2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)
29 November - 1 December 2019, Dhaka, Bangladesh

Beat Based Realistic Dance Video Generation using Deep Learning

Md Shazid Islam, Md Saydur Rahman, M Ashraful Amin
Computer Vision and Cybernetics Group, CSE, Independent University, Bangladesh
araf.shazid@gmail.com, saydur.tonmoy@gmail.com, aminmdashraful@iub.edu.bd

***Abstract*—Deep learning based feature extraction has enabled us to synchronize audio and body movements. It is a promising research field which has great applications in generating sign language, computer animations as well as dance. Previously, computer generated choreography was limited to just stick figure representation. This paper adds image translation technique in dance generation which produces realistic dance moves. With this technique it is possible to produce dance video of a amateur person dancing like a professional. The mapping of stick figure to realistic image is done using Generative Adversarial Network (GAN). We created our own dataset and after adversarial training reconstructed images have SSIM mean 0.864 and LPIPS mean 0.0168. This method produces realistic dance video which is beat based. Body movement speed varies according to the tempo of music which makes it more relevant to real life dance movement.**

***Index Terms*—Deep Learning, GAN, Image translation, Beat,OpenPose, Pix2pixHD.**

I. Introduction

Series of activities establishing relationship between human body movement and music is known as choreography. In previous works we saw researchers generated 2D or 3D stick figure representation of dance from music. Our work adds a new dimension to it by generating realistic video of any person dancing.

Choreography generator using artificial intelligence can have a great role in entertainment and dance learning field. In addition, it can be used in animated video games as well as robotics. Furthermore, realistic video of person dancing will enable us to replace background dancers in movies. As our work synchronizes audio with body movements, similar method can have application in martial arts and sign language generator.

Deep learning has made image to image translation possible using conditional adversarial networks. Let assume A and B are related images. Image to image translation means converting A image to B image. Applications like creating sketch of object, black and white to color image conversion , pose transfer between two persons can be done using it.

Our method focuses on beat sensing action generation. We generate stick figure movements from music. Later we perform stick figure to real image translation using conditional adversarial network. We analyzed the reconstructed image quality with different established image quality evaluation methods.

II. Background Study

Before using artificial neural networks, researchers introduced an idea of "Motion graph" which can generate sequential motions . In this method music features like chord,amplitude,beat changes are extracted from music. Shiratori *et al.* [6] focused on poses from dance sequences and links them using suitable transitions. Wang *et al.* [7] proposed a novel model named non-parametric hidden Markov model of human motion (NPHHMM). This model was trained on motion capture data which contained ballet roll,walk etc. Employing Viterbi algorithm and Hidden Markov Model, Ofli *et al.* [8] proposed a model which could perform many-to-many statistical mappings.It was used to map from audio features to dance movement.

Deep learning based Chor-RNN [1] uses mixture density LSTM for generating novel choreography with certain style. In addition to generating sequences of movement, it has compositional cohesion,. Yalta *et al.* [2] proposed a deep recurrent neural network which is the combination of convolutional neural network (CNN) and long short-term memory (LSTM).This part extracts musical features.Then it uses LSTM layers as decoder part to generate rhythmic actions. GrooveNet [3] uses factored conditional restricted boltzmann machines (FCRBM) and RNN to produce movements . Tang *et al.* [4] proposes a LSTM-autoencoder based dance generator and provided a dataset of four types Chinese dance . Lee *et al.* [5] proposed a method which uses Mel-Frequency Cepstrum Coefficients (MFCC) from audio and coordinates from body parts from video. Using an auto-regressive encoder-decoder network on these features it generates 2D dancing skeleton.

Image to image translation has reached a new level due to Generative Adversarial Networks (GANs). CoGAN, Cycle-GAN,Pix2pix [18], Pix2pixHD [19], are well known frameworks for image to image mapping. Generating images with sharp details and high quality is possible through conditional GANs. Sketching, creating neural cities, live drawing interfaces are remarkable applications of Pix2pix. Chan et al. [17] used Pix2pixHD framework for pose transfer between two persons.

III. Method Overview

Our main target is producing video where a person is dancing with music. So there are two parts- audio analysis and video analysis. Audio analysis comprises beat detection and repeated patterns extraction from a music. Video analysis

978-1-7281-5852-5/19/$31.00 ©2019 IEEE

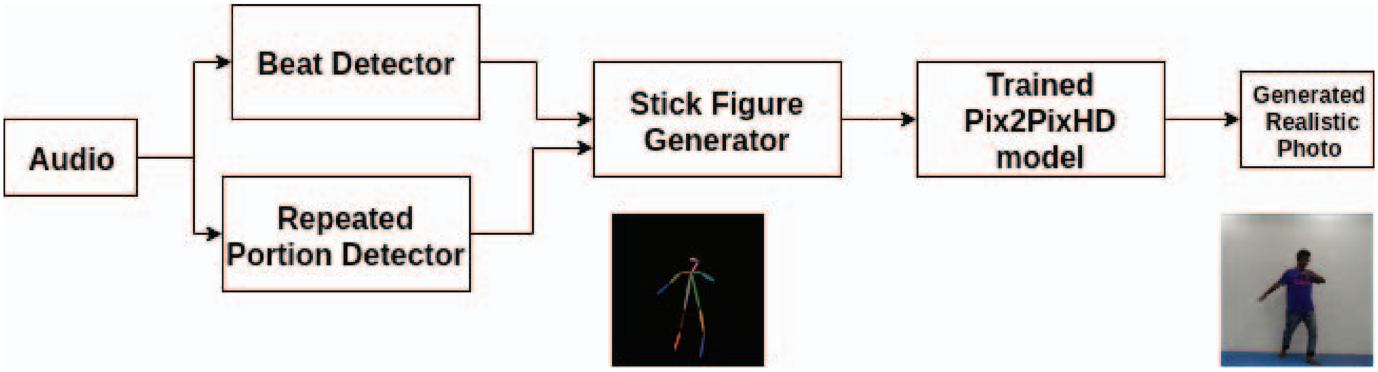

Fig. 1. **Method Overview:** Consists of audio processing and video processing. Audio processing occupies beat detection and repeated audio portion extraction. Video analysis part generates stick figures from audio analysis and creates realistic fake images through Pix2pixHD training.

comprises generating mathematical models of a person dancing and convert in into realistic images of target person. In Fig. 1 the overview is demonstrated.

IV. Audio Analysis

A. Beat Detection

For beat detection we use the librosa [11] library. This is based on dynamic programming [12]. Firstly, onset strength measured. From onset correlation, tempo is determined. Then peaks of onset strength are detected which are consistent with the estimated tempo. We assign one action between two beats.

B. Repeated pattern of Music Extraction

In most of the music we observe a repeated pattern. To extract repeated pattern, chorus section detection method [13] is used.The algorithm finds out the notes played. It searches for repeated portions by comparing short sections. Then the longest portions with decent interval is chosen as final repeated section.

From fourier transformation of the audio we find out what notes are playing. Representation of this information is called chromagram. There are 12 notes in western scale. Using one hot encoding we encoded the notes in every 0.2s interval. For a 3 min long audio it gives $3 \times 60/0.2 = 900$ frames of audio. So the encoded chromagram is 900×12 matrix.

If V_1 and V_2 are note vectors at any two instances in the song, similarity function is defined as:

$$f(V_1, V_2) = 1 - \frac{\|V_1 - V_2\|}{\sqrt{12}} \tag{1}$$

If two vectors are of same note we get 1, otherwise the value is less than 1 .

The next step is analyzing these vectors in time-time similarity matrix and time-lag similarity matrix.Time-time similarity matrix M is defined as

$$M[x][y] = f(x, y) \tag{2}$$

If we use color map , repeated sections are diagonal dark lines in the time-time similarity matrix. Time-lag similarity Matrix T is defined by

$$T[x][y] = M[x][x-y] = f(x, x-y) \tag{3}$$

Fig. 2. Samples from dataset.

In color map, repeated sections are horizontal dark lines in the time-lag similarity matrix. We shall use time-lag similarity matrix method because it is easier to extract a horizontal line than a diagonal line in color map. To separate those horizontal lines we use denoising and smoothing techniques.

V. Video Analysis

A. Dataset

We capture a video of about 14 minutes of our target person with free movements. Then we convert it into images at sampling rate 25 fps. So we total had 21000 images. We make sure in the video background and intensity of light remain unchanged. Samples from dataset are given in Fig. 2.

B. Pose Estimation

For pose estimation OpenPose [9] has been used. Its COCO pretrained model can detect 18 key points of body. Then connecting those points we generated stick figures.

C. Training

GAN network called Pix2pixHD [19] has been used for training . To explain this lets look at Fig. 3. P is a pose detector which is mentioned in the previous sub section. It maps a real person's image y into stick figure x. Generator G aims to synthesize realistic image of target person from the stick figure by participating in a minimax game against discriminators $D = (D1, D2, D3)$

The Discriminator D's goal is to recognize if $G(x)$ is real (belongs to the original dataset) or if it is fake (generated by forgery). $G(x)$ is optimized using discriminator and reconstruction loss, "dist" which can be obtained using VGGNet [10].

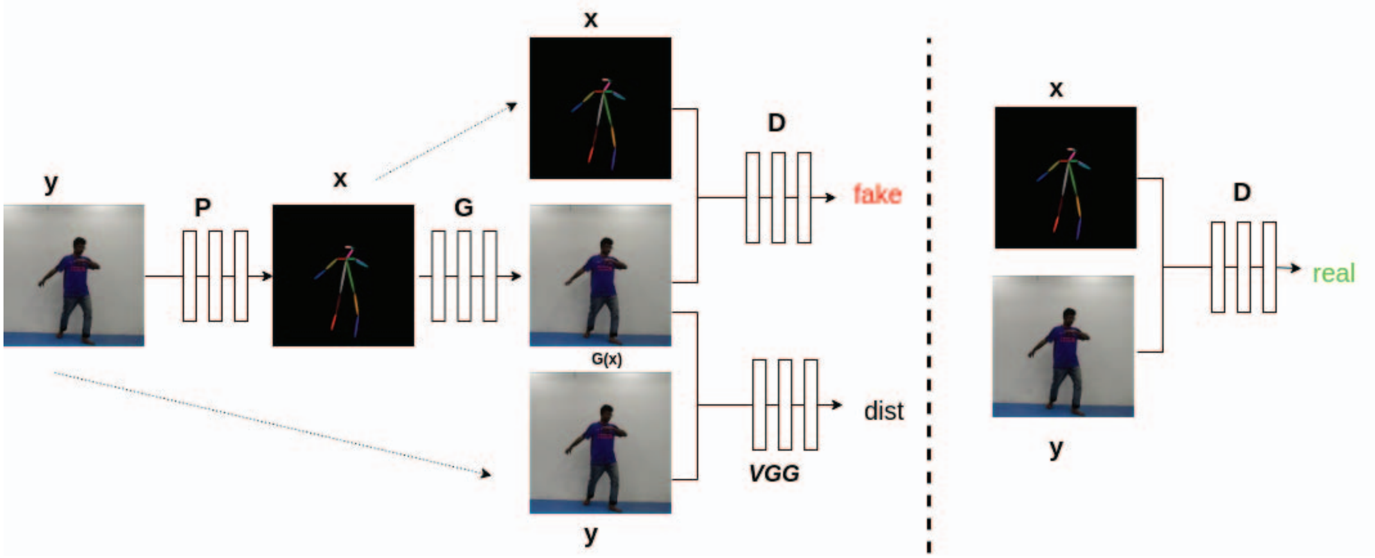

Fig. 3. **Pix2pixHD training:** Pose detector P detects body key-points from images extracted from videos. At the time of training generator G learns the mapping from pose stick figure to realistic image translation. Discriminator D learns to distinguish between the real pair (x, y) and the fake pair (G(x), y). Thus G and D are simultaneously improved.

Generator and Discriminator are trained simultaneously and gradually improves which means generator produces more realistic image to deceive discriminator and discriminators learns nuance differences between generator output and original image. The total loss function becomes

$$\mathcal{L}_{total} = \min_{G}\left[\left(\max_{D_1,D_2,D_3}\sum_{k=1,2,3}\mathcal{L}_{\text{GAN}}(G,D_k)\right) + \lambda_{FM}\sum_{k=1,2,3}\mathcal{L}_{\text{FM}}(G,D_k) + \lambda_{VGG}\mathcal{L}_{VGG}(G(x),y)\right] \tag{4}$$

Here, $\mathcal{L}_{GAN}(G,D)$ is adversarial loss which is described in Pix2pix paper [18].

$$\begin{aligned}\mathcal{L}_{GAN}(G,D) = E_{(x,y)}[\log(D(x,y))] \\ +E_x[\log(1-D(x,G(x))]\end{aligned} \tag{5}$$

$\mathcal{L}_{FM}(G,D)$ is the discriminator feature-matching loss, and $\mathcal{L}_{VGG}(G(x),y)$ is the perceptual reconstruction loss.

D. Stick Figure to Image Translation

With the help of professional dancers, we design 30 basic dance actions mathematically using coordinate geometry. Two dimensional coordinates of different key points of body create those actions in stick figure form. These actions can be speed up or slowed down according to the time allocated to it. Basic dance actions for a segment of music are selected based on amplitude, beat interval, frequency of the music. The actions will be changed after each beat. For transition from one action to another we use linear interpolation of body key points. If there is repetition of same music, the dance actions will be same at those parts. Then we concatenate the basic actions selected for the music and get the full choreography by stick figure. After that we shall use our trained generator model discussed in previous sub section to convert the stick figures into real life images as shown in Fig. 4. Creating video from the images will generate our final output.

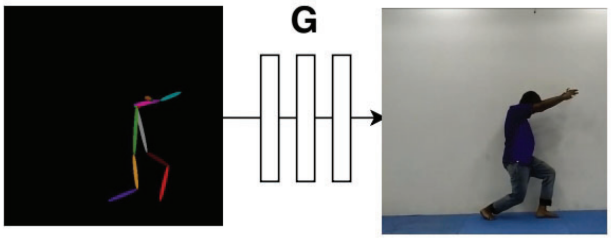

Fig. 4. Stick figure to Realistic Image Translation.

VI. Result

We shall analyze results for both audio and video. In determining audio repeated portion we use time-lag similarity matrix. In Fig. 5 color-maps of (a) time-lag similarity matrix and (b) denoised time-lag matrix are demonstrated. The dark part of x axis indicates repeated segment and corresponding y axis value indicates the time lag value. For example, we have a dark line from 2:05-2:15 at a y coordinate of 30s which means 2:05-2:15 segment is repeated 30s back at 1.35-1.45.

In Fig. 6 we show a sequence of generated images of dance from stick figure. We analyze the quality of reconstructed images based on Mean Square Error(MSE), Structural Similarity (SSIM) [14], Multi-scale Structral Similarity (MSSSIM) [20] and Learned Perceptual Image Patch Similarity (LPIPS) [15].

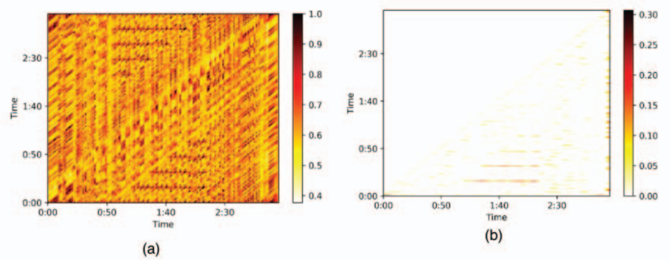

Fig. 5. Repeated pattern of audio extraction. (a) shows time-lag similarity matrix color-map and (b) shows color-map after filtering and repeated pattern of music is shown by horizontal lines.

MSE(Mean Square Error) may be of two types. One is pixel by pixel. Another is MSE of reconstructed key-points. However, MSE sometimes becomes very high value and it is hard to standardize. PSNR [16] is one of the most commonly used metric to measure the quality of reconstructed image defined as

$$PSNR = 20\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{6}$$

where MAX_I is maximum possible value of a pixel. Here it is 255.

On the other hand SSIM looks for similarities within pixels and scales the value between -1 and 1. -1 means images are very different and 1 means they're identical. SSIM can be expressed as

$$\mathrm{SSIM}(x, y)=\frac{\left(2 \mu_x \mu_y+c_1\right)\left(2 \sigma_{xy}+c_2\right)}{\left(\mu_x^2+\mu_y^2+c_1\right)\left(\sigma_x^2+\sigma_y^2+c_2\right)} \tag{7}$$

LPIPS normalizes features in all pixels. Then it scales all features by a feature-specific weight and determines square of L2 distance between weighted activations and average of these squared values are summed over the layers .The image distance metric becomes-

$$d_{\mathrm{LPIPS}}(x, y)=\sum_l \frac{1}{H_l W_l} \sum_{i,j}\left\|w_l \odot\left(\hat{x}_{ij}^l-\hat{y}_{ij}^l\right)\right\|_2^2 \tag{8}$$

$\hat{x}_{ij}^l$,$\hat{y}_{ij}^l$ are normalized feature vectors of pixel; (i, j) at layer l and w_l is the weight matrix in layer l. In TABLE I all image quality analysis metrics with corresponding values are demonstrated.

TABLE I
RECONSTRUCTED IMAGE QUALITY ANALYSIS TABLE.

Metric	Metric value
MSE(pixel by pixel)	11.95
MSE per body keypoint (in $pixel^2$)	7.5
PSNR(dB)	37.358
SSIM mean	0.864
MSSSIM mean	0.887
LPIPS mean	0.0168

VII. CONCLUSION AND FUTURE WORK

Our method can produce dance choreography from different musics. Previously all works of dance choreography were limited to generating stick figures. Our work adds a new dimension by creating realistic video using Pix2pixHD. However, our work has some limitations.Although the output image quality is convincing, they suffer from jitters and shakiness. In addition, our training process is very sensitive to light intensity change and even slight movement of camera. Furthermore, background having too much details may threaten the quality of reconstructed image. Using a larger dataset may solve this problem to a certain level.Our dance actions are mainly beat based. It ignored the vocabulary context in generating actions. In future, we'll pay heed to create meaningful expression on basis of music lyrics. Our work has been designed for 2D action generation. We'll try to make the system compatible for 3D system in future. Moreover, in future we set to develop a model which can learn different styles of dances from music and available dance videos.

REFERENCES

[1] Luka Crnkovic-Friis and Louise Crnkovic-Friis, "Generative choreography using deep learning," arXiv preprint arXiv:1605.06921, 2016.

[2] N. Yalta, S. Watanabe and T. Ogata, "Weakly supervised deep recurrent neural networks for basic dance step generation," arXiv preprint arXiv:1807.01126, 2018.

[3] O. Alemi, J. Françoise and P. Pasquier,"GrooveNet: Real-time music-driven dance movement generation using artificial neural networks," networks, vol. 8, no. 17, pp. 26, 2017.

[4] T. Tang, J. Jia, and H. Mao, "Dance with Melody: An LSTM-autoencoder Approach to Music-oriented Dance Synthesis," in *ACM Multimedia Conference on Multimedia Conference*, pp. 1598-1606, 2018.

[5] J. Lee, S. Kim, and K. Lee, "Listen to dance: Music-driven choreography generation using autoregressive encoder-decoder network," arXiv preprint arXiv:1811.00818, 2018.

[6] Shiratori, Takaaki, Atsushi Nakazawa, and Katsushi Ikeuchi. "Synthesizing dance performance using musical and motion features." Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.. IEEE, 2006.

[7] Yi Wang, Zhi-Qiang Liu, and Li-Zhu Zhou, "Learning hierarchical non-parametric hidden markov model of human motion," in *International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3315-3320, 2005.

[8] F. Ofli, E. Erzin, Y. Yemez, and A M. Tekalp, "Learn2dance: Learning statistical music-to-dance mappings for choreography synthesis," in *IEEE Transactions on Multimedia*, pp. 747-759, 2011.

[9] Z. Cao, T. Simon, S.E. Wei, and Y. Sheikh,"Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291-7299, 2017.

[10] J. Johnson, A. Alahi, L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, pp. 694-711, 2016.

[11] B. McFee, C. Raffel,D. Liang, D.P. Ellis, M. A McVicar, E. Battenberg and O. A Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference,*Vol. 8, 2015.

[12] D.P Ellis,"Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 35, no. 1, pp. 51-60, 2007.

[13] M. Goto, "A chorus section detection method for musical audio signals and its application to a music listening station," *IEEE Transactions on Audio, Speech, and Language Processing*, vol.14,np. 5, pp. 1783-1794, 2006.

[14] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli, "Image quality assessment: from error visibility to structural similarity." in *IEEE transactions on image processing*,vol. 13, no. 4,pp. 600–612, 2004.

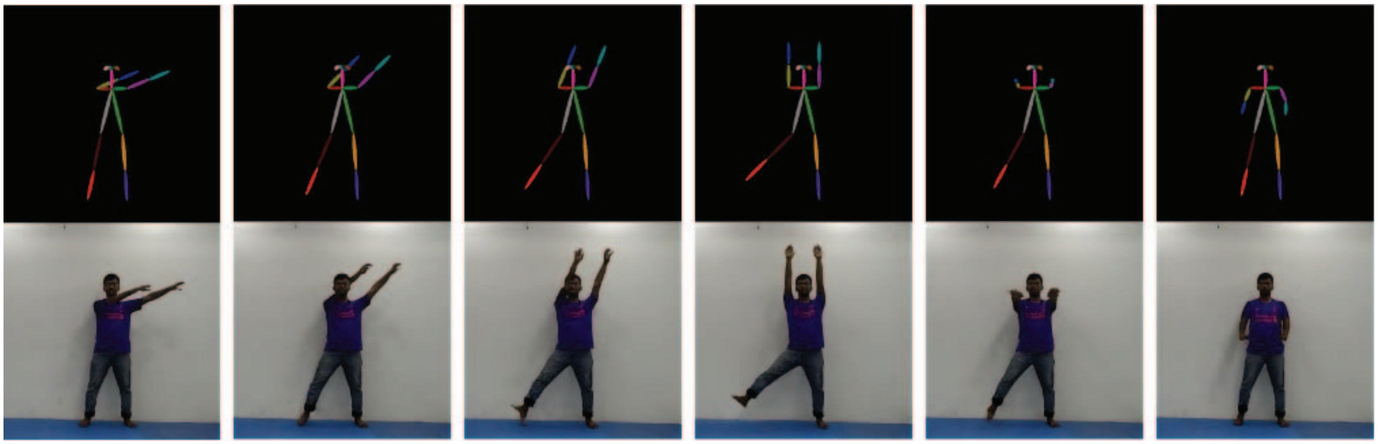

Fig. 6. Sequence of generated dance images: at the top dance stick figures are shown. Below each stick figure corresponding image is shown which is produced by image translation.

[15] Z. Richard, P. Isola, A.A Efros, E. Shechtman and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 586-595, 2018.

[16] Q. Huynh-Thu, and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," in *Electronics letters*, vol. 44, no. 13, pp. 800-801, 2008.

[17] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, "Everybody dance now," in Proceedings of the IEEE International Conference on Computer Vision, pp. 5933–5942, 2019.

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of IEEE conference on computer vision and pattern recognition*, pp. 1125-1134, 2017.

[19] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro, "pix2pixHD: High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8798-8807, 2018.

[20] Nasr, M Abdel-Salam, M.F. AlRahmawy and A.S. Tolba, "Multi-scale structural similarity index for motion detection," in *Journal of King Saud University-Computer and Information Sciences,* vol. 29, no. 3, pp.399-409, 2017.