Human Activity Recognition Using Motion History Images and Deep Learning

Aral Sarrafi
Department of Mechanical Engineering
University of Massachusetts Lowell
Lowell, Massachusetts, 01854
aral_sarrafi@student.uml.edu

***Abstract*: With in this paper a hybrid human activity recognition algorithm is studied. This approach uses optical flow to estimate the relative motion in the field of view and then constructs the motion history images (MHI) associated with each of the activities being performed in the video. Then then the motion history images are classified by mean of the LeNet-5 convolutional neural network. The performance of the algorithm is evaluated on the WEZMANN data set consisting of 90 videos with 10 different classes.**

Keywords: Human Activity Recognition, Convolutional Neural Network, Deep Learning

I. Introduction

Image classification has been being studied for a long period in computer vision [1]. Most of the effort in this scope has been focused on classifying static images. Activity recognition is a more challenging task in which the objective is to predict the correct label for a video [2]. A video is in fact a sequence of static images, which makes this problem even more complicated than the image classification or object recognition. The previous researches available in literatures has proved that as expected motion plays a major role in video classification and activity recognition. It has been shown that humans are able to recognize the correct labels for an activity even from the highly blurred sequence of images. This paper will be using the key role of motion estimation in activity recognition by computing the motion history images (MHI) [1]. In previous studies for MHI based activity recognition the classification of the images, where conducted by means of Hu moment which are known to provide a reasonable shape discrimination that is translation and scale-invariant. Afterwards the extracted Hu moments were classified by means of classical classification methods such as support vector machines (SVM).

With in this paper the classification of the MHI images will be conducted differently. The feature extraction from MHI images by Hu moments is eliminated and classification process will be performed by a LeNet-5 deep neural-network.

The paper starts with a brief theoretical background on the motion estimation and optical flow, followed by introducing the motion history images. In section three the LeNet-5 deep neural network will be discussed. The last section will represent the results and evaluates the performance of the algorithm.

II. Motion estimation

Optical flow methods could be considered as one of the earliest approaches in computer vision for motion estimation [3]. A large group of optical flow estimation methods including Lucas-Kanade are based on brightness consistency assumption and the fact that the motion in two consecutive frames are relatively small and Taylor series expansion could be used to obtain a linear sets of equations which are straightforward to solve with digital computers [3]. Since Lucas-Kanade trackers are based on Lucas-Kanade optical flow estimation, a brief discussion on the theoretical background of Lucas-Kanade optical flow estimation is presented in this section. Assuming that the image intensity for the frame at time t is $I(x,y,t)$ and $I(x+\Delta x, y+\Delta y, t+\Delta t)$ at time $t+\Delta t$, in order to satisfy the assumption of the brightness consistency the equation (1) should be true.

$$I(x,y,t) = I(x+\Delta x, y+\Delta y, t+\Delta t)\,. \tag{1}$$

Assuming that the motion between two consecutive frames are small the Taylor series expansion could be utilized to expand the right side of the equation (1) as follows.

$$I(x+\Delta x, y+\Delta y, t+\Delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + H.O.T \tag{2}$$

Substituting (2) into (1) and dividing by Δt will result the flowing equations for each pixel which is known as the optical flow general equation or referred to as the aperture problem in computer vision community.

$$\frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0\,. \tag{3}$$

It is more conventional to write equation (3) in the following format in which V_x and V_y are the components of optical flow vectors at each pixel.

$$\frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + I_t = 0\,. \tag{4}$$

As it is clear, there is only one governing equation at each pixel and two unknowns V_x and V_y, as a result, there is no unique solution for the aperture problem to estimate the optical flow components. Several approximate methods have been introduced in order to obtain the optical flow vector field including Lucas-Kanade [4] and Horn-Schunck [5]. In this paper iterative Lucas-Kanade method has been used to solve the aperture problem to extract the motion field between two consecutive frames of a sequence of images. The underlying assumption in Lucas-Kanade method is that the nearby pixels have equal optical flow vector components. In this case additional constraints are introduced to the problem and made the aperture problem an over constrained set of equations. This over constraint set of equations are then solved using the least squares algorithm.

III. MOTION HISTORY IMAGES(MHI)

Motion history images are the core part of this algorithm for activity recognition. MHI can be computed based on the estimated motion using the optical flow theory that was discussed in the previous section. Let $D(x,y,t)$ be a binary image sequence indicating regions of motion. We will be computing $D(x,y,t)$ by applying a threshold to the amplitude of motion in a specific pixel. Thereby, the motion history images can defined as below:

$$H_\tau(x,y,t)=\begin{cases}\tau & if \quad D(x,y,t)=1\\ max(0,H_\tau(x,y,t-1)-1) & otherwise\end{cases} \qquad (5)$$

The motion history images gives the maximum value to the most recent pixels that have moved, and decreases this value for the pixels that have moved in previous frames. In other words, the pixels that are moved most recently will be having larger values and the pixels. MHI is a useful way of mapping activities in videos (a 3D signal) to a 2D image while preserving some of the temporal information that is associated with the video. In the results and discussion section the motion MHI for different actions will be presented.

IV. LENET-5 DEEP NEURAL NETWORK ARCHITECHTURE

After extracting the MHI form optical flow the action recognition task can be completed with classifying the MHI images. As mentioned in the introduction section the previous researches on using MHI for activity recognition have used Hu moments for feature extraction and a support vector machine for classification. We will be using a different approach by combining the features extraction and classification in LeNet-5 deep neural network, which will be trained on the data set. In fact, LeNet-5 [6] will handle both the feature extraction and classification together. The architecture of LeNet-5 has been discussed thoroughly in the literature, and we will be providing an overview of this deep neural network for the record.

Figure shows the architecture of the LeNet-5 neural Network.

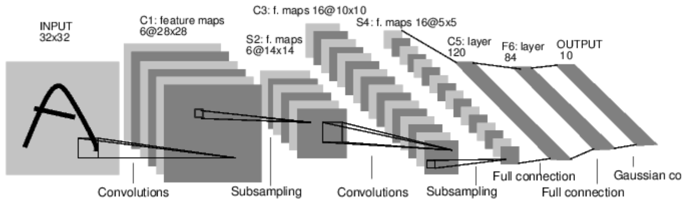

Figure 1. LeNet-5 deep neural network architecture[6]

The architecture of the LeNet-5 can be summarized as follows:

Convolutional Layer: 6 filters, filter size: (5, 5) , stride : 1
Max Pooling Layer: pooling size: (2, 2), stride, (2, 2)
Convolutional Layer: 16 filters. filter size: (5, 5), stride: 1
Max Pooling Layer: pooling size: (2, 2), stride, (2, 2)
Fully connected Layer with 120 neurons
Fully connected Layer with 84 neurons
Output layer with 10 classes (number of activities in the data set).

Moreover, after all the convolutional layers and fully connected layers a Rectified Linear Unit (Relu) activation function is applied to introduce non-linearity to the model.

This network is implemented in Tensorflow using the Keras user interface for easier implementation. Moreover, dropout layers were added to fully connected layers of the network to avoid any potential overfitting of the training data.

V. RESULTS AND DISCUSSION

Within this section the results of the several steps of the algorithm is presented. The data set that has been studied in this paper consisting of 10 video classes. First only one of the videos from the data set will be selected to show the procedure to compute the, motion history images, and the examples of motion history images for all the other activities will be presented.

A. Motion history images for jacking

In this section, the formation of the motion history images for the jacking activity will be presented. The figure below shows 6 different frames of a video from a person while jacking.

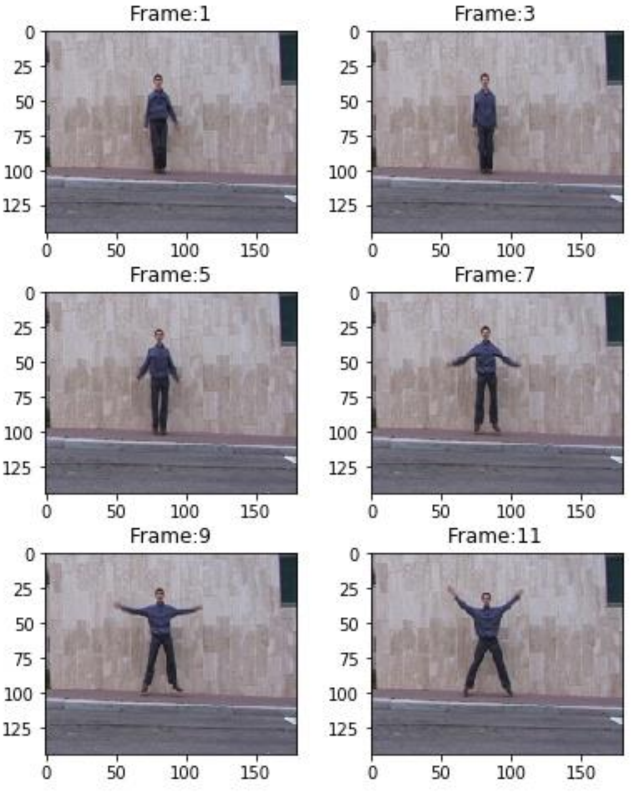

Figure 2. six different frames of a video from the jacking action

As mentioned earlier the first step in this action recognition pipeline is motion estimation. There are several motion estimation methods available in OpenCV, and we will be using the Lucas-Kanade method to obtain the motion field.

Figure 3 shows the estimated motion vectors associated with each of the frames. It should be also highlighted that the motion filed will be computed for each frame with respect to the next frame. Figure 3 is a representation of dense optical flow and for each pixel a vector has been plotted which is a bit hard to see because of the large number of vectors concentrated in the image.

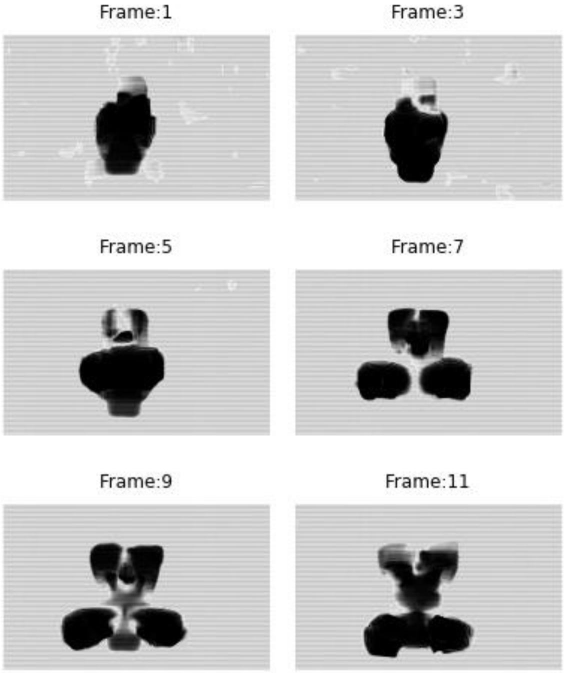

Figure 3. Dense optical flow for at 6 different frames of the jacking video

Motion filed is a vector filed and as any other vector field at each pixel the amplitude and direction of the vector can be obtained. In this paper, we are only concerned with the amplitude of the motion vector. The amplitude of the motion field will be used to determine the pixels that have moved. In other words the amplitude of the motion filed will be used to acquire the binary sequence $D(x,y,t)$ which is essential for evaluation of the motion history image. The amplitude of the motion filed at each pixel for 6 different frames are presented in the Figure 4.

The next step of this activity recognition approach is the from the binary image sequence indicating the regions of motion $D(x,y,t)$. Applying a threshold on the amplitude of the motion filed which was estimated in the previous section will form the $D(x,y,t)$. The threshold is to 15% of the maximum value of the motion amplitude for each frame. All the pixels with a motion amplitude less than the threshold will be set to zero and all the other pixels with a motion amplitude of larger than the threshold will be set to 1. The binary sequence of images $D(x,y,t)$ for 6 different frames of the jacking are presented in Figure 5. Now that the $D(x,y,t)$ is available the motion history image (MHI) can be computer based on the definition as described in equation (5). The motion history image of this specific jacking video is shown in Figure 6. Ideally, motion history images for different samples of an action should be very similar, and across different actions the motion history images should have distinct patterns. At this point all the video can be converted to motion history images, and the rest of the pipeline will be working with 2D images, and the activity recognition from videos will be reduces to an image classification problem that has been studied widely in the literature of computer vision.

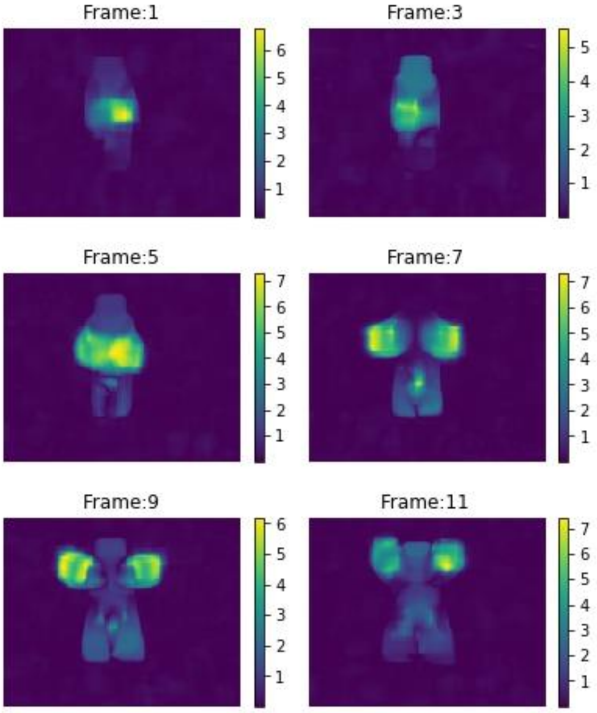

Figure 4. Amplitude of the motion field at 6 different frames of the jacking video

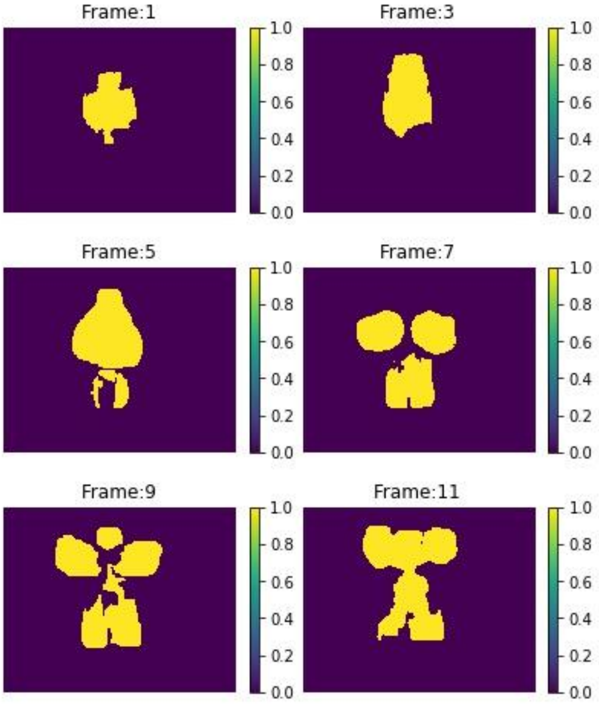

Figure 5. Binary images indicating the regions of motion at each frame

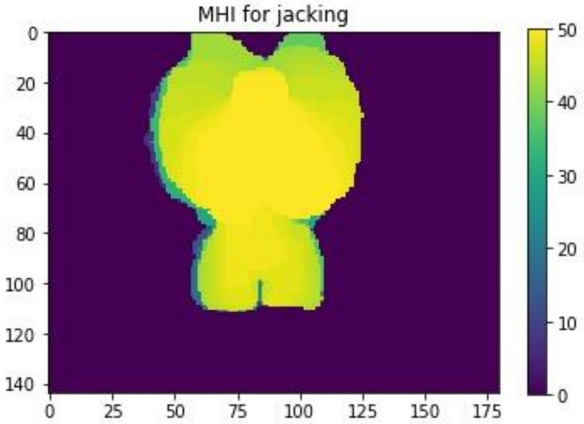

Figure 6. Motion history image (MHI) for an example jacking video

Figure 7 shows the sample motion history images for all the different actions that are included in the data set.

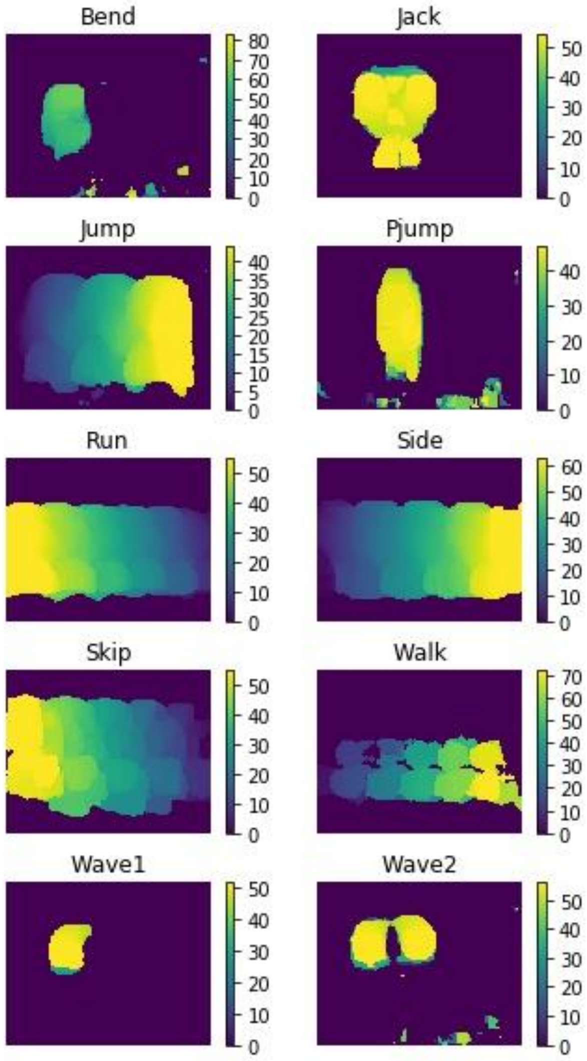

Figure 7. Sample motion history images for 10 different activities

Now that all the motion history images are available, the MHIs can be fed to the LeNet-5 for training. Before training the network a simple data augmentation is also applied to the motion history images by flipping the image along the vertical axis which doubles the number of MHIs. However, even after applying data augmentation the data set that has been studied in this research enough data points. Therefore, the neural network will be overfitting. The only way to avoid overfitting to have more data from different action. Considering this, we will not be splitting the data to training and validation set, because the data set is very small consisting of only 90 videos and after data augmentation the data points will be increased to 180 which is very small for deep learning applications.

Similar to other classification approached in deep learning the categorical cross entropy function is used as the loss function for this study. The model has been trained for 300 EPOCHS. Figure 8 shows the loss function versus each EPOCH during the training process. As it is expected the loss function will decrease generally as the number of EPOCHs increases and the accuracy of the model will increase respectively. The best way to improve the results of this approach is collecting more data because generally deep learning approaches require large amount of data. Having large amount of data will help to train more accurate models while avoiding overfitting.

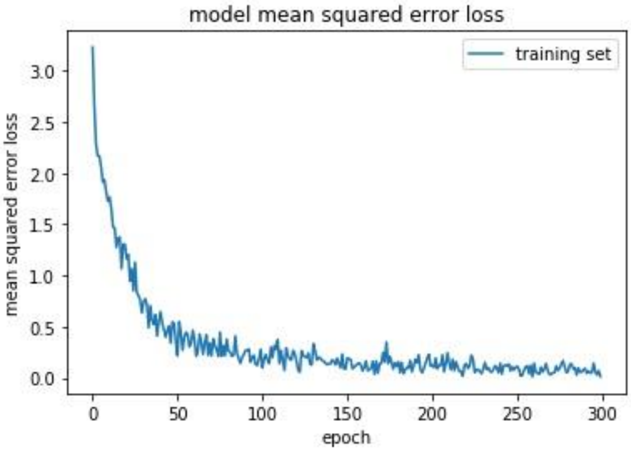

Figure 8. Categorical cross entropy loss function

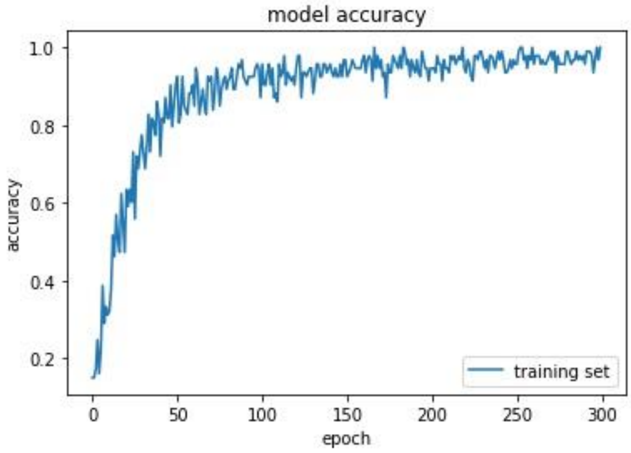

Figure 9. Accuracy of the model

VI. Conclusion

Within this paper, a new approach for human activity recognition has been proposed. This algorithm builds on top of some of classical approaches for activity recognition including motion estimation and motion history images. The process of activity recognition starts with estimating the optical flow from the sequence of images. Afterwards a threshold was applied on the amplitude of the motion to find the pixels that were moving in that specific frame. The motion history images were formed based on the sequence of binary images which were indicating the pixels that have moved. This procedure maps the videos as 3D signals to a 2D image. Therefore, the complicated task of action recognition will be reduced to an image classification problem. In the last stage of the algorithm the LeNet-5 deep neural network have been used for image classification. The data set has been studies in this research does not have enough data for deep learning algorithms to perform well, and the deep neural network is very likely to overfitting. It highly recommended to use larger data sets to avoid overfitting and to obtain better results from the proposed activity recognition pipeline in this paper.

VII. References

[1] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 23, pp. 257-267, 2001.

[2] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing,* vol. 28, pp. 976-990, 2010.

[3] D. Fleet and Y. Weiss, "Optical flow estimation," in *Handbook of mathematical models in computer vision*, ed: Springer, 2006, pp. 237-257.

[4] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision,* vol. 56, pp. 221-255, 2004.

[5] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence,* vol. 17, pp. 185-203, 1981.

[6] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker*, et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, p. 276, 1995.