İskelet Verisine Dayalı İnsan Aktivitesi Tanıma ve Zamanda Aktivite Yeri Belirleme
Human Activity Recognition and Temporal Action Localization Based on Depth Sensor Skeletal Data

Yusuf Erkan Görgülü[1], Kubilay Taşdelen[2]
[1]Bilgisayar Mühendisliği Bölümü, Isparta Uygulamalı Bilimler Üniversitesi, Isparta, Türkiye
{yusufgorgulu}@isparta.edu.tr
[2]Elektrik-Elektronik Mühendisliği Bölümü, Isparta Uygulamalı Bilimler Üniversitesi, Isparta, Türkiye
{kubilaytasdelen}@isparta.edu.tr

***Özetçe* —Atalet ölçüm birimlerinin çoğalması, derinlik algılayıcı sistemlerin yaygınlaşması sonucunda anlamlı hareket verilerinin kolay elde edilebilmesi sonucunda insan aktivitesi tanıma alanında çalışmalar ve bu alanda kullanılan veri kümeleri de artmıştır. Bu çalışmada derinlik algılayıcısı ile elde edilen iskelet verileri kullanılarak yirmi yedi farklı aktivitenin sınıflandırılmasına yönelik bir çalışma yapılmıştır. Bahsedilen iskelet verilerini analiz etmek için dört katmanlı Uzun Kısa Süreli Bellek (UKSB) mimarisi kullanılmıştır. UKSB hücreleri, ilgili verilerin dizileri arasındaki uzun süreli ilişkiyi hatırlayabilen özel bir tekrarlayan sinir ağı türüdür. Ayrıca basit özyinelemeli sinir ağı kullanılarak belirli bir hareketin verilen sekans içerisinde yerini tespit etmeye yönelik bir uygulamaya da yer verilmiştir. Bu mimari dört katmandan oluşmaktadır. Özniteliklerin otomatik olarak bulunmasını sağlayan derin öğrenme yapıları sayesinde %93 gibi yüksek test sınıflandırma başarısı elde edilmiştir.**

Anahtar Kelimeler—Derin öğrenme,İnsan aktivitesi tanıma,RNN, LSTM

***Abstract*—Abundance in inertia measurement units and wide usage of depth sensors has led to the effortless acquisition of significant activity data. As a result, studies in the area of human activity recognition and datasets related to this area have risen. In this paper, skeletonized action sequence data; obtained by a depth sensor, is used to classify among twenty seven different class of activities. A three-layer Long-Short Term Memory(LSTM) architecture was used to analyze the mentioned skeleton data. LSTM cells are a special type of recurrent neural network which can remember long time relationship between sequences of related data. Also, another application was carried out using a recurrent neural network(RNN) architecture to address where a specific action occurs on a video stream. This architecture consists of four layers. Thanks to this deep learning structures; which led to the automated extraction of features, high classification accuracy rates around %93 can be achieved.**

Keywords—Deep learning,Human activity recognition, RNN, LSTM

978-1-7281-9136-2/20/$31.00 ©2020 IEEE

I. INTRODUCTION

Thanks to the exponentially increasing developments in the field of machine learning, which is a sub-branch of artificial intelligence, studies such as object recognition, segmentation, and classification on pictures or videos have been carried out rapidly and with great accuracy, and studies for the recognition of human activities have also increased.

Presence of human element in activities, the recognition of near-human objects, and extraction of such small details, combining these elements using sub-processes with complex systems is a time consuming and difficult task. In addition, using temporal information such as which activity occurred when makes this task more exhaustive.

When studies in this field are analyzed, it is seen that applications for convolutional neural networks (CNN) outshine among other applications using other structures [1] [5] [6]. In addition to these, some studies involving the use of CNNs are conducted by converting the sensor data from inertial measurement units to Short Time Fourier Transform(STFT) based image data rather than using human activity image sequences. [7] [8]. STFT is a method used to analyze how the magnitude and phase content of a signal; windowed over a period of smaller sections, changes over time. Recursive neural networks (RNN) stand out in the analysis of time-dependent systems. LSTM [2] stores data for a long time and gives better results compared to the RNN models.

In this paper, the classification of twenty-seven different human activities, using an LSTM model, has been studied. Besides, another model to find where a specific action happens through a sequence of skeletal frames has been studied. In this second model basic RNN model was used.

The dataset being used includes skeleton data, depth data, inertial measurement data, and RGB data. It has a rich data range in this respect. In this study, among all data modalities, only the skeletal data was used.

LSTM model used for classification purposes yielded an accuracy result of %93.49 for the test set and whole dataset accuracy for the classification was calculated as %98.03. The basic RNN model used for temporal action localization yielded

%93.45 accuracy for the test-set and for the whole dataset a %99.19 accuracy has been achieved.

The methods used, information about the dataset, pre-processing of the data, and implementation of deep learning models are included in section 2, and experimental comments are given in section 3.

II. METHODS

In this study, the classification accuracy analysis of human actions was performed using an LSTM network architecture. In addition, another network architecture consisting of simple RNN cells was designed for detecting whereabouts of a specific human action from a sequence of actions. Both networks are implemented using KERAS with Tensorflow backend.

A. Data Set

The data set used in this study; Multimodal Human Action Dataset is obtained from the University of Texas-Dallas [4]. In other words, the human movements data set obtained using multiple sensor elements. The sensors used in this data set are the depth sensor of Microsoft Kinect V1 and a wearable inertial sensor. In the data set, twenty-seven different movements were recorded four times by eight different people. Some examples of these movements are *'right hand wave'*, *'draw x'*,*'draw circle'*, *'draw triangle'*, *'jogging'*, *'sit to stand'* and *'stand to sit'*. Since three of these records are corrupt, there are eight hundred and sixty-one records left. There are four different inputs for data acquisition, these are RGB(Red-Green-Blue) videos, depth videos, skeleton joint points, and three-axis acceleration and three-axis rotation data from the inertial sensor.

Subjects of the data-set perform given actions alone. They are aligned, facing directly in front of the camera. In Figure 1, a frame of a sample is given. In this sample subject is performing the action thirteen or as stated by the authors 'boxing'. Figure 1, (a) is a depth frame, (b) is an RGB frame, and (c) is the twenty point skeleton frame data depicting the subjects standing position.

B. Preprocessing of Skeletal Data

Skeleton data consists of twenty points. Since the points are in the cartesian coordinate system, each point is represented by a three-element vector. When the skeleton sequence of each sample was examined, it was observed that the highest number of frames was 125. Each sequence was subjected to z-score normalization on a frame basis. In cases where the sample sequences have a shorter length, the sequences are filled with zeros so that the length becomes constant for the entire dataset. In an attempt to rather than filling with zeros, remaining sequences were interpolated by using the closest two vectors and fed to the classifier model. LSTM only accepts two dimensional input so the input fed into the system need to be organized as number of frames and 20 points with their corresponding 3 dimensional vectors (a 125x20x3 tensor) which is reshaped as a 125x60 tensor. Thus, each sample becomes a 125 by 60 matrix input to the system.

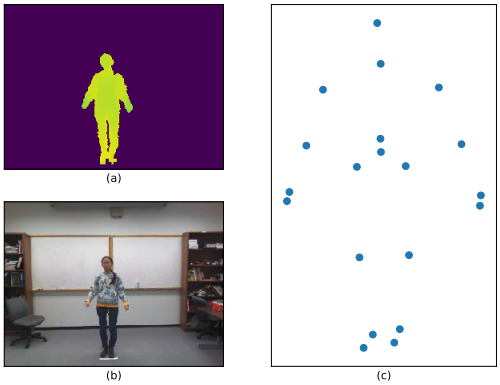

Figure 1: Frame shots of a sample performing action 'boxing', (a) Depth frame, (b) RGB frame and (c) skeleton frame showing joints (20 points).

C. Z-Score Normalization

Each sample consists of one hundred and twenty-five frames. Each frame is twenty by three matrix. Z-score normalization is applied with the formula given in equation (1) for each sample, using its own mean and standard deviation for each frame. x_i represents i^{th} sample μ_{x_i} represents average for this sample and σ_{x_i} represents the standard deviation for this sample.

$$\text{z-score}(x_i) = \frac{x_i - \mu_{x_i}}{\sigma_{x_i}} \tag{1}$$

D. LSTM Network Architecture

The model prepared for the data-set is a 3-layer model. The model consists of the input layer,the first LSTM layer and the second LSTM layer. The first LSTM layer consists of 90 units, kernel initializer is chosen as *Lecun normal* and bias initializer as *Lecun uniform*. The second LSTM layer uses softmax activation to classify among twenty-seven classes. The network structure designed is given in Figure 2. After the raw data is passed through the Z-Score normalization it is given to the first layer then it is processed and classified as one out of twenty-seven classes. To guarantee the network converges safely to global minima and lessen the effects of over fitting a batch normalization layer is added between the first and the second LSTM layers. Adam optimizer with KERAS' standard parameters is used except the learning rate was chosen as 10^{-5}.

E. RNN Architecture

The other model used for this data-set is a four-layer model. An input layer, first RNN layer, second RNN layer, Time-distributed layer, and final binary classification output layer. In this model input is a combination of four actions in which one of them is the desired action to be detected, distributed among the other three. In Figure 3 the structure of the RNN architecture is given. Like the previous one, the raw data is normalized with Z-Score then it is fed into the consecutive

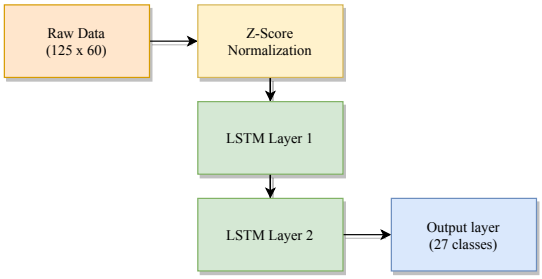

Figure 2: LSTM architecture layers

layers. The output classification is carried out as a binary classification, as if there exists the desired action or not.

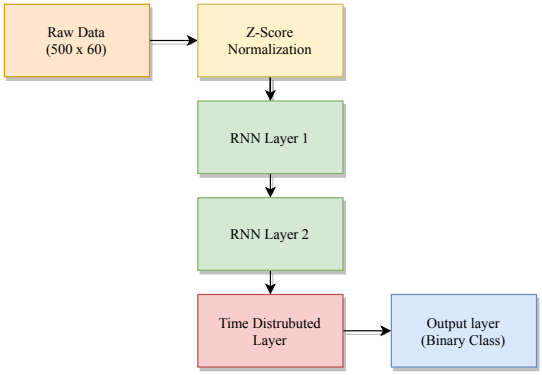

Figure 3: RNN architecture layers

III. RESULTS

Training of LSTM network architecture is accomplished using 600 samples out of 861 samples, which is approximately %70 of total samples. During training, ten-times repeated ten-fold cross-validation method for hundred epochs was used. The batch size was chosen as thirty samples. The remaining 261 samples were used as test-set. The average classification accuracy plot of training is given in Figure 4. Also, average validation classification accuracy plot is given in Figure 4. Both figures are given with their standard deviation added and subtracted from the mean accuracy. Red line shows the mean accuracy, blue line shows the upper bound and the green line shows the lower bound of the standard deviation envelope.

The classification accuracy for the test-set was calculated as %93.49. The whole data-set classification accuracy was calculated as %98.03. In Figure 5 confusion matrix for the predicted labels and real labels is given. Out of 27 classes, 6 classes were predicted with some errors. The remaining 21 classes were predicted as is. As it is shown in the confusion matrix, the third action was the worst predicted of all. Eight out of thirty-two samples were falsely predicted.

For the second model (RNN), an action was chosen as the action of interest (Aoi) to be detected. Remaining actions and Aoi were grouped together to create sequences of action arrays with a length of four actions per array. For training and validation purposes, 32 samples including an Aoi (positive sample) and 100 samples free of any Aoi (negative sample)

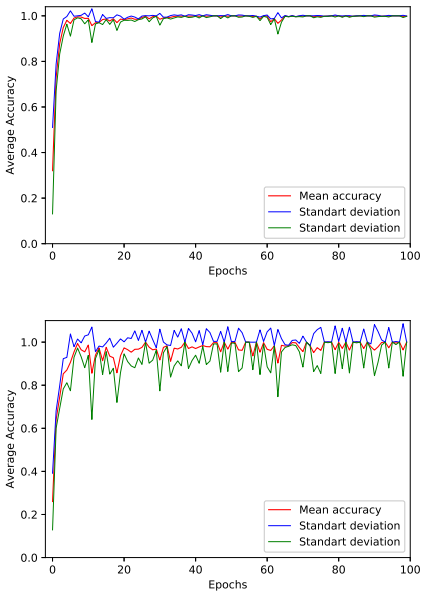

Figure 4: Average training accuracy plot (above), average validation accuracy plot of LSTM model (below).

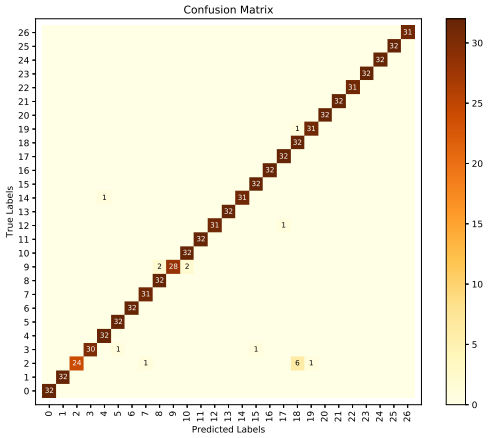

Figure 5: Confusion Matrix for LSTM model

were assembled together. Since there are only positive and negative samples, the classification goal becomes a binary classification problem. Among these samples 25 positives and 95 negative samples were spared for training, the remaining 12 were used as the test set. A total of 120 samples were cross-validated as ten times five-fold for fifty epochs. For the test accuracy, a rate of %93.25 was achieved. The whole data-set (132 samples) classification accuracy was calculated as %99.19. The average classification accuracy plot of training and validation classification accuracy plot is given in Figure 6.

RNN model consists of four layers which are given as, input layer, two consecutive RNN layers, and a time-distributed layer. A sample sequence consists of 125 frames, so in this

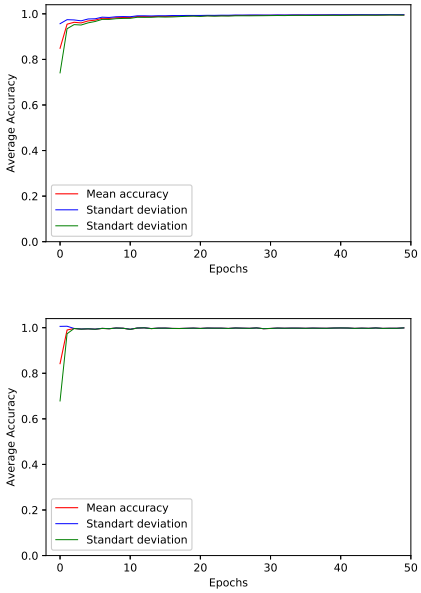

Figure 6: Average training accuracy plot (above), average validation accuracy plot of RNN model (below).

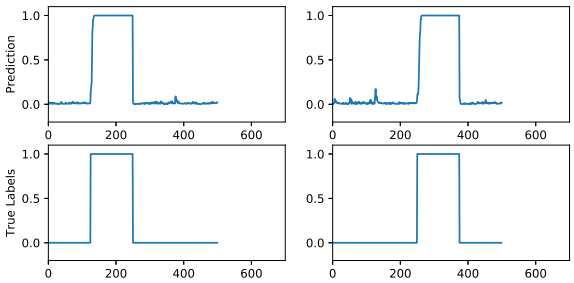

Figure 7: Two true predicted sample outputs of training set. First row shows predicted labels. Second row shows true labels.

model, one sample consists of 500 frames. The label of the sample consists of ones during the Aoi and zeros elsewhere. In Figure 7 true labels and non-thresholded prediction values are given for the training set and in Figure 8 the same values are given for the test set.

This model makes the classification based on per frame. In other words, it labels the frame as zero or one, if it belongs to a positive sample or not. Action *'jogging'* was chosen as the positive action. As Figure 8 indicates, 9 out of 12 samples are detected with success. In fact, in the first three samples, negative samples were also detected successfully. But from frames 125 to 250 for these samples, the model could not classify the true label. Instead, we can see some peaks during the action of interest. But these peaks also exist with a lower score at the beginning of the actions. If Figure 8 predictions(left) are further examined, those slight peaks occurs approximately at frames 125, 250 and 375. This may happen since every action starts in the same manner, like raising an arm from a starting position to some other location.

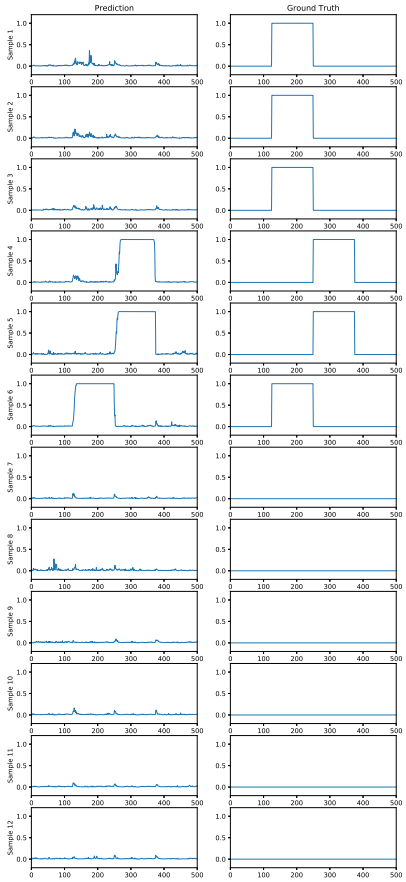

Figure 8: Twelve predicted sample outputs of test set. From above, first six sequences include positive samples, remaining six are negative samples.

IV. CONCLUSION

A two-layer LSTM based architecture yielded the classification accuracy result of %93.49 for the test-set, which may be considered as a high accuracy rate. This may suggest a fast and reliable classification system can be implemented only using skeletal data. When confusion matrix results are scrutinized; focusing on the worst-case scenario, the action *'right hand wave'* was confused with actions *'draw x'* or action no.7, *'hand catch'* or action no.19 and mostly *'knock on door'* or action no.18 as labels indicate. This confusion may suggest actions that occur with similar body members can be confused more often or the performer may lack the perceptual information of how the action happens. To minimize this confusion, performers of the action may be trained for how the actions will be performed. However, this may raise the budget of the project and consume more time for the data-set

preparation.

To test how the implemented LSTM system performs, we compiled a program and imitated the actions of the mentioned data-set. Some actions are well recognized by the program like *'sit to stand'*, *'stand to sit'* and *'front boxing'*. Yet, for the actions that include arms, distinction drops and action can be confused more easily. For the second model(RNN) future studies will include other complex structures of recurrent networks like LSTM or gated rectified units.

REFERENCES

[1] M. Yalçın, N. Tüfek and H. Yalcin, "Activity recognition of interacting people," 2018 26th Signal Processing and Communications Applications Conference (SIU), Izmir, 2018.

[2] S.Hochreiter and J. Schmidhuber. "Long short-term memory." Neural computation, 9(8), 1997.

[3] D. Bahdanau, C. K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473, 2014.

[4] Chen, Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor." ICIP, IEEE, 2015.

[5] Z. Cao,, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh,"OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields." arXiv preprint arXiv:1812.08008, 2018.

[6] O. Köpüklü, X. Wei, and G. Rigoll, You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Action Localization. arXiv preprint arXiv:1911.06644, 2019.

[7] S.M. Lee, S. M. Yoon and H. Cho, "Human activity recognition from accelerometer data using Convolutional Neural Network." 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), 2017.

[8] A. Bevilacqua,K. MacDonald, A. Rangarej, V. Widjaya, B. Caulfield and T. Kechadi, "Human activity recognition with convolutional neural networks. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases", Springer, Cham, 2018, pp. 541-552.