Chapter 6

ACTIVITY RECOGNITION USING RGB AND SKELETON DATA INPUT

6.1 INTRODUCTION

Human Activity Recognition (HAR) is an active research area in the field of computer vision. Human activities have been observed as a series of basic movements. As an instance, activities like hand waving and brushing hair have been represented as a series of continuous lowering and raising of the hand. The HAR system recognizes these activities performed in videos and the images. The system has wide applications in the fields of robotics, security, industrial automation, and human-computer interaction. The key challenge of the HAR is to identify the class of actions robustly regardless of the variations

in external conditions and clothing of people performing the actions. Many of the previous approaches [228] mostly utilize only RGB videos for HAR. The spatio-temporal features have also been widely utilized for action recognition [229]. Recently, the availability of low-cost RGB-D sensors such as Microsoft Kinect [230] largely motivates the development of cost-effective solutions and simplify the problem of HAR. The skeleton images are ineffective in the lighting conditions and provide better body shape with a low error rate [229]. Furthermore, the availability of skeleton information provides the skeleton joints that act as 3D information to recognize the action. Skeleton data representation has another advantage of lower dimensionality which makes the code compact. Accessibility of highly recognized and diverse datasets like CAD-60[231], MSR Action 3D[232], SBU Kinect Interaction[230], UTD-MHAD[233], Berkeley MHAD[234] and NTU-RGB+D120 [235] has allowed for substantial growth in the research for HAR, using both RGB images and skeleton joints.

In recent years, due to the availability of valuable 3-dimensional structural information, most of the current approaches are principally based on Depth (RGB-D) and RGB and videos. Recently, many reserachers have addressed to use MHI and MEI from RGB videos for activity recognition [236] [237] [238], gesture and gait identification [239] [240]. It includes the prior information and also gradually gathers the most recent information. It holds the prior knowledge and also constantly collects the most current information. Therefore it can easily handle moving subjects, like human gait, action, moving cars, and gestures. Also, both the algorithm is simple to do motion analysis in a video. For

the MHI, the sequence of the silhouette is consolidated within grayscale images, meanwhile, it also preserves the powerful motion information. Hence, it can compactly express the motion sequence. This template is also much less sensitive to noises of silhouette, such as missing parts, holes, and shadows. MHIs are the best easiest way to represent a video in a scalar-valued image where the pixel that has moved recently is the brightest, with pixels with lower intensity indicating a past motion. MHIs stores the history of temporal changes at each pixel location which decays over time. MEI is a slight modification of MHI, with MEI encoding the whole movement into a single binary image characterising only the motion regions with no information regarding the direction of motion. Motivated by above discussion, to detect the existence and direction of motion in a frame, we have exploited these template matching approaches (MHI, MEI) for activity recognition.

In recent years, deep learning-based approaches have been extensively used for image classification and recognition purposes [218]. Among the various deep learning architectures, convnet[218] is most widely used for feature extraction for images and has replaced hand-crafted features. For solving the HAR problem many of the techniques were only based on a convnet based program [241]. Another class of deep learning architecture includes RNN [242] where connections between nodes form a directed graph along a temporal sequence allowing it to use a dynamic behavior. LSTM[17] is an excellent RNN algorithm in processing the sequence of inputs and is used for tasks like speech recognition.

In this paper, we aim to take full advantage of the data available from the RGB video and the 3D skeleton joints movement. These data have been employed with the power of

convnet supported by the ability of LSTM to process the sequence of inputs. The given method builds an end to end learning framework for an accurate HAR. The introduced architecture consists of two main streams: (i) the first stream utilizes the RGB video frames for the formation of MHI and MEI images. The convnet has been trained using these images. (ii) the second stream uses skeleton data. To handle the skeleton data, we have proposed an algorithm that develops a skeleton intensity image, for three views (top, front and side). These views create the three branches, where every individual branch uses the convnet, along with intermediate supervision for each branch. On top of convnet sub-network, a recurrent sub-network named LSTM takes the feature maps from the convolutional layers as input to exploit the temporal dependency. The individual stream has been trained separately giving multiple models with their softmax scores being fused later at decision level using WPM. The performance of our system has been analyzed according to two constraints: the first by using the LSTM module to make temporal refinement which leads to give better accuracy as compared to the other state-of-the-art techniques and the second by using a cyclic learning rate approach without the LSTM module that decreases accuracy but makes the learning faster.

The motivation behind using the cyclic learning rate is that the deep neural networks usually do not converge to the global minimums, there is a notion of bad or good minima in terms of generalization. It should be intuitive that flat basins are better as compared to the sharp minimal because slight changes in the weights may change the model predictions dramatically. It is very clear that as the number of parameters increases the number of local minima increases exponentially. So large learning rates help to escape such sharp

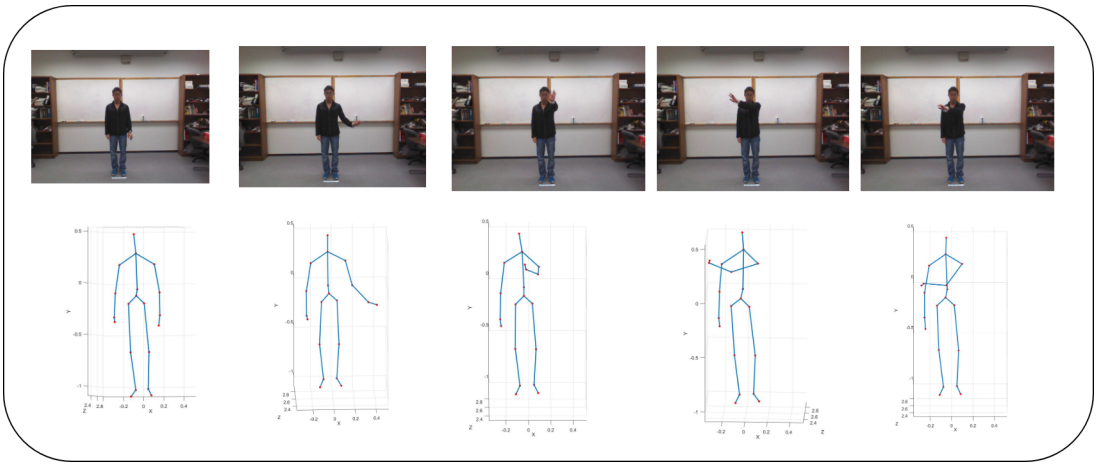


FIGURE 6.1: Few RGB and skeleton sequences for the activity 'Right arm swipe to the left' from UTD-MHAD dataset.

minimums from time to time.

Experimental results on three benchmark action datasets UTD-MHAD, CAD-60 and NTU-RGB+D120 show that the proposed learning framework achieves state-of-the-art accuracy for the same experimental conditions. To the best of our knowledge, the proposed method is the first in the literature which combines the knowledge of convnet with the LSTM in a unique style of multi-modal framework where we need to train the model only once by cyclically changing the learning rate.

6.2 Related Work

The variety of activity recognition approaches in literature were largely utilized the RGB data as compared to skeleton data. In this section, we review the methods that use RGB and skeleton information.

After the use of convnet for classification [218], deep learning techniques have been largely utilized. Multiple convnet architectures had been used for RGB frames for activity recognition in [243]. A 2-stream convnet technique in [241], was utilized for action recognition incorporated with spatial and temporal data, multi-frame optical flow with convnet. In [236], deep learning technique was utilized for a multi-model approach using RGB, depth and skeleton data as their input.

In [17], the authors proposed an action recognition technique using both the convnet and LSTM deep architecture. Where convnet was utilized for simple feature extraction and LSTM for sequential information. Similarly [228], introduced a technique for video representation to recognize the action using convnet with LTC(Long-term Temporal Convolution), which increased the temporal extent to make the system accurate.

Skeleton information has been widely utilized for human pose and activity recognition. A fusion technique based on the skeleton, RGB and depth data was performed in [244] for action recognition. In [245], skeleton based information along with intensity and depth data were utilized for gesture recognition. In [246], multiple machine learning procedures were used for skeleton body joint detection, used to detect the pose involved in the performed activity. Then pose based spatio-temporal evaluation exploited to classify the activity.

The proposed methodology is motivated by the work done using RGB and skeleton sequences using convnet based approaches. The main aim of the proposed architecture is to take advantage of both RGB frames and skeleton data using two-stream architecture as shown in Fig. 6.2.

6.3 Methodology

An overview of the introduced framework has been illustrated in Fig. 6.2. The proposed method is a two-stream deep network. The main aim of the proposed architecture is to take advantage of both RGB frames and skeleton data. In the first stream of the network, the RGB frames have been used to make the powerful MHI and MEI. After that convnet has been trained with MHI and MEI using feature-level fusion. In the second stream, the skeleton data has been used with the proposed algorithm to develop the skeleton intensity image, for three views (top, front and side). These views make the three branches using convnet, along with intermediate supervision for each branch. Each view is first analyzed by a convnet, generating a set of feature maps which are fused for further analysis. On top of convnet sub-network, a recurrent sub-network called LSTM takes the feature maps from the convnet layers as input to exploit the temporal dependency. Both the streams have been trained independently and then the softmax scores have been fused at the decision level with the help of WPM. Fig. 6.1 shows the few RGB and skeleton data for activity 'right arm swipe to the left' from UTD-MHAD dataset. Fig. 6.2 describes the overall pipeline of our network involving the 2-streams fusion of MHI, MEI and skeleton images.

6.3.1 RGB frames

The first stream of the network deals with the RGB data. The RGB frames have been used to make the MHI and MEI. The overall description of this stream and both the image

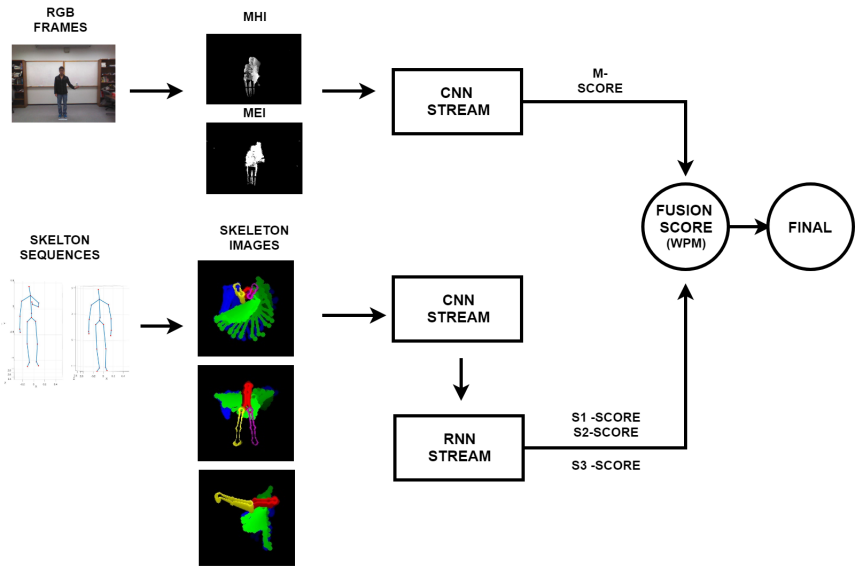


FIGURE 6.2: Proposed framework.

representations are described in the below sections:

6.3.1.1 Motion History Image (MHI) and Motion Energy Image (MEI)

MHI is one of the simplest and easiest ways to represent a video in a scalar-valued image where the pixel that has moved recently is the brightest (pixel with lower intensity indicates a past motion) and vice-versa. The idea of MHI came with [5]. The advantage of this solution is that it gives temporal information of the video in a single image. MHI's expresses motion flow using the temporal density of pixel. The intensity value of zero at a particular pixel tells that there is no recorded motion at that position in the given history of frames. The $M_\tau(m, n, t)$ is the representation of MHI which can be computed utilizing

the update function ψ(m,n,t) shown in Equation 1.

$$M_\tau(m,n,t) = \begin{cases} \tau & \text{if } \psi(m,n,t) = 1 \\ max(0, M_\tau(m,n,t-1) - \delta) & \text{otherwise} \end{cases} \tag{6.1}$$

Here ψ(m,n,t) indicates the presence of motion in the recent frame of video. Variables (m,n), δ, t indicate pixel location, decay parameter and time respectively. τ determines the temporal duration of the MHI. This update function is used over every frame of the video. The value of δ has to be determined empirically depending on the sets of actions to be classified. In our framework, we have kept the value of the δ at 23. The update function can be obtained from the frame subtraction using the threshold :

$$\psi(m,n,t) = \begin{cases} 1 & \text{if } D(m,n,t) \geq \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

where D(m,n,t) is the frame difference.

$$D(m,n,t) = |I(m,n,t) - I(m,n,t \pm \Delta))| \tag{6.3}$$

Here I(m,n,t) interprets the value of the pixel intensity of a frame at the particular timestamp. For the experiment, we have kept the value of Δ as 1 to consider consecutive frames.

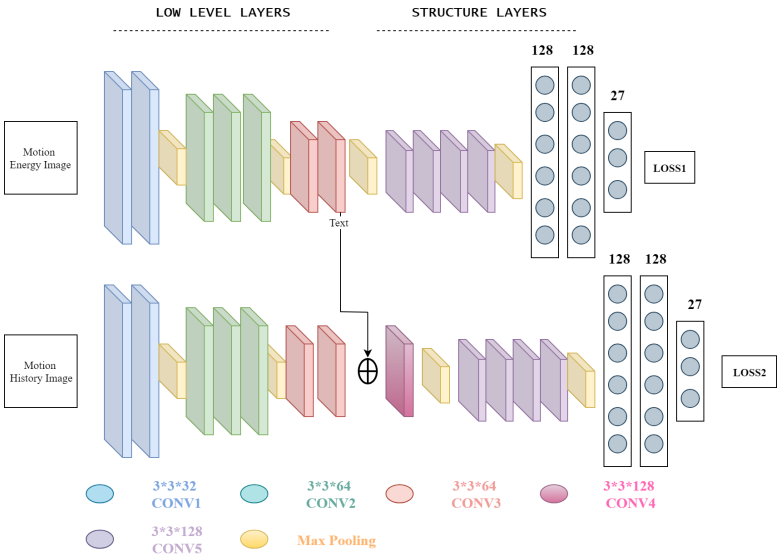


FIGURE 6.3: Architecture of the multi-input convnet. Here the features extracted from the MEI are combined with the features extracted from the MHI. This network has been inspired by the VGG-16 model but in order to prevent overfitting, the number of elements of the layers has been reduced.

The MEI [238], on the other hand, encodes the whole movement into a single binary image characterizing only the motion regions with no information regarding the direction of motion.

6.3.1.2 First Stream

This network takes MHI and MEI as the input of size 240X320. Fig. 6.3 depicts the pipeline of the introduced architecture for the 1^{st} stream. The whole network has been divided into two branches where each branch takes separate input. The top branch takes MEI and the lower branch takes MHI. The intermediate supervision as shown in Fig. 6.3 with an arrow has been used which replenishes the gradients for further layers. The network consists of a total of 29 layers, 23 convolutions, and 6 fully connected layers.

After each convolution layer, we use ReLU as the non-linear function. We use the max-pooling layer with a kernel size of 2 and strides 2. Before each max-pooling layer, we add the dropout layer with a ratio of 0.6 for generalization to improve the network performance (prevent overfitting).

Each branch of the network consists of two parts, the first is low-level layers for simple feature extraction and the second is structure layers for learning the structure relations. Low-level layers are general layers in convnet which are used to extract low-level features from the RGB images. The features obtained from the MEI after the lower level layers are combined with the features from MHI. By using the intermediate supervision we can efficiently combine the features extracted from MEI with MHI and continue the learning to make the final prediction. Our motivation to perform this fusion is that MHI starts to perform poorly with a higher value of decay parameter. So, MEI which coarsely describes the spatial distribution for a given view of the action can provide intermediate supervision as it can store larger memory before the results start to degrade. MEI supervision also allows the network to focus on a smaller subset of actions. Each of the two branches has been trained individually and later the weights of the pre-trained model are used in the current network to improve the performance of given system. The Mean Square Error (MSE) loss function has been used here. For the whole network the loss function is:

$$loss = loss2 + \lambda loss1 \tag{6.4}$$

$$loss1 = \frac{1}{U}\sum_{u=1}^{U}\sum_{v=1}^{V}(Y_{uv} - \bar{Y_{uv}})^2 \tag{6.5}$$

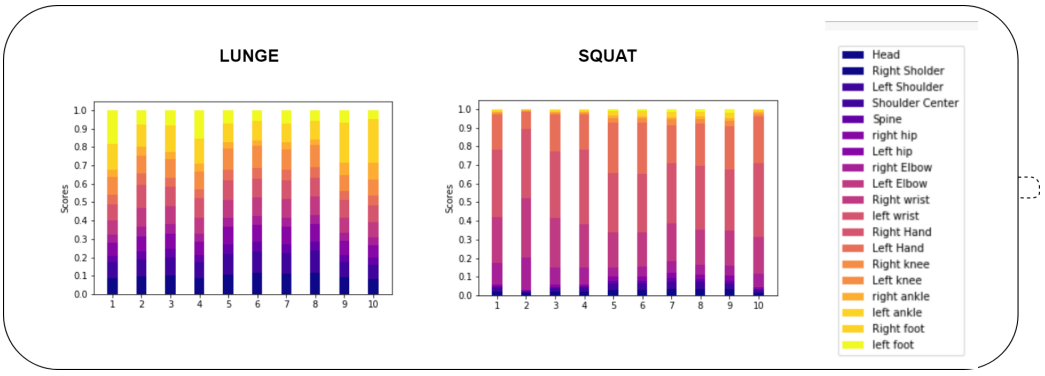


FIGURE 6.4: Stacked histogram of joint contribution for 2 actions in UTD-MHAD dataset: 1. Forward Lunge 2. SQUAT. Each bar indicates the contribution ratio of each joint to the action in one instance.

$$loss2 = \frac{1}{U}\sum_{u=1}^{U}\sum_{v=1}^{V}(Y_{uv} - Y_{uv}{}')^2 \tag{6.6}$$

Here in equation 4, λ is the balancing factor for the two-loss functions (loss2 and loss1 function for individual streams corresponding to MHI and MEI). We added the MEI part only as a guide to the network trained on MHI, but the emphasis is made on the prediction made out of MHI. For the experiment, we kept the value λ as 0.2.

In equation 5 and 6, Y_{uv} is the truth value, $\bar{Y_{uv}}$ is the value obtained corresponding to stream having MEI as input, U and V are the number of training examples and classes, $Y_{uv}{}'$ is the value obtained corresponding to the stream having MHI as input.

6.3.2 Skeleton Joints

The second stream of the network deals with the skeleton data. The skeleton joint sequences have been used to make the skeleton images with the given algorithm in section

3.2.1. The convnet and LSTM have been utilized to train the system with generated skeleton images. The whole description of the skeleton image generation and second stream architecture has been discussed below.

6.3.2.1 From Skeleton sequences directed to the Images

The particular activity with the skeletal information composed of k joints and n number of frames. The number of frames per activity is going to differ from activity to activity and as well as the person performing the activity.

Each joint is represented by J_i, a three-dimensional vector. Here i represents the joints for an activity. Since the person can be found at any place in the coverage area of the sensor it is necessary to center the coordinate space with respect to any one of the joint. Here we are using the hip Center joint of the first frame for normalization purpose.

The given expressions have been used to normalize the joint coordinate:

$$D_i(P) = J_i(P) - J_o(1) \tag{6.7}$$

$$D_{max} = max(D_i(P)) \quad \forall i \varepsilon joints \quad p \varepsilon frames \tag{6.8}$$

$$D_{min} = min(D_i(P)) \quad \forall i \varepsilon joints, \quad p \varepsilon frames \tag{6.9}$$

$$d_i(p) = (D_i(P) - D_{min})/(D_{max} - D_{min}) \tag{6.10}$$

Let $J_i(p)$ denote the coordinate of the i^{th} joint in the p^{th} frame. $D_i(p)$ is the distance between the vectors $J_i(p)$ and $J_0(1)$. $d_i(p)$ is the distance between the vectors $J_i(p)$

and $J_0(1)$ normalized such that each joint feature is within the range of 0 and 1. $J_0(1)$ represents the coordinates of the Hip Center in the first frame.

To make the skeleton images from skeleton sequences we are dividing the skeleton into five parts named trunk, left and right arm, left and right leg. We prepare three inputs of size 160×160×5 from the skeleton sequences, one corresponding to each view(top, side, bottom). Each input comprises of 5 independent images of size 160×160 corresponding to each body part, which is stacked together. Each image is an independent MHI of that body part for the given view. The intuition behind this was to keep the MHI's of every body part independent and prevent it from overlapping of the body parts and loss of information. The motivation behind keeping three view inputs for the 2D convnet instead of using a single 3D input is that it reduces the amount of computation and memory usage without losing the extra depth information available with us. Each skeleton joint information is converted to an image corresponding to a view. Let $I(x, y, v_i)_{pq}$ be the template image of the size of (x,y) corresponding to the given v_i^{th} view and p^{th} body part and the q^{th} frame.

p_x and p_y indicate the pixel locations of the image.

$$P_x(top)_{pq} = c_1 + k_1 + d_i(q)_x \tag{6.11}$$

$$P_x(side)_{pq} = c_3 + k_3 + d_i(q)_y \tag{6.12}$$

$$P_x(front)_{pq} = c_1 + k_5 + d_i(q)_x \tag{6.13}$$

$$P_y(top)_{pq} = c_2 + k_2 + d_i(q)_z \tag{6.14}$$

$$P_y(side)_{pq} = c_2 + k_4 + d_i(q)_z \tag{6.15}$$

$$P_y(front)_{pq} = c_4 + k_6 + d_i(q)_y \tag{6.16}$$

The value of c_1, c_2, c_3, c_4, c_5, c_6, k_1, k_2, k_3, k_4, k_5, k_6 are constants. The value of c_1, c_2, c_3, c_4, c_5, c_6 are values used to center align the skeleton image. The value of k_1, k_2, k_3, k_4, k_5, k_6 determines the spacing between the joints and depends on the stretch of the given joint feature in a given dimension for a given frame. The values may change depending on the joints selected for normalization.

All the joints for the p^{th} body part are joined according to human structure information, using the concept of informative body parts which are reflected in the generated image through the relative thickness of parts. Instinctively, human tends to pay more attention to the moving objects and pay less attention to static ones. Motivated by this tendency we aim to find informative body parts. We have calculated the distance moved by each joint over the course of action and calculated the softmax score corresponding to all the joint to get contribution produced by each joint in each action. The information related to the body parts is computed by averaging over the softmax score of each joint associated with the body part, which defines the thickness of limbs involved with a part p. Fig. 6.4 illustrates the detailed contribution of 10 instances of 2 different actions of the UTD-MHAD dataset. For instances in the same action class the contribution bar follows a similar pattern.

$$s(i) = e^{Y^i} / \sum (e^{Y^i}) \tag{6.17}$$

Algorithm 5: Skeleton joints → Skeleton images

Result: Skeleton Image

Input: *Skeleton Joints (SJ)*: a 3D array consisting of the coordinates of the joints for every frame (here SJ(i,j) represents all the 3 coordinates of i^{th} joint in the j^{th} frame).
Informative Joints Score (IJS): a 1-D array, indicating the score corresponding to each joint.
Distance Joint (DJ): a 2-D array indicating the sum of displacement for each joint between consecutive frames.
Number of Frames (NF): indicating the number of frames in the given skeleton joints sequence.
Parts (P): The body parts of human (the leg, right and left arm, and trunk).
IXY, IYZ, IXZ: Images corresponding to the skeleton joints for each pair (a 3-D array).
IXY(k, a, b): represents the pixel intensity at the k^{th} frame, a^{th} row, and b^{th} column.
Thickness Score (T): a 1-D array consisting of the thickness corresponding to each particular joint.
Skeleton Images (SI): for each pair, i.e., SIXY, SIYZ, SIXZ.
Skeleton Images Temporary (SIT) is used to store the temporaray skeleton images.
i=1,2,....,k (represents all the joints for a activity)
p= 1,2.. P (body parts)
Part 1: *Calculation of Informative Joints Score for each joint*;

```
j=0
DJ is initialized to zero for all the joints
while j!=NF do
    for m = i do
        DJ(m) = DJ(m) + |SJ(m, j) − SJ(m, j − 1)|
        j=j+1
    end
end
```

$$Y^i = \sum_p |J(i)(P) - J(i)(P-1)| \tag{6.18}$$

where P=2,3,....,n(for all frames in activity)

Here S(i) is the softmax score corresponding to each joint and Y^i is the distance moved by a particular joint over all the frames involved in the activity. Here i=1,2,...,k for all joints in the activity. Fig. 6.5 shows the sample images of the MHI, MEI, skeleton image (for visualization purposes we have used an image of 160×160×3 with each body part is

for $m = i$ **do**
    IJS(m)=DJ(m)/($\sum(DJ(i))$) (; // Here $\sum(DJ(i))$ represents the summation of DJ over all joints
)
end
Part 2: *Calculating the Thickness Score for each part*
N=0;
for $j \leftarrow 1$ *to* p **do**
    while $k \leq j$ **do**
        T(j)=T(j)+IJS(k)
        N=N+T(j)
    end
end
for $j \leftarrow 1$ ***to*** p **do**
    T(j) = T(j) / N
end
Part 3 and Part 4 are done for each (x,y), (y,z) and (x,z) pair. We have shown an example for (x,y) pair.
Part 3:
while *j!=NF* **do**
    for $k \leftarrow 1$ ***to*** i **do**
        $I_{xy}(1) = c_1 + k_1 \times SJ(k, j, 1)$
        $I_{xy}(2) = c_2 + k_2 \times SJ(k, j, 2)$;
    end
end
All the joints in IXY are joined according to the human structure information and the thickness of each limb is determined by array T, where T(i) scores determine the relative thickness of the corresponding body parts. Here c1, k1, c2, and k2 are experimentally determined constants.
Part 4- *Creating Skeleton Images from results obtained from Part3*
while *j!=NF* **do**
    $$SI_{xy} = \begin{cases} SIT(x, y, j) = \tau \textbf{ if } Q(x, y, j) = 1) \\ max(0, SIT_{xy}(x, y, j - 1)) \textbf{ else} \end{cases}$$
end
Here Q(x,y,t) is basically an indication of motion in the current frame. It is calculated with the help of Equation 2 and 3. Basically, SIXY is MHI developed from IXY.
SIXY = SITXY(x,y,j-1)

shown using a different color) of activity dataset of UTD-MHAD: 1. Draw X, 2. Boxing and 3. Tennis Swing. In the skeleton sequences, the intensity change shows the body part movement from the past to the current frame to capture the temporal information just like MHI. The clear visualization is that the given method to generate the skeleton sequences from the joints is a basic but efficient technique. It allows keeping both the structural as well as temporal data for the joints locations and dividing the human structure into five modules to provide more auxiliary information about the spatial data of the human structure and its movements.

6.3.2.2 Second Stream

The second stream network has 3 branches with 3 views. The size of input for every branch is 160×160×5 as shown in Fig. 6.7. The input image representing a particular view passes through 12 convolution layers (strides are 1) and 2 fully connected layers with ReLU as the activation function, which extracts spatial features. We use the max-pooling layer with a kernel size of 2 and a stride of 2. The 3×3 convolution filters have been used throughout the network. At last, posterior probability has been given by the softmax layers applied for supervision. Two dropout layers are added and given the ratio with 0.6 among the fully connected layers to avoid overfitting and the batch normalization layer is added in between convolution and activation layers. For the top view, the number of elements in the hidden layers was reduced (reduced by half for each convolution layer) to prevent overfitting.

The intermediate supervision has been utilised for all the views (loss1, loss2, loss3) to

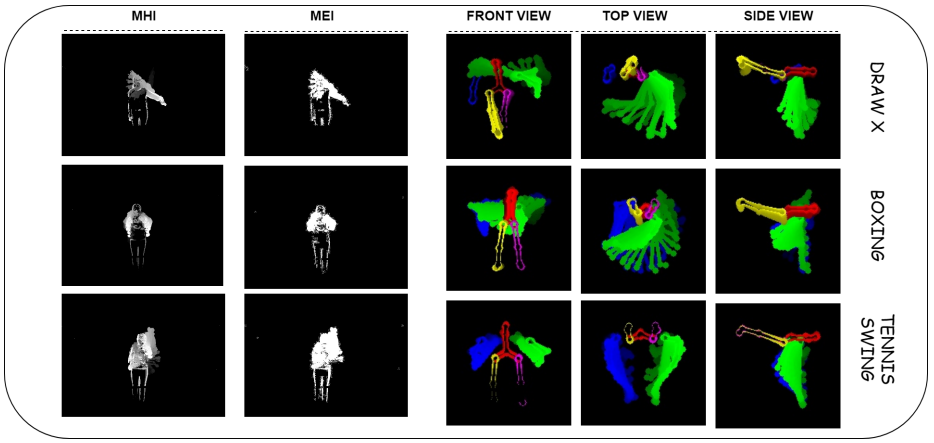


FIGURE 6.5: Sample images of MHI, MEI, Skeleton Images(Front View, Side View, and Top View) created RGB and skeletal data of UTD-MHAD dataset for activities 1. Draw X, 2. Boxing, 3. Tennis Swing

handle the problem of vanishing gradient. Every branch has been trained independently and the resultant weights are used for the initialization of whole network training. With this kind of transfer learning, we try to exploit the prominent features from each of the branches independently to improve the system performance. This kind of transfer learning also reduces the overall training time and ensures that all three views contribute independently towards final prediction. Once the features from each branch have been concatenated represented by concatenation symbol in Fig. 6.7. The combined features are then passed through 4 convolution layers and 3 fully connected layers, the last layer producing the probability scores for final prediction.

We have used MSE as our loss function. For the whole network, the overall loss function is shown in Equation 19. The value of λ is kept at 0.5. Loss1, loss2, loss3 are used to learn the spatial information from each view (Top, Front and Side View) but final prediction (loss) should be made after combining the information from each branch for better

performance.

$$loss = loss0 + \lambda(loss1 + loss2 + loss3) \tag{6.19}$$

$$loss0 = \frac{1}{P}\sum_{o=1}^{O}\sum_{p=1}^{P}(Y_{op} - \bar{Y_{op}})^2 \tag{6.20}$$

$$loss1 = \frac{1}{P}\sum_{o=1}^{O}\sum_{p=1}^{P}(Y_{op} - K_{op})^2 \tag{6.21}$$

$$loss2 = \frac{1}{P}\sum_{o=1}^{O}\sum_{p=1}^{P}(Y_{op} - R_{op})^2 \tag{6.22}$$

$$loss3 = \frac{1}{P}\sum_{o=1}^{O}\sum_{p=1}^{P}(Y_{op} - B_{op})^2 \tag{6.23}$$

In equation 20, 21, 22 and 23, Y_{op} is the truth value, $\bar{Y_{op}}$ is the value obtained after combining the features from all the views, K_{op} is the value obtained corresponding to the stream having Top view as input, R_{op} is the value obtained corresponding to the stream having Front view as input, B_{op} is the value obtained corresponding to the stream having Side view as input, and O and P are the number of training examples and classes.

The convnet is followed by an LSTM network. LSTM is a class of neural networks that maintains internal hidden states to model the dynamic temporal behavior of sequences with arbitrary lengths through directed cyclic connections between its units. LSTM extends RNN by adding three gates to an RNN neuron: a forget gate f to control whether to forget the current state; an input gate i to indicate if it should read the input; an output gate o to control whether to output the state. These gates enable LSTM to learn long-term dependency in a sequence. The motivation behind using LSTM along with convnet comes from one disadvantage present with MHI. In case of MHI if some movement happens in a

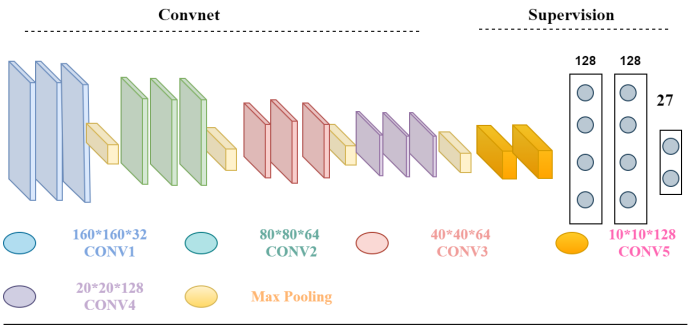


FIGURE 6.6: Architecture of the convnet model with supervision layers used for testing/training of a single view of the skeleton image.

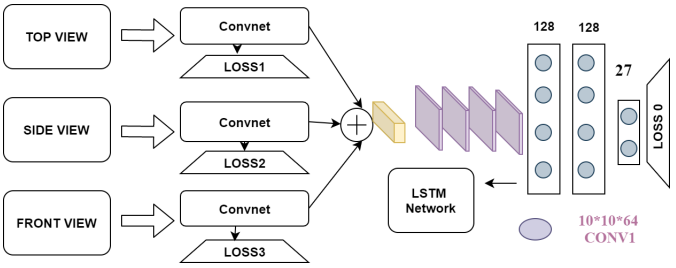


FIGURE 6.7: The overall network with the fusion of 3-convnet streams of skeleton image. Intermediate supervision is present to avoid overfitting.

frame k, then ψ(m,n,t) =1 else ψ(m,n,t) =0. According to Equation 1, it is clear that once the value of ψ(m,n,t) becomes 1 at a given pixel it sets the corresponding pixel value in the MHI equal to τ. Thus MHI is capable of holding only the latest pixel value and with actions comprising of a larger number of frames, MHI starts losing part of the information.

The skeleton images have been used to extract features using trained convnet network. Then these features are passed to the single-layer LSTM network.

Instead of generating one single skeleton image for a single view for the entire activity process, we aim to generate 1 image per J frames and run through convnet network and saving them before the final output layer. Those extracted features have been concatenated to generate a sequence that is utilised for training the LSTM. For this LSTM, we have

used a single 256-wide LSTM layer, followed by a dense layer of size 128, followed by a softmax layer. In between the dense layers, the dropout layer has been used along with a ratio set of 0.6. This final step gave a boost to our result and allowed us to achieve a state-of-the-art result on the UTD-MHAD dataset. This technique of combining the power convnet with LSTM allows us to recognize relatively longer actions in which case the intensity images start to lose information.

6.3.3 Cyclic Learning Rate

Inspired by Huang and Yixuan Li [7], we have proposed a technique to boost the performance of our multi-model network without having to spend extra training time. It is important to understand that deep neural networks do not converge to global minima. It is a known fact that the number of local minima grows exponentially with the number of parameters. Our deep network would consist of millions of them. It has been shown that though most of them are having the same error rates the model corresponding to a particular local minimum would make different mistakes. This technique can be exploited through ensembling whereby training multiple neural networks with different initialization and with each converging to a different solution, averaging them over they achieved drastically reduced error rates. Instead of using different initialization we have used the above property by changing the learning rate in a cyclic fashion whereby changing the learning rate to a very high value will enable the neural network to escape the current minima and start converging to another one. This method enabled us to improve error

rates and improve the prediction accuracy over the convnet part. This method gave us 3 models by changing the learning rate from 0.01 to 0.001 over 50 epochs for 3 cycles each.

6.3.4 Decision Level Fusion(DLF) approach

The DLF procedure has been used to integrate the output of each of the individual streams. DLF involves processing the classification results of the individual classifiers to take advantage of the independent classifiers to achieve higher robustness by combining their results. There are two major techniques involved in DLF, one involving additional training (Supervised) and others requiring no training (unsupervised). The supervised technique involves providing the output of individual classifiers (softmax scores) as input to the training model. In this paper, we have discussed an unsupervised method for our experiment, which is WPM.

6.3.4.1 Weighted Product Model (WPM)

The WPM [247] is a popular multi-criteria decision making (MCDM) method [248]. The problem of aggregating the softmax scores given by individual classifiers is taken as an MCDM problem. It is similar to the weighted sum model (WSM) [249]. The main difference is that instead of adding in the computation, multiplication has used.

Suppose the MCDM problem is defined on n decision specification and m alternatives. Here w_j refer the relative weight for the value of specification j and a k_j is the representation of performance for the value of alternative A_k. Which is evaluated in the form of the

specification j. The WPM can be calculated as follows.

$$P(A_k) = \prod_{j=1}^{n} (a_{k_j})^{w_j}, \quad for \quad k = 1, 2, 3, ..., m. \tag{6.24}$$

Let MHI_{PS} and SI_{PS1}, SI_{PS2}, SI_{PS3} are the scores obtained from the individual streams of (MHI and MEI), and multiple models from skeleton images respectively. These scores act as the criterion whereas the number of classes is the alternatives. Every score in the discrete stream is a matrix-vector of the same size as the number of classes. According to these decision specifications, the prominent alternative class (with the highest P(A_k)) gets picked. The values of W_1,W_2,W_3, and W_4 was kept 1 for every individual dataset. As long as we have the values of the same weights then the WPM normally acts as a 'Product Rule'.

$$WPM_S = max(MHI_{PS})^{W_1} \times (SI_{PS_1})^{W_2} \times (SI_{PS_2})^{W_3} \times (SI_{PS_3})^{W_4} \tag{6.25}$$

6.4 Experiments

In this section, we discuss the details of various datasets and evaluation metric used and the quantitative analysis of the proposed work. We discuss the effectiveness of the proposed method on three publicly available datasets UTD-MHAD, CAD-60 and NTU-RGB+D120. The evaluation parameter and criteria have been discussed to make a fair

comparison with the results of the already existed techniques. The proposed method is compared with the various state-of-the-art techniques based on their accuracy, precision, and recall.

$$Accuracy = TP + TN/TP + FP + FN + TN \tag{6.26}$$

$$Precision = TP/TP + FP \tag{6.27}$$

$$Recall = TP/TP + FN \tag{6.28}$$

In the above expressions, TP stands for true positive, TN is a true negative, FN is false negative and FP is a false positive. The proposed framework of the multi-modal system outperforms most of the state-of-the-art methods. We used 'Adam' optimizer with a learning rate set at 0.001 and batch size was kept at 16. We have been implemented our system in python with TensorFlow as our backend and training have been done on Tesla K80 GPU with 12 GB RAM.

6.4.1 UTD-MHAD Dataset

UTD-MHAD is one of the most complicated multi-modal action datasets. It consists of 27 actions, being performed by 8 subjects(4 males and 4 females). Every single actor performs the specific action 4 times to give a total of 861 sequences. This dataset provides four temporally synchronized data modalities; RGB videos, depth videos, skeleton positions, and inertial signals from Kinect camera and a wearable inertial sensor. In the

proposed approach, we required only the RGB videos and skeleton positions. The ‘Draw Circle CW’ and ‘Draw Circle ACW’ are samples of similar activities that make the dataset ambiguous. We have used a 60-40 rule for the train-test split.

The comparative analysis of the proposed work with the other state-of-the-art techniques has been shown in Table 1. We have also evaluated our results on all the combinations of individual streams and compared how individual streams perform as compared to the overall multi-modal system as shown in Table 2. Results have shown that combining individual streams boosts our performances, thus enabling the multi-modal system in achieving better accuracy as compared to the other reported results. We conducted 5 tests on our network each time using the 60-40 split ratio. Over the 5 tests, we achieved an average accuracy of 95.4 percent with the highest being 96.6 percent and lowest 94.6 percent. The class-specific accuracy for 340 test samples is given in the Confusion Matrix Table 5. Using the WPM technique, the accuracy increased by 3-4 percent in all random trials. It is quite evident that from the confusion matrix that our network is successful in recognizing similar activities like ‘Draw Circle CW’ and ‘Draw Circle ACW’.

6.4.2 CAD-60

The CAD-60 dataset consists of 12 challenging activities. Each activity is conducted by 4 subjects including 2 male and 2 female of which one is left-handed, that makes the dataset complex. These activities were conducted in five distinct settings which are office, kitchen, bedroom, bathroom, and living room. The dataset has been handled based on

the considered settings for the experiments. However, the proposed approach has been evaluated by combining all the settings to increase the complexity level. For performance testing, the "new person setting" is followed as given in [9]. To make the model further robust, the mirrored copies of each action have also been added to the dataset. Table 3 shows the comparison of the proposed approach with the recent state-of-the-art results. Table 6 shows the confusion matrix on this dataset for all 12 activities. Irrespective of the complexity of the dataset the proposed approach has achieved satisfactory results. This has happened mainly because of the utilization of the multi-modal data cues which lowers the error rate and the ability of convnet and LSTM that is invariant against the orientation of the image and makes the recognition robust irrespective of person is left or right-handed.

6.4.3 NTU-RGB+D120 Dataset

This is presently the biggest in-door obtained dataset. This is an expansion of the NTU-RGB+D dataset including 120 activity sets and furthermore added 114,000 video specimens. The lately added activity sets make activity recognition jobs more challenging. Because the distinct activities may have alike body movements but diverse subjects. They may have fine-grained finger motions and hand and so many. This dataset contains 32 setup ID and 106 subjects. Cross-setup and cross-subject benchmarks are set. For cross-setup, all the setup IDs have arranged in two separate train and test parts. Analogously, for cross-subject, fifty-three subjects organize to act as the train set and the rest fifty-three subjects utilized as the test set. Analogously, the 32 setup IDs are also divided equally

TABLE 6.1: UTD-MHAD dataset evaluation for activity recognition techniques

Sr.No.	Method	Accuracy
1	DMM-CRC(Depth+Inertia)[233]	79.9
2	GF+LF(RGB+Skeleton+Depth)[245]	84.4
3	SDSR(Skeleton)[251]	86.12
4	DMM-CTHOG-LBP-EOH(Depth)[252]	88.4
5	Ours(Skeleton(convnet only))	92.94
6	TPM-LLC-BoA(Skeleton)[253]	93.02
7	RGB+3-Depth+Skeleton [236]	95.11
8	Ours(Skeleton(convnet)+RGB)	95.16
9	Ours(Skeleton(convnet+LSTM))+RGB	96.5

TABLE 6.2: Performance of individual streams and complete system on UTD-MHAD

Sr.No.	Method	Accuracy
1	MHI(RGB)	84.7
2	MHI+MEI(RGB)(Decision Level)	85.29
3	MHI+MEI(RGB)(Feature Level)	89.11
4	Skeleton(Top View)	84.7
5	Skeleton(Side View)	85
6	Skeleton(Front View)	87.94
7	Skeleton(Front+Side+Top)(convnet)	92.94
8	Skeleton(convnet)+RGB	95.16
9	Skeleton(convnet+LSTM)+RGB	96.5

into two parts for training and testing in cross-setup.

For this dataset, the outcomes on cross-setup and cross-subject settings of the proposed method and current state-of-the-art techniques are listed in Table 4. The introduced method obtains 77.9% on cross-setup and 76.7% on cross subject and it beats the technique Caetano et al. [250] which gives very much close result to ours method by 11.0% and 9.2% on the cross-setup and cross-subject settings respectively.

TABLE 6.3: Activity Recognition methods on CAD-60

Sr.No.	Method	Precision	Recall
1	Gaglio et al.[246]	77.3	84.7
2	Zhu et al.[254]	93.2	84.6
3	Parisi et al.[255]	91.9	90.2
4	Cippitelli et al.[256]	93.9	91.9
5	Khaire et al.[236]	93.0	90.0
6	Ours(Skeleton(convnet+LSTM))+RGB	93.2	91.9

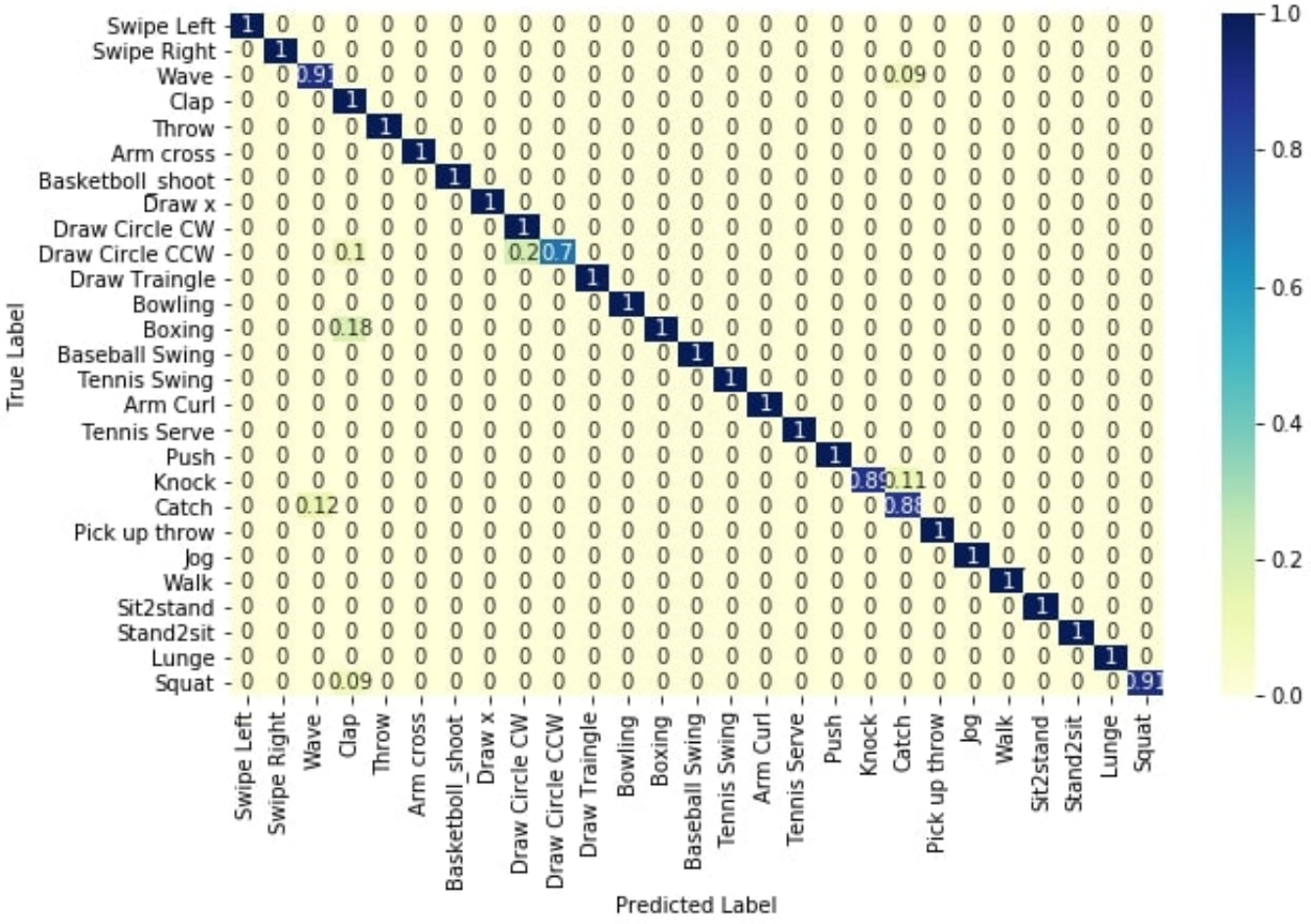
TABLE 6.4: NTU-RGB+D120 dataset evaluation for activity recognition techniques

	Technique	Cross-subj. Acc(%)	Cross-setup Acc(%)
Literature Results	Liu et al. [257] (given in [258])	58.2	60.9
	Caetano et al. [250] (given in [250])	67.7	66.9
	Ke et al. [259] (given in [258])	58.4	57.9
	Ke et al. [260] (given in [258])	62.2	61.8
Ours	Skeleton (convnet) + RGB	73.9	75.4
	Skeleton(convnet + LSTM) + RGB	76.7	77.9

6.5 Conclusion and Future scope

With this chapter, we have tried to tackle the task of activity recognition by employing multiple vision cues (skeleton Joints and RGB videos) which are accessible from the RGB-D sensor. The proposed system utilizes a combination of convnet and LSTM. The convnet simply extracts spatial features from an image and the LSTM was inclined to process a sequence of inputs. The given method also discussed an algorithm for the creation of skeleton images using skeleton joint sequences. The cyclic learning rate was used to build the single training of the whole multi-modal system, which makes the system efficient and robust. Experimental results on the UTD-MHAD, CAD-60 and NTU-RGB+D120 datasets show that the proposed technique gives competing results to the other state-of-the-art methods.

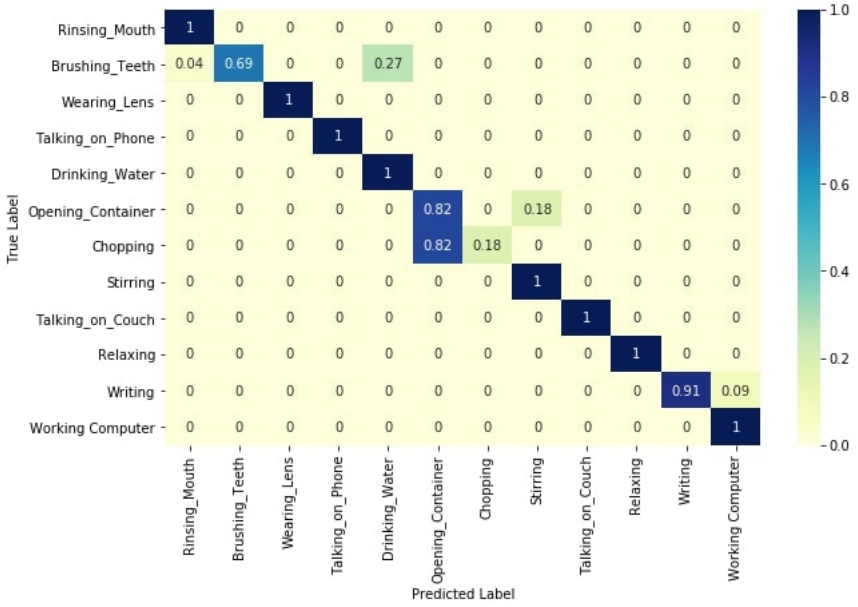
TABLE 6.5: Confusion Matrix of 27 classes of UTD-MHAD



The method performs amazingly with the indoor scene. There is a future scope towards making the system more robust for the real-time environment by including the external conditions or features. We will also use the multi-view data along with the multi-modal concept to make the system more accurate.

Our method performs amzingly with for indoor scenes static environment. We aim to make our model robust to external conditions and get an output accuracy good enough such that the model can be applied in real-time cases. We aim to get depth values of joint locations from a 2D image directly, instead of using depth sensors. Using this we

TABLE 6.6: Confusion Matrix of 12 classes of CAD-60.



will have a fully functional model which would recognise an activity directly from a RGB video feed, without requiring any depth information. This would reduce the cost of operation and the model could be used by any organisation having simple RGB cameras.