURACY [13]	PROPOSED ACCURACY
Walking	0.84	0.96

Jogging	0.60	0.90
Handwaving	0.74	0.73
Boxing	0.60	0.69
Running	0.55	0.87
Hand Clapping	0.57	0.61

Table 1. Comparison Results

V. Conclusions

In this paper, we have proposed a method for human action detection and identification as part of surveillance system. Six primitive actions have been classified on recognition of human action dataset. The prediction accuracies have been compared with the benchmark model. It has been observed that our model outperforms the benchmark model especially in walking, jogging, and running activities. A variety of improvement could be made to this system like shadow detection, difference between before and after sunset, capture with camera motion (from different angles) to construct 3D human models that would give more correct result.

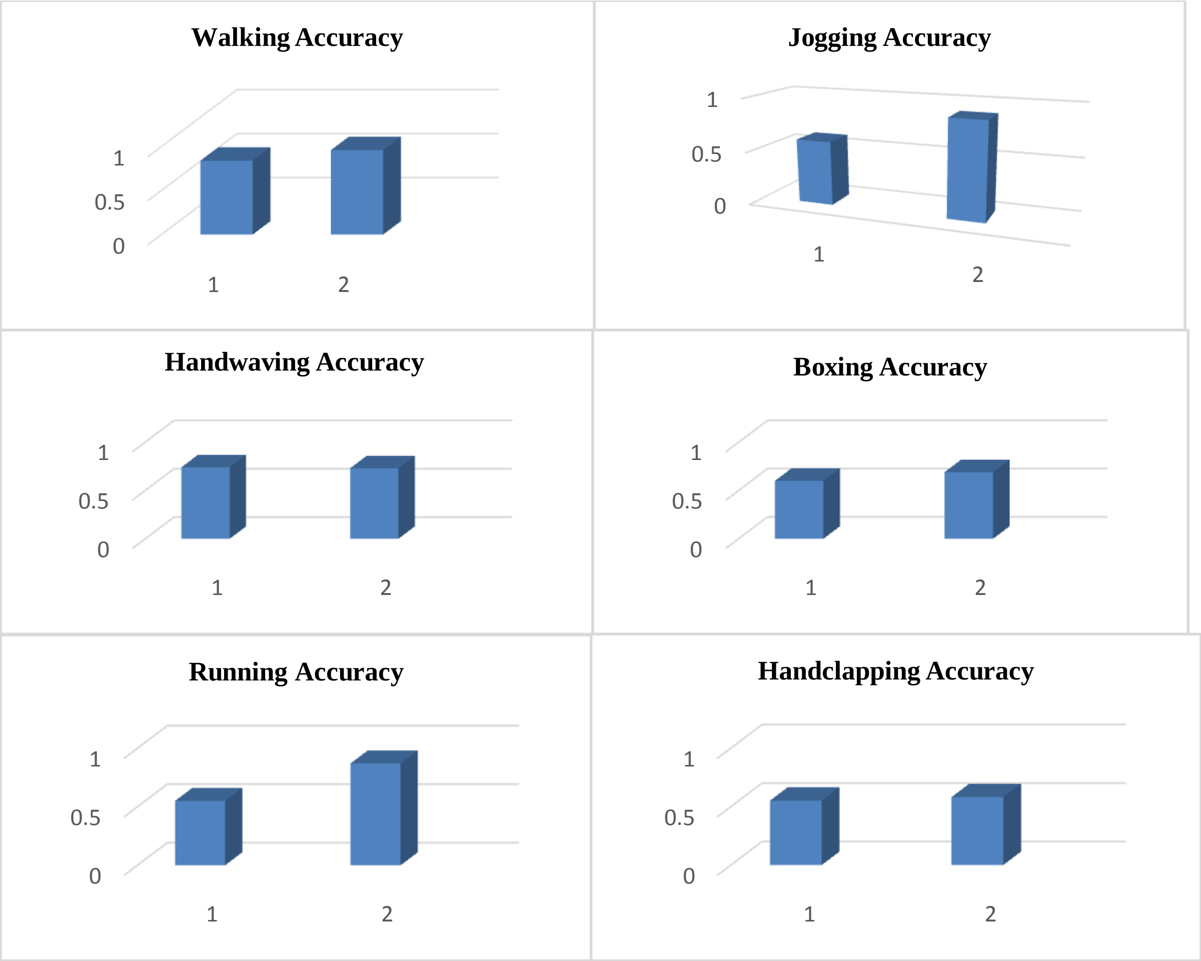


Figure 9. Comparison Results

REFERENCES

[1]Shabani, A. H., Clausi, D., and Zelek, J. S. “Improved spatio-temporal salient feature detection for action recognition,” in Proc. British Machine Vision Conference (Dundee), 1–12.

[2]Li, R., and Zickler, T. “Discriminative virtual views for cross-view action recognition,” in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Providence, RI), 2855–2862.

[3]Vrigkas, M., Karavasilis, V., Nikou, C., and Kakadiaris, I. A. “Action recognition by matching clustered trajectories of motion vectors,” in Proc. International Conference on Computer Vision Theory and Applications (Barcelona), 112–117.

[4]Lan, T., Wang, Y., and Mori, G. “Discriminative figure-centric models for joint action localization and recognition,” in Proc. IEEE International Conference on Computer Vision (Barcelona), 2003–2010.

[5]Morariu, V. I., and Davis, L. S. “Multi-agent event recognition in structured scenarios,” in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Colorado Springs, CO), 3289–3296.

[6]Chen, C. Y., and Grauman, K. “Efficient activity detection with max-subgraph search,” in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Providence, RI), 1274–1281.

[7]Sigal, L., Isard, M., Haussecker, H., and Black, M. J. (2012b). Loose-limbed people: estimating 3D human pose and motion using non-parametric belief propagation. Int. J. Comput. Vis. 98, 15–48. doi:10.1007/s11263-011-0493-4

[8] Tran, K. N., Kakadiaris, I. A., and Shah, S. K. (2012). Part-based motion descriptor image for human action recognition. Pattern Recognit. 45, 2562–2572. doi:10.1016/j.patcog.2011.12.028

[9] Wu, Q., Wang, Z., Deng, F., Chi, Z., and Feng, D. D. (2013). Realistic human action recognition with multimodal feature selection and fusion. IEEE Trans. Syst. Man Cybern. Syst. 43, 875–885. doi:10.1109/TSMCA.2012.2226575

[10] Martinez, H. P., Yannakakis, G. N., and Hallam, J. (2014). Don’t classify ratings of affect; rank them! IEEE Trans. Affective Comput. 5, 314–326. doi:10.1109/TAFFC.2014.2352268

[11]Song, Y., Morency, L. P., and Davis, R. (2012a). “Multimodal human behavior analysis: learning correlation and interaction across modalities,” in Proc. ACM International Conference on Multimodal Interaction (Santa Monica, CA), 27–30.

[12]Castellano, G., Villalba, S. D., and Camurri, A. (2007). “Recognising human emotions from body movement and gesture dynamics,” in Proc. Affective Computing and Intelligent Interaction, Lecture Notes in Computer Science, Vol. 4738 (Lisbon), 71–82.

[13]Christian Schuldt, Ivan Laptev and Barbara Caputo. "Recognizing Human Actions: A Local SVM Approach", in Proc. ICPR'04, Cambridge, UK.

[14]Laptev, I. and Lindeberg, T. Local descriptors for spatio-temporal recognition, in Proc. ECCV Workshop on Spatial Coherence for Visual Motion Analysis (SCVMA).

[15]Laptev, I. and Lindeberg, T. Velocity-adaptation of spatio-temporal receptive fields for direct recognition of activities: An experimental study, in Proc. ECCV Workshop on Statistical Methods in Video Processing, pp. 61–66.