Human Motion Generation Using Variational Autoencoder and Mixture Density Network

Wafaa Shihab Ahmed [a] Abdulamir A. Karim [b]

a University of Technology/Department of Computer Science/ Baghdad, Iraq
b University of Technology/Department of Computer Science/ Baghdad, Iraq/
111789@student.uotechnology.edu.iq, 110004@uotechnology.edu.iq

Abstract. The modelling of Human motion is critical in several fields such as computer graphics, vision and virtual reality, with applications of interaction between human and computer, synthesizing motions, and motion generation for virtual reality. In this work the CNN-VAE and RNN-MDN models have been used for modelling the human motions by learning time-dependent representations to obtain short-term motion prediction and long-term human motion synthesis. The new generated human motion video achieved good quality. This proposed system has been implemented on the KTH dataset and Weizmann dataset to generate the boxing and waving motions.

Keywords: Variational Autoencoder (VAE), Convolution Neural Network (CNN), Mixture Density Network (MDN), Wavelet Transform (WT).

1 Introduction

Natural human motion modeling is a hot topic in a variety of fields, including computer animation, biomechanics, virtual reality, and others, where high-quality motion data is required. Despite motion capture systems' better accuracy and decreased costs, it is still highly desirable to make maximum use of existing data to generate diverse new data. One of the most difficult aspects of motion production is dynamics modeling, where it has been demonstrated that due to the great coordination of body motions, a latent space can be discovered [1, 2, 3]. However, despite the fact that the spatial component is being researched, dynamics modeling, particularly with the goal of diverse motion creation, remains an unsolved challenge [4].

The goal of human motion video generation is to create high-fidelity future frames by learning dynamic visual elements from video. Because the model may have to learn to separate a number of effects based on dynamic visual properties, such as how objects move and distort over time, how sceneries change as the camera moves, and how the background changes, it's an excellent method for learning video representations [5, 6].

Researchers have recently focused on using deep recurrent neural networks (RNNs) to model human motion, with the goal of learning time-dependent representations that perform tasks such as predicting short-term human motions and combining long-term human motions, based on the success of deep learning techniques in a variety of computer vision tasks [7,8]

One of the deep neural network methods which has been used in this paper is convolutional neural network with variational auto encoder (CNN-VAE) model and MDN-RNN (LSTM) model. In this paper the wavelet transform has been used with CNN-VAE model to analyze the input data to multi Structure scales and to make the time of training and testing faster and MDN with RNN have been used for predicting the new distributed latent vairables to generate new video.

2 Related Work

Below are some related works clarify some methods used for generating the human motion.

K. Fragkiadaki et al., in 2015 [9], they proposed a model of Encoder-Recurrent-Decoder (ERD) to recognize and predict the position of human body in video and in motion capture. The human motion temporal dynamic learned by a long short term memory (LSTM) model. They constructed a nonlinear transformation to encode the features of human pose and decode the LSTM output. They tested representations of ERD architectures to generate motion capture (mocap), labeling pose of body and predicted it in video. They tested this model on the dataset named H3.6M, which is consider largest dataset for video pose. P. Ghosh et al., in 2017 [10], Proposed a modern framework to learn the models of spatio-temporal motion prediction from data only. This approach, known as the Dropout Autoencoder LSTM (DAELSTM), will synthesize natural sequences of motion over long-term horizons1 without drastic drift or loss of motion. This Dropout Autoencoder (DAE) then is used by a 3-layer LSTM network to filter each expected pose, reducing the accumulation of associated errors and, subsequently, drifted over time. R. Villegas et al., in 2017 [11], proposed a deep neural network to predict future frames of realistic video sequences. To solve complicated development of pixels in video, they proposed decomposing motion and content, two main components producing dynamics in video. This model built for pixel level forecasting by the Encoder-Decoder Convolutional Neural Network and Convolutional LSTM, which separately identify the spatial structure of an image and the associated temporal dynamics. Trying to predict the next frame by separately modeling motion and content decreases the conversion the extracted features of content to the next frame content by the motion features defined, which simplifies the prediction job. They evaluated the proposed system on videos of human motion, using KTH, Weizmann action, and UCF-101 datasets. C. Li et al., in 2018 [12], they presented a new approach built on convolutional neural networks (CNN) for modelling human motion. The encoder of the long-term and encoder of the short-term have the same architecture, i.e. the CEM, which consist of three convolution layers and one fully connected layer. For each convolution layer the number of feature maps was 64, 128 and 128, and for fully connected layer the number of the output nodes was 512. A stride number for each convolution layer is set 2 to capture the long term correlations and enhance the accuracy of prediction. So they suggested a model of convolutional sequence-to sequence to predict human motions. They adjusted 2 types of convolutional encoders, the encoder of long-term and encoder of short-term, so that the information of the both distant and temporal motion used to predict the future. In the long term prediction this model outperform on state-of-the-art RNN models, in the testing, they used 2 datasets: the dataset named Human 3.6M and dataset named Motion Capture CMU. Y. Li et al., in 2018 [13], proposed a conditional variational autoencoder (cVAE) dependent on probabilistic models, for modeling the uncertainty. There are two unique attributes of their probabilistic model. Firstly, this model is a 3D-cVAE, i.e. the autoencoder is built in an architecture of spatialtemporal convolutions used to predict consecutive optical flows. Secondly, is the method of frame generation named the Flow2rgb model, the model will "imagine" the existence of the next frame by flow and start frame. A spatial temporal correlations and future uncertainty have been modelling in a 3D-cVAE model. For evaluating the model they testing their algorithm on 3 datasets. The KTH dataset, and 2 datasets the Waving Flag and Floating Cloud which collected form website. These 2 datasets represent dynamic texture videos. A. Augello, et. al., 2017 [14]. Proposed a approach of deep learning to introduce a computational creativity behavior in a dancing robot. They used the variational autoencoder for converting the input to latent representation. The generation has been achieved by injecting the representations of the listened music into the encoder network's latent space. This method applied on a set of movements captured from various skilled dancer. As a result, the robot can improvise dancing movements based on the music being played, even if it has never been done before.

3 Transform Coding

The wavelet transform has been used in this work to transform the frame of video from spatial domain to frequency domain and the result will image decomposed into four subbands (LL, LH,HL and HH). This is done by applied the haar wavelet transform equations.

a) Forward Haar Wavelet Transform (FHWT)

Given an input sequence (xi) i=0…N-1, it is FHWT produce (Li) i=0…N/2-1 and (Hi) i=0…N/2-1 by using the following transform equations [14]:

1. If N is even

$$\left.\begin{aligned} L(i) &= \frac{x(2i)+x(2i+1)}{\sqrt{2}} \quad ,i=0\ldots\left(\frac{N}{2}\right)-1 \\ H(i) &= \frac{x(2i)-x(2i+1)}{\sqrt{2}} \quad ,i=0\ldots\left(\frac{N}{2}\right)-1 \end{aligned}\right\} \tag{1}$$

2. If N is odd

$$\left.\begin{aligned} L(i) &= \frac{x(2i)+x(2i+1)}{\sqrt{2}} \quad ,i=0\ldots\left(\frac{N-1}{2}\right) \\ H(i) &= \frac{x(2i)-x(2i+1)}{\sqrt{2}} \quad ,i=0\ldots\left(\frac{N-1}{2}\right) \\ L\left(\frac{N+1}{2}\right) &= x(N-1)\sqrt{2} \\ H\left(\frac{N+1}{2}\right) &= 0 \end{aligned}\right\} \tag{2}$$

b) Inverse Haar Wavelet Transform (IHWT)

The inverse one-dimensional HWT is simply the inverse to those applied in the FHWT; the IHWT equations are [15]:

1. If N is even

$$\left.\begin{aligned} x(2i) &= \frac{L(i)+H(i)}{\sqrt{2}} \quad ,i=0\ldots\frac{N}{2}-1 \\ x(2i+1) &= \frac{L(i)-H(i)}{\sqrt{2}} \quad ,i=0\ldots\frac{N}{2}-1 \end{aligned}\right\} \tag{3}$$

2. If N is odd

$$\left.\begin{aligned} x(2i) &= \frac{L(i)+H(i)}{\sqrt{2}} \quad ,i=0\ldots\frac{(N-1)}{2} \\ x(2i+1) &= \frac{L(i)-H(i)}{\sqrt{2}} \quad ,i=0\ldots\frac{(N-1)}{2} \\ x(N-1) &= L\left(\frac{N+1}{2}\right)\sqrt{2} \end{aligned}\right\} \tag{4}$$

4 Convolution Neural Network (CNN)

CNN consider, for deep learning, a very popular model which are specifically suitable with images as inputs, but in the same time they are often used in other tasks such as text, signals and other continual responds. The main difference between the CNN and NN is the CNN received image as input while the NN input is numerical value (e.g. a feature vector). There are three main layers included in CNN which are convolutional layers, subsampling layers (pooling layers), and finally fully-connected layers [16, 17, 18].

5 Variational Autoencoder (VAE)

Autoencoders have sparked a lot of attention. Because this technique can do data compression in lossy begining from a specific database, Once trained, they represented all previously exposed data in the hidden layer; their presentation is "lossy" since the produced x didnot not completely similar to the original, with the differences specified by the distance or "error" function. Compression and prediction are closely related fields, as shown in [19], and compressors could be employed to produce new data. The ability of VAE, first presented in [20], to generate a variation of learning input data.

The compression and prediction, as it shown in [19], are closely related fields, and new data can also be produce by using the compressors. Variational autoencoders, that introduced firstly in [20], have a strong interest for their capabilities to generate a variation of the input data learning. The input data can be represented by the capability of the most interesting features which autonomously draw the boundaries of the compressed space. Given input data x and calling p(x) the probability distribution of the data, the latent variable z has been learning with its probability density p(z) so the data has been generated when the value of z are varied:

$$p(x) = Zp(x|z)\,p(z) \qquad (5)$$

To estimate the distribution p(x|z), the variational autoencoder training is depended on the variational inference which is often used with Bayesian method when there is a desire to deduce a posterior that is hard in computation. To reduce the Kullback-Leibler divergence between two distributions a simpler distribution qλ(z|x) has been selected . A family of distributions is referred to as a variational parameter λ, which in the case of a Gaussian family would represent mean and variance. The divergence is computed in equation (6).

$$DKL(q\lambda(z|x)\|p(z|x)) = Eq[logq\lambda(z|x)p(x)\,p(x,z)] \qquad (6)$$

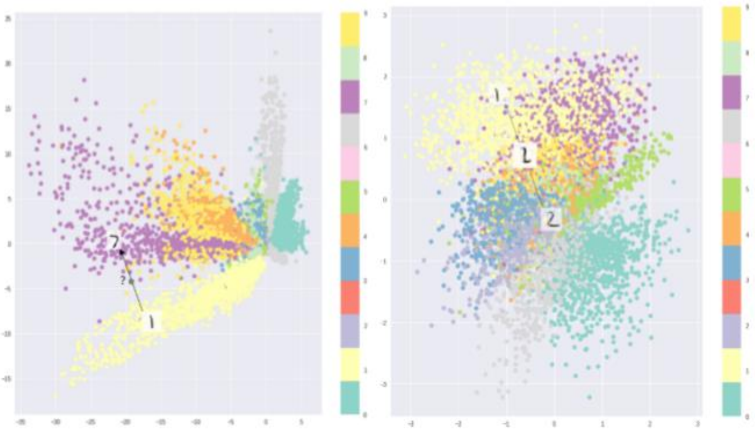
It can be demonstrated that:

$$\log(p(x)) = Lv + DKL(q\lambda(z|x)\|p(z|x)) \qquad (7)$$

Since, to reduce the log(p(x)) it is sufficient to reduce Lv. This value can be computed in equation (8):

$$Lv = -DKL(q(z|x)\|p(z)) + Eq(z|x)\log(p(x|z)) \qquad (8)$$

The distribution of p(z) as similar as possible to q(z|x), where the first term (DKL(q(z|x)||p(z)) is refers to regularization part, whereas the second part Eq(z|x)log(p(x|z)) put into consideration a suitable reconstructing of the values of x. After the training stage objected to minimize the value of log(p(x)), This is the same as saying "maximize the likelihood." As a result, the zeta values indicate the optimal compression for the input values, and the variance in the z space corresponds to a variation in the input sample construction. [14]



(a) Autoencoder

(b) Variational autoencoder

6 Long Short Term Memory (LSTM)

It is very difficult to train the Standard RNNs in a stable way (concerned to the problems of vanishing/exploding gradient) so that a Long Short-Term Memory (LSTM) type of RNN is used. Long training runs keep LSTMs stable, and they can be layered to construct deeper networks without losing their stability [21]. Unlike a traditional RNN, which uses basic recurrent neurons, an LSTM's central unit is a memory cell that maintains a state across time and is controlled by gates that regulate signal flow in and out of the cell. . Because the signal flow is precisely controlled, the risk of overloading or extinguishing the cell through positive or negative feedback is reduced. The relationships in an LSTM cell are shown in the following equations (see figure 2) [22]:

Fig 1 : Example of the difference in latent space density between an autoencoder and a variational autoencoder, trained on the MNIST dataset] [20].

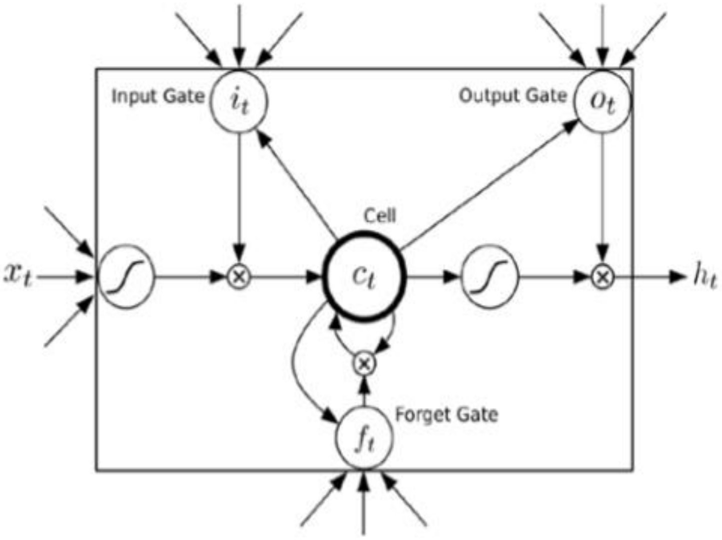


Fig 2. : LSTM Neuron [22]

The memory cell c works as the major innovation of LSTMt, which essentially represents as an accumulator of the state information, and numerous self-parameterized controlling gates update it. If the input gate is engaged when a fresh input xt reached at time t, its information will be gathered in the memory cell. Also, if the forgotten gate ft is on, the previous cell ct−1 may be forgotten throughout this operation. The output gate ot controls whether the most recent cell output ct is transferred to the last state ht. All gates it, ft, ot, memory cell ct, and hidden state ht are nstate ×1 vectors with the same dimension. The hidden state h0 and memory cell c0 are all set to zero vector 0 at time 0. The forward pass of the LSTM is executed at time step t (t = 1,2,...,T), given ht−1,ct−1 from the last time step t−1 and current input xt, by first calculating the modified memory cell cť:

$$\widetilde{c_t} = tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \qquad (9)$$

The an input gate it is established to regulate how much information in cell cť should be flowed into new memory and a forgotten gate ft is also created to regulate how much information from a previous memory cell ct−1 must be remembered:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \qquad (10)$$

$$f_t = \sigma\left(W_{fx}x_t + W_{fh}h_{t-1} + b_f\right) \qquad (11)$$

$$c_t = i_t{}^\circ\widetilde{c_t} + f_t{}^\circ c_{t-1} \qquad (12)$$

Lastly, the output gate ot is used to verify what part of the memory cell ct must be sent to the hidden state ht:

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \qquad (13)$$

$$h_t = o_t{}^\circ tanh(c_t) \qquad (14)$$

The symbol ° represent the element wise multiplication of the vectors.

The gradient will be trapped in the cell and maintained from fading too soon that is a common problem in RNN, which is one advantage of using the memory cells and gates to administrate the flowing information. Fully Connected LSTM (FC-LSTM) is another name for this multi-variate input version of LSTM [23].

7 Mixture Density Network (MDN)

MDN is a technique that has been successfully used to a variety of applications, including robotic arm control [24], handwriting generating [25], and now human motion generation. We output a probability density function for each dimension in the tensor rather than simply a single location tensor. The LSTM provides a layer of linear output units that act as parameters for a mixture model defined as the probability of a target t given an input x as in equation (15) [22]:

$$P(t|x) = \sum_{k=1}^{K} \pi_k(x)N(t|\mu_k(x), \sigma^2\,(x)) \qquad (15)$$

Where K refers to number of components in the mixture, and π is the mixing coefficients as a function of the inputs (x), means (μ) is component location, and standard deviations (σ) is component width. The height of each curve is a weight (π). The number of mixture components, K, is arbitrary and can be considered of as the number of diverse options available to the network at each time step [22].

8 The Proposed System

In the proposed system CNN-VAE model and LSTM model have been used to learn the representation of the input subband frames (LL) which acts the human motions such as (walking, boxing and waving). CNN-VAE model including cnn-encoder and cnn-decoder. The encoder receive the features from cnn and representing as latent variables by compute the variance (σ) and mean (μ) values and used in sampling operation using equation (16).

$$Sampling = \mu + exp(0.5 * \sigma) * epsilon \qquad (16)$$

Where epsilon is random normal [0,1].

These latent variables have been learned by an encoder. The decoder part is the opposite of the encoder, where the sampled point is entered as input to the dense layer (fully connected) for decoding the values and the output of this layer will enter as input to the convolution layers. The result (Y) is compared with input image (X) and compute the loss value by using equation (8) , the kullback divergence value has computed by equation (6). These steps repeated until reach to minimum loss value. The weights have been saved in file (vae-weights.h5).

CNN-VAE model has been illustrated in figure 3. This model consist of two phases, training and prediction phases.

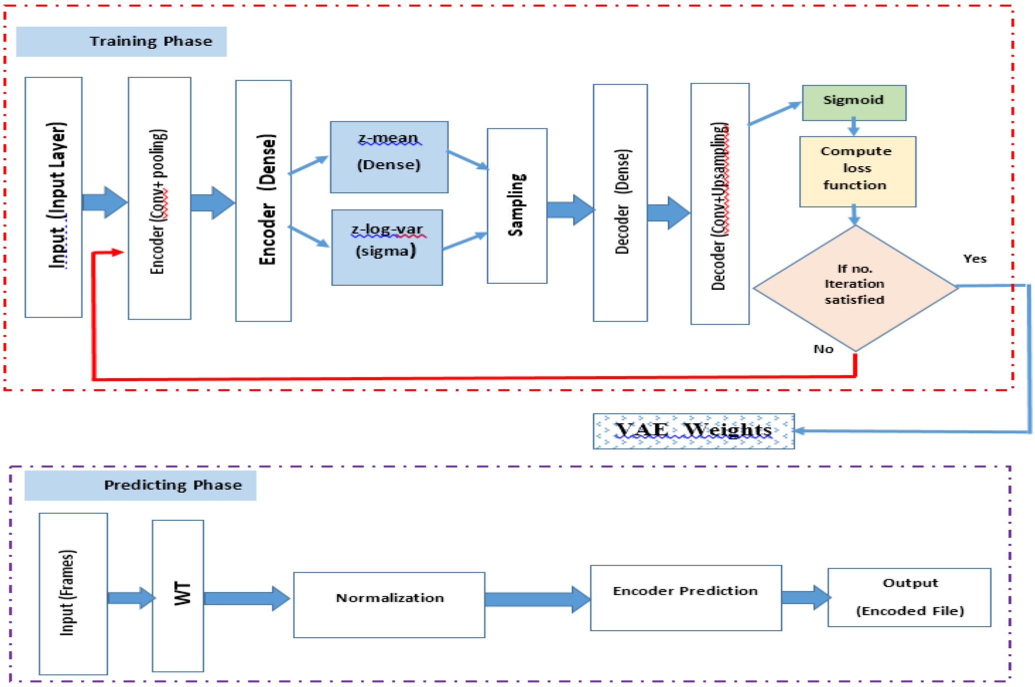


Fig 3. : CNN-VAE model

Figure 3. - The Structure of proposed CNN-VAE Model

In the proposed system the process of generation is based on CNN-VAE model and on LSTM model. CNN_VAE model used to extract features and encoding it, after that the LSTM model has been used for training the encoding data (compressed data) to generate new frames. Figure. 4. Show the LSTM –MDN for training.

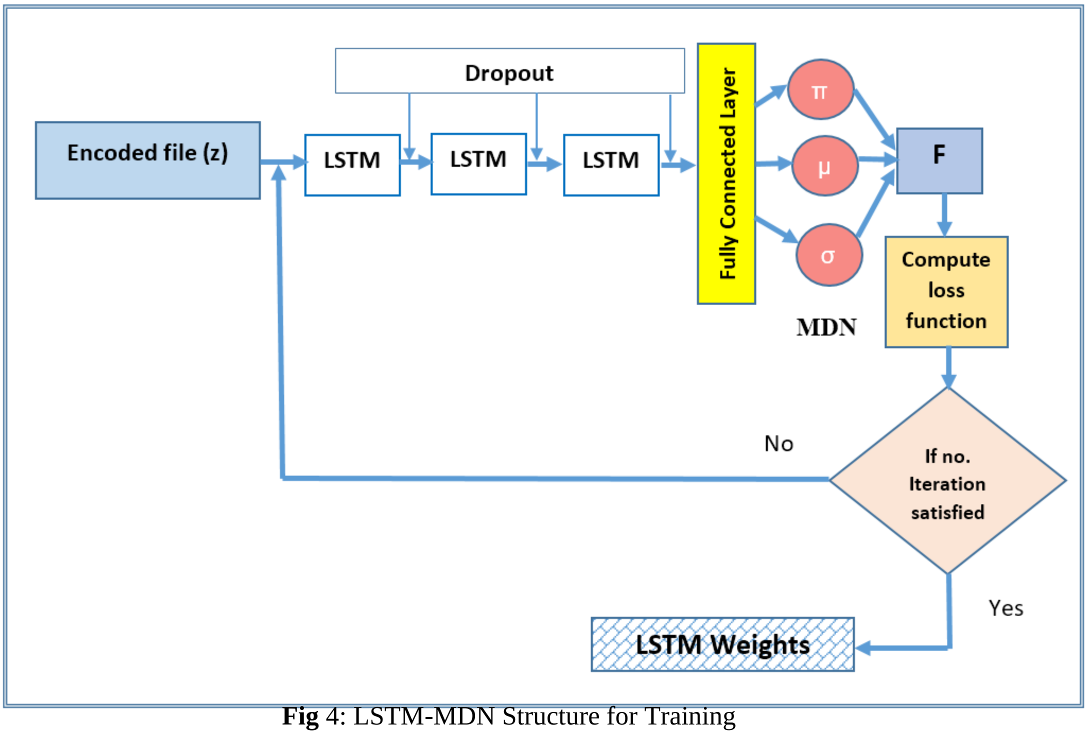


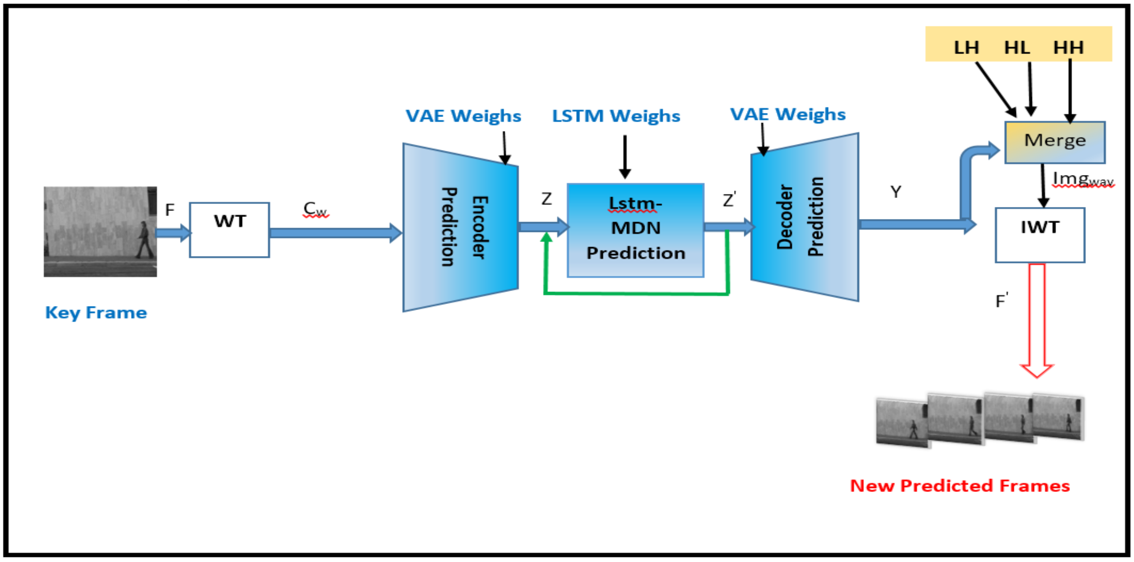
Fig 4: LSTM-MDN Structure for Training

8.1 The Training System

In the proposed system there are two training: training the CNN-VAE model and training the LSTM-MDN model. In the training CNN-VAE model the CNN-VAE encoder includes three convolutions layers with different number of filters (32, 64 and 128) and filter size 3×3. The number of max pooling layers are three and two dense layers (fully connected layer) with number of nodes 128. The activation function which has been used with each convolution layer is Relu. in the CNN-VAE decoder the number of convolution layers are four with three upsampling layers and two dense layers, sigmoid activation function used in the last convolution layer.

In the LSTM training the input to this model is encoded representation which stored in file and three LSTM-MDN layers have been used in the training, each layer have 512 nodes, MDN layer have 24 components and after each layer the coefficients have been dropout. The output of lstm entered as input to the fully connected with number of nodes 1000. The error value between the input (Z) and output (Z ') has been computed by MSE measure. The weights of this network has been stored in file (lstm-weight.h5).

8.2 The Testing (generation) System In the testing or generation phase the future frames have been generated by using the weights of cnn-vae model and weights of lstm-mdn model.In the process of



generation as in figure. 5. The input frame will transformed from spatial domain to frequency domain by using haar wavelet transform (1 and 2) and normalized, the coefficients will be encoded and predicted LSTM-MDN model, the result will be the new predicted encoded samples these samples have been decoded by using CNN-VAE decoder and the same result will back to LSTM model as input to predict the next encoded samples. The CNN-VAE decoder produced new reconstructed image this image has been normalized to original range. The result image represent the LL band from the original image. Each of remaining bands (LH, HL and HH) have been training and predicted as the same steps which the LL band has been trained and predicted. All reconstructed bands (LL, LH, HL and HH) have been concatenated to produce the new image. This new image enter to the inverse wavelet transform using equations (3 and 4) to produce the new frame, these new frames have been converted to a video.

9 The Experiments Results

In this paper the experiments have been implemented on two datasets Weizmann and KTH datasets. The motions which have been generated are waving and boxing.Below are the datasets which have been used in this work.KTH dataset: This is includes 6 types of actions (boxing, hand clapping, hand waving, jogging, running and walking). This dataset contain on 599 action videos, these are taken by 25 various subjects with 4 scenarios (outdoors, outdoors with scale variations, outdoors with various cloths and indoors) [26, 27, 28]. This dataset is download from the website in reference.

Weizmann dataset: This dataset consists of 10 classes of actions like "walking", "jogging", "waving" taken by 9 separate individuals to get a sum of 90 video clips. The video is shot with a static camera under a simple background [15], [29]. This database is downloaded from the website in reference [30].

In this paper the wavelet transform has been implemented to transform the image from spatial domain to frequency domain the results of this implementation are show in figure. 6:

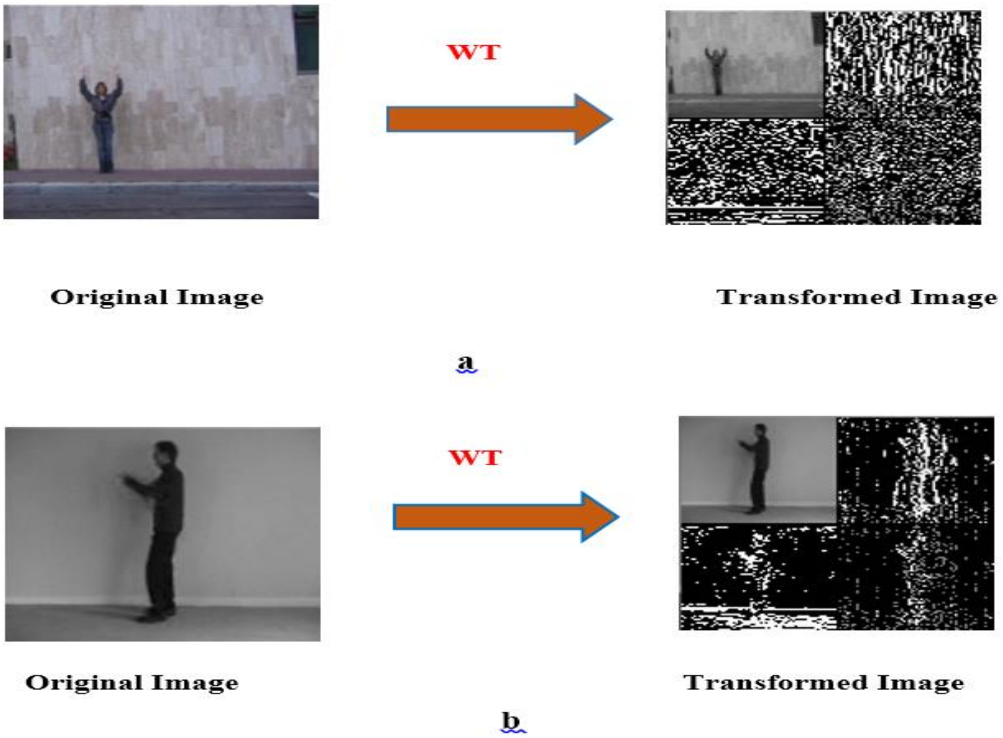


Fig 6: (a) Weizmann dataset for waving motion; (b) KTH dataset for boxing motion.

In this paper the experiments have been implemented on the subbands of transformed images in KTH dataset to generate new frames for boxing motion and implemented on Weizmann dataset to generate waving motion by using the proposed system. The PSNR, MSE and similarity SSIM measures have been computed to measure the quality of new frames. Table 1. Show the quality measures, number of generated frames and time of each generated video measured in millisecond (ms). In this work we used 10 frames per second.

TABLE 1: THE MEASURES VALUES OF QUALITY FRAMES FOR THREE VIDEOS OF WEIZMANN DATASET OF (WAVING ACTION)

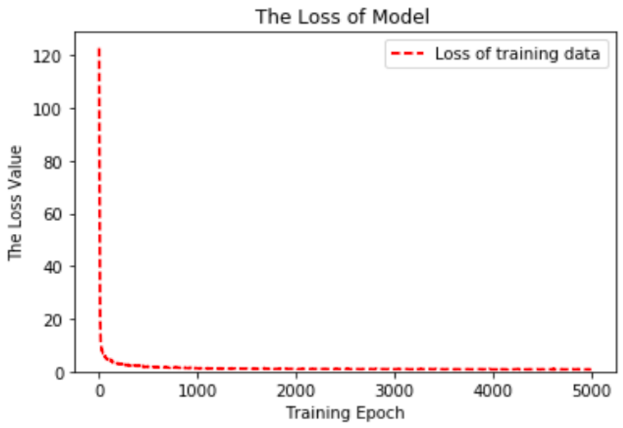
Dataset	Motion	Video	No. of generated Frames	MSE	PSNR	SSIM	Time of video 10 frame/s
Weizmann	Waving	Person 1	25	Max=8.17 Min=4.03 Ave=5.86	Max=42.08 Min=39.01 Ave=40.49	Max=0.987 Min=0.982 Ave=0.984	2500 ms
Weizmann	Waving	Person 2	15	Max=12.72 Min=4.84 Ave=6.60	Max=41.28 Min=37.09 Ave=40.10	Max=0.985 Min=0.973 Ave=0.981	1500 ms
Weizmann	Waving	Person 3	16	Max=13.72 Min=5.41 Ave=7.98	Max=40.80 Min=36.76 Ave=39.53	Max=0.982 Min=0.973 Ave=0.978	1600 ms
Weizmann	Waving	Person 4	21	Max=9.51 Min=6.20 Ave=7.28	Max=40.21 Min=38.35 Ave=39.14	Max=0.983 Min=0.979 Ave=0.981	2100 ms
Weizmann	Waving	Person 5	36	Max=13.03 Min=4.91 Ave=6.12	Max=41.22 Min=36.98 Ave=40.34	Max=0.985 Min=0.969 Ave=0.982	3600 ms
Weizmann	Waving	Person 6	25	Max=7.14 Min=4.65 Ave=5.77	Max=41.45 Min=39.60 Ave=40.54	Max=0.985 Min=0.979 Ave=0.983	2500 ms
Weizmann	Waving	Person 7	17	Max=9.38 Min=6.49 Ave=7.10	Max=40.01 Min=38.41 Ave=39.43	Max=0.985 Min=0.977 Ave=0.979	1700 ms
Weizmann	Waving	Person 8	18	Max=11.15 Min=5.41 Ave=7.49	Max=40.80 Min=37.66 Ave=39.45	Max=0.986 Min=0.980 Ave=0.983	1800 ms

The accuracy and loss value of CNN-VAE model training with number of epochs and batch size have been illustrated in Table 3 and in figure 7 and 8, the PSNR values for the two datasets have been illustrated in figure 9, figurer 10. Show the quantitative comparison between the ground truth (original) frames and our proposed model while figure 11 display the qualitative comparison between out model with others.

Dataset	Motion	Video	No. of generated Frames	MSE	PSNR	SSIM	Time of video
KTH	Boxing	Person 1	20	Max=26.33 Min=6.56 Ave=12.41	Max=39.96 Min=33.94 Ave=37.44	Max=0.987 Min=0.974 Ave=0.986	2000 ms
KTH	Boxing	Person 4	20	Max=59.91 Min=6.28 Ave=13.89	Max=40.15 Min=30.36 Ave=37.66	Max=0.991 Min=0.975 Ave=0.987	2000 ms
KTH	Boxing	Person 7	20	Max=28.33 Min=6.9 Ave=10.64	Max=40.21 Min=33.6 Ave=38.19	Max=0.981 Min=0.952 Ave=0.973	2000 ms
KTH	Boxing	Person 8	20	Max=14.86 Min=5.83 Ave=8.12	Max=40.48 Min=36.41 Ave=39.14	Max=0.987 Min=0.981 Ave=0.984	2000 ms
KTH	Boxing	Person 9	20	Max=25.08 Min=6.26 Ave=10.55	Max=40.16 Min=34.14 Ave=38.19	Max=0.985 Min=0.959 Ave=0.976	2000 ms
KTH	Boxing	Person 15	20	Max=16.33 Min=6.42 Ave=9.15	Max=40.05 Min=36.00 Ave=38.67	Max=0.992 Min=0.986 Ave=0.988	2000 ms
KTH	Boxing	Person 17	20	Max=13.96 Min=4.24 Ave=7.14	Max=41.85 Min=34.18 Ave=40.07	Max=0.988 Min=0.968 Ave=0.983	2000 ms
KTH	Boxing	Person 20	20	Max=35.61 Min=8.93 Ave=15.36	Max=38.62 Min=32.62 Ave=36.67	Max=0.962 Min=0.985 Ave=0.977	2000 ms

TABLE 3: THE ACCURACY AND LOSS VALUE FOR THE CNN-VAE MODEL TRAINING

dataset	No. of Training Frames	No. of epochs	Batch size	Learning rate	Accuracy of Training	Loss value
KTH	500	5000	50	0.001	0.92	0.082
Weizmann	201	5000	20	0.001	0.97	0.037



ı con

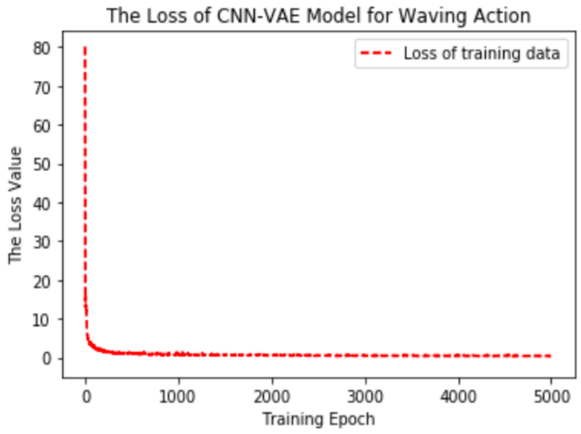
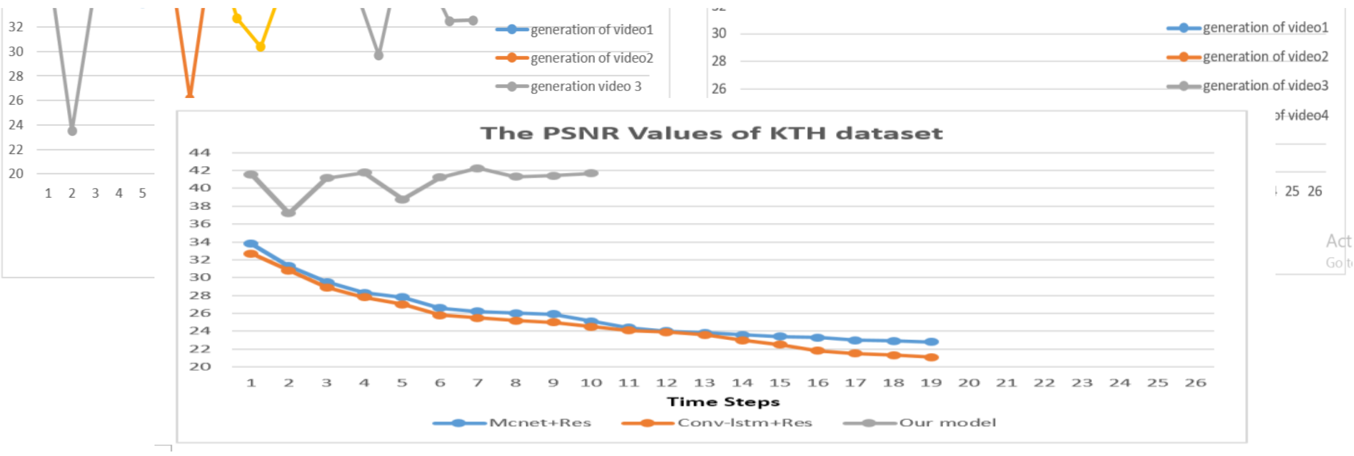


Fig 7: The loss value of cnn-vae model of KTH

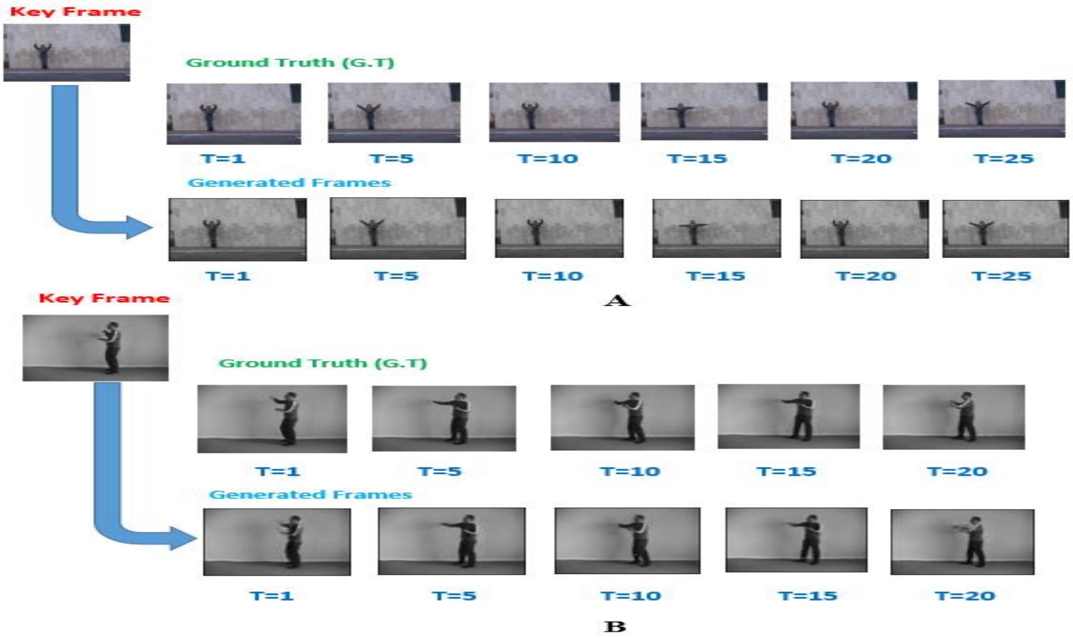
Fig 8:The loss value of cnn-vae model of Weizmann dataset.

Fig 9: The PSNR values for generated frames of (a) Weizmann dataset (waving motion), (b) KTH dataset (boxing motion).



figures showed that the our model achieved good generation comparison with other models because the quality of the

Fig 11: Qualitative comparison between our model and ground truth (A) Weizmann dataset for generation waving motion; (B) KTH datase for generation boxing motion.



generated frame have good values approximately the PSNR between 40 to 37 and the length of generated video exceeded 1000 ms While the length of the videos generated by other models did not exceed 1000 milliseconds. This due to that the prediction and generation in the proposed system depended on the probability distribution not on single position in tensor so in the end the system must predict sample has meaningful value and produce new predicted sample which the system decoding this new sample to produce new reconstructed frame.

10 Conclusion

In this paper the wavelet transform with VAE and LSTM-MDN layers have been proposed for generating new video from one input frame. The CNN-VAE has been used to extract features from input subbands and mapping features to latent space by computing the mean and variance values for each input sample this is done in encoder stage. The Mixture Density Network has been used to increase the probabilities of distributed each latent variable, and each predicted latent variable has been decoded and reconstruct the new frame. This model can produce new generated video with good quality in display.

References

[1] Holden, D., Saito, J., and Komura, T., 2016. A deep learning framework for character motion synthesis and editing. ACM Transactions on Graphics (TOG) 35, 4(2016), 1–11.

[2] Safonova, A., Hodgins, J. and Pollard, N., 2004. Synthesizing physically realistic human motion in low dimensional. ACM TransactionsonGraphics-TOG (012004).

[3] Wang, H., Edmond, SL, Ho, Hubert, PH. And Zhanxing, Z., 2019. Spatiotemporal Manifold Learning for Human Motions via Long-horizon Modeling. IEEE transactions on visualization and computer graphics.

[4] Chen, W., Wang, H., Yuan, Y., 2020. Dynamic Future Net:Diversified Human Motion Generation. arXiv preprint arXiv:2009.05109v1.

[5] Jang, Y., Kim, G. and Song, Y., 2018. Video Prediction with Appearance and Motion Conditions, Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

[6] Ahmed, W. S. and Karim, A. A., 2021. Human Motion Imagination and Prediction- A Survey, MJPS, vol. 8, no. 2, pp. 30-45.

[7] Martinez, J., Black, M. J. and Romero, J., 2017. on human motion prediction using recurrent neural network, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), arXiv preprint arXiv:1705.02445, pp. 2891-2900.

[8] Khalaf, Mohammed, Abir Jaafar Hussain, Robert Keight, Dhiya Al-Jumeily, Russell Keenan, Carl Chalmers, Paul Fergus, Wafaa Salih, Dhafar Hamed Abd, Ibrahim Olatunji Idowu, 2017. Recurrent neural network architectures for analysing biomedical data sets, in IEEE Conference on Developments in eSystems Engineering (DeSE), pp. 232-237. IEEE, 2017.

[9] Fragkiadaki, K., Levine, S., Felsen, P., and Malik, J., 2015. Recurrent Network Models for Human Dynamics, In Proceedings of the IEEE International Conference on Computer Vision, pp. 4346-4354.

[10] Ghosh, P., Song, J., Aksan, E., and Hilliges, O., 2017. Learning Human Motion Models for Long-term Predictions, In 3D Vision (3DV), International Conference on IEEE.

[11] Villegas, R., Yang, J., Hong, S., Lin X , and Lee, H. 2017. Decomposing Motion and Content for Natural Video Sequence Prediction, in ICLR (2017), pp. 1-22, 2017.

[12] Li, C.. Zhang, Z.,. Sun, W ., Gim, L, and H. Lee, 2018. Convolutional Sequence to Sequence Model for Human Dynamics, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, , pp. 5226-5234.

[13] Li, Y., Fang, C., Yang, J., Wang, Z., X. Lu and M. Yang, 2018, Flow-Grounded Spatial Temporal Video Prediction from Still Images, ECCV, Springer, pp. 1-16.

[14] Augello, A., Cipolla, E., Infantino, I., Manfre, A., Pilato, G. and Vella, F., 2017. Creative robot dance with variational

encoder. arXiv preprint arXiv:1707.01489.
[15] K. Xu, G. Li, H. Xu, W. Zhang and Q. Huang, 2018. Edge Guided Generation Network for Video Prediction, IEEE.
[16] Ahmed, W.S., 2020, November. Motion Classification Using CNN Based on Image Difference. In 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA) (pp. 1-6). IEEE.
[17] Krizhevsky, A., Sutskever, I., and Hinton, G. E., 2012. ImageNet classification with deep convolutional neural networks, International Conference on Neural Information Processing Systems. Curran Associates Inc. pp. 1097-1105.
[18] Karpathy, A., Toderici, G., and Shetty, S., 2014. Large-Scale Video Classification with Convolutional Neural Networks, Computer Vision and Pattern Recognition. IEEE, pp. 1725-1732.
[19] Vit´anyi, P., and Li, M., 1997. On prediction by data compression. In European Conference on Machine Learning, 14–30. Springer.
[20] Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.
[21] Irhum Shafkat. Intuitively Understanding Variational Autoencoders. URL: https : / / towardsdatascience . com / intuitively understanding-variational-autoencoders-1bfe67eb5daf.
[22] Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Networks, 61, 85-117.
[23] Crnkovic-Friis, L. and Crnkovic-Friis, L., 2016. Generative choreography using deep learning. arXiv preprint arXiv:1605.06921.
[24] Wu, M., 2019. Sequential Images Prediction Using Convolutional LSTM with Application in Precipitation Nowcasting (Master's thesis, Science).
[25] Bishop, C. M., 1994. Mixture density networks.
[26] Graves, A., 2013. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.
[27] https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73].
[28] http://www.nada.kth.se/cvap/actions/
[29] Ahmed, W.S., 2020, April. The impact of filter size and number of filters on classification accuracy in cnn. In 2020 International conference on computer science and software engineering (CSASE) (pp. 88-93). IEEE.
[30] Choi, E., Schuetz, A., Stewart, W.F. and Sun, J., 2017. Using recurrent neural network models for early detection of heart failure onset. Journal of the American Medical Informatics Association, 24(2), pp.361-370.
[31] http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html