Fast human activity recognition

Shane Reid, Sonya Coleman, Dermot Kerr, Philip Vance, Siobhan O'Neill
[1]*School of Computing, Engineering and Intelligent Systems, Ulster University Magee Campus, Derry/Londonderry, Northern Ireland*
[2]*School of Psychology, Ulster University Coleraine Campus, Coleraine, Northern Ireland*
{Reid-S22, sa.coleman, d.kerr, p.vance, sm.oneill}@ulster.ac.uk

Keywords: Social signal processing, Activity recognition, MLP, key points, feature extraction.

Abstract: Human activity recognition has been an open problem in computer vision for almost two decades. In that time there have been many approaches proposed to solve this problem, but very few have managed to solve it in a way that is sufficiently computationally efficient for real time applications. Recently this has changed, with keypoint based methods demonstrating a high degree of accuracy with low computational cost. These approaches take a given image and return a set of joint locations for each individual within an image. In order to achieve real time performance, a sparse representation of these features over a given time frame is required for classification. Previous methods have achieved this by using a reduced number of keypoints, but this approach gives a less robust representation of the individual's body pose and may limit the types of activity that can be detected. We present a novel method for reducing the size of the feature set, by calculating the Euclidian distance and the direction of keypoint changes across a number of frames. This allows for a meaningful representation of the individuals movements over time. We show that this method achieves accuracy on par with current state of the art methods, while demonstrating real time performance.

1. INTRODUCTION

Human activity recognition, defined as the challenge of classifying an individual's activity from a video, is one of the oldest problems in the field of video processing, having been studied for almost two decades. In that time there has been a number of proposed approaches to solving this problem, with the majority based on either spatio-temporal features (Dollar et al., 2005; Laptev, 2004; Zelnik-Manor & Irani, 2001), optical flow (Efros et al., 2003; Guo et al., 2010; Ke et al., 2005; Schüldt et al., 2004; Wang et al., 2011) or deep learning (D'Sa & Prasad, 2019; Lee & Lee, 2019; Sheeba & Murugan, 2019; Subedar et al., 2019). These methods have been shown to achieve high accuracy on common benchmark datasets but come with a significant computational cost. As such, their use for real time applications is limited.

Feature extraction is an approach to reduce computational cost in image and video processing, for example, by compressing an image into a sparse set of interest points (Camarena et al., 2019). Early attempts to do this used general interest point detectors such as SIFT and SURF. However, these methods had a number of drawbacks, most notably that there was no agreed standard for human representation (Sun et al., 2010). To solve these problems, specialized "key point" detectors were developed, which can be applied to an image and a set of locations of key body joints for each individual within the image is returned. Two of the most popular approaches are OpenPose (Cao et al., 2017) and AlphaPose (Xiu et al., 2018).

Recently, (Camarena et al., 2019) presented an approach for fast human activity recognition, based on the method used in (Wang et al., 2013). In order to speed up this approach, they used a reduced feature set of six keypoints (those for the neck, right wrist, left elbow, left wrist, mid hip and left ankle), generated using OpenPose (Cao et al., 2017). In doing so they reduced the number of features used by approximately a factor of 5 and achieved an approximate 8 times improvement in speed over the original method (Wang et al., 2013), with a reduction in accuracy of only 1.4%. This enabled the approach to run sufficiently fast for real time classification, a breakthrough for human activity recognition. In order to achieve this speed gain, their approach only sampled a small number of body keypoints. However, by doing this, they have a less generalizable representation of the individuals body pose; this may limit the type of activity that can be detected. For example, in a situation where it is necessary to detect whether an individual is kicking with their right leg, this approach would struggle as they have extracted no keypoints relating to the right leg. In contexts where it is necessary to detect a large range of different actions, using a reduced set of keypoints may not be feasible.

Recently the work of (Reid et al., 2020) showed that by reducing the framerate and sample size used for keypoint based activity recognition, the computational cost can be reduced enough to perform real time activity recognition on upwards of 14 individuals simultaneously. However, this approach also comes with downsides, the most obvious of which is that by reducing the sample rate in this way, it may be difficult to detect actions which are characterized by rapid movements, such as clapping, where the movement may be completed between frames being sampled. Earlier methods for overcoming this issue using traditional keypoints involved measuring keypoint trajectories, but these approaches are limited by the fact that they are unable to track specific landmarks (e.g elbows, hands etc.) (Matikainen et al., 2009). Later improvements to such methods

achieved impressive accuracies on a number of benchmark datasets but were still hampered by poor run-time performance (Jain et al., 2013). Due the recent breakthroughs in the area of human landmark detection, keypoint trajectories are once again coming into focus as a viable method for human action recognition (Choutas et al., n.d.)(Yi & Wang, 2018).

In this paper we present a keypoint trajectories based approach that builds on the approach of (Reid et al., 2020), where the set of key points for an individual, extracted over a given time period, are converted into a feature set of "keypoint changes". These keypoint changes encode a history of the Euclidian distance and the direction of keypoint movement, measured over time. We measure the keypoint changes using a reduced sample rate and reduced sample size, but we also measure the short term keypoint changes between concurrent frames. In this way we still maintain a sparse approach of (Reid et al., 2020) but are also able to detect actions which are characterised by rapid movements.

The remainder of the paper is organized as follows. In Section 2 we outline the proposed approach, and the experimental design. In Section 3 we present the performance evaluation results and discussion. In Section 4 we compare the results with other state-of-the-art methods. Finally, in Section 5 we conclude the paper and discuss possible future work.

2. METHODOLOGY

This section will describe the proposed keypoint based approach for fast human activity recognition based on the history of keypoint changes over time in terms of the Euclidian distance and direction. We use OpenPose for keypoint extraction (Cao et al., 2017) as it provides a high level of accuracy with very low computational cost that remains constant when more individuals are detected, unlike with other methods such as AlphaPose (Xiu et al., 2018).

For each individual within an image OpenPose extracts a set of 25 body keypoints. This method works by first using a feedforward neural network to predict a set of 2D confidence maps of body part locations and a set of 2D vector fields of part affinity fields (PAFs) which encode the degree of association between parts. Then these confidence maps and the PAFs are parsed by a greedy inference method to output the 2D keypoints for all people in the image. For more details on the model architecture please see (Cao et al., 2017).

It is worth noting, however, that the novel contributions of this paper are not reliant on any specific keypoint estimation approach and can be implemented with any methods, such as AlphaPose (Xiu et al., 2018), Megvii (Cai et al., 2019), or similar techniques. Regardless of the method used for keypoint extraction, each keypoint is defined as:

$$k_i = \{x_i, y_i\} \tag{1}$$

where x_i and y_i are the image coordinates of the extracted keypoint. We define the Euclidian distance between two keypoints k_i and k_j as:

$$\Delta(k_i, k_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{2}$$

and the angle between them as:

$$\theta(k_i, k_j) = atan2\left(\frac{y_i - y_j}{x_i - x_j}\right) \tag{3}$$

where atan2 is the function which returns the unambiguous angle θ between the two keypoints on the Euclidian plane. We can then define the keypoint change between these two keypoints as:

$$c(k_i, k_j) = \{\Delta(k_i, k_j), \theta(k_i, k_j)\} \tag{4}$$

For two sets of keypoints L and M extracted for an individual at time t and t-λ defined as:

$$L_t = \{l_1, l_2, l_3 \dots l_\gamma\} \tag{5}$$

$$M_{t-\lambda} = \{m_1, m_2, m_3 \dots, m_\gamma\} \tag{6}$$

where λ is the time difference in seconds and γ is the number of keypoints that are extracted (as we are using OpenPose the value for γ used is 25). The set of keypoint changes between L and M are calculated as:

$$C(L_t, M_{t-\lambda}) = \{c(l_1, m_1), c(l_2, m_2), c(l_3, m_3), \dots c(l_\gamma, m_\gamma)\} \tag{7}$$

To compute the coarse representation of the individual's movement (in our experiments this was done using a 0.2s time period) we calculate 15 such sets of keypoint changes in order to build up a temporal history. The final feature vector at time t is defined as:

$$Coarse_t = \{C(L_t, M_{t-\lambda}), C(L_{t-\lambda}, M_{t-2\lambda}), \dots C(L_{t-14\lambda}, M_{t-15\lambda})\} \tag{8}$$

To compute the fine-grained representation of an individual's movement, again a set of 15 such keypoint changes is used in order to build up a temporal history of the individuals movement over time. This feature vector is defined as:

$$Fine_t = \{C(L_t, M_{t-\varepsilon}), C(L_{t-\lambda}, M_{t-\lambda-\varepsilon}), \dots C(L_{t-14\lambda}, M_{t-14\lambda-\varepsilon})\} \tag{9}$$

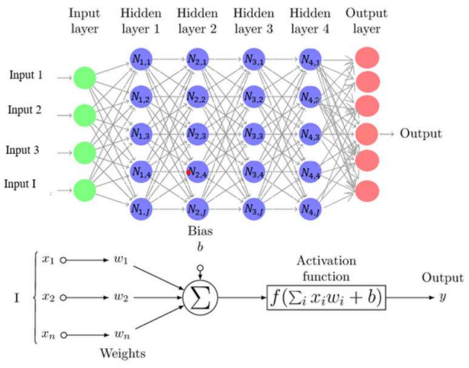

Figure 1 Graph representation of an MLP. The weights are represented by the edges of the graph.

where ε is defined as a short time period such that $\varepsilon < \lambda$ (in our experiments the value for λ was 0.2 seconds and the value for ε was 0.04 seconds).
For the combined approach, the feature vector is simply defined as:

$$Combined_t = \{Coarse_t, Fine_t\} \quad (10)$$

These features were subsequently used to train a multi-layer perceptron for classification.

Multilayer perceptron (MLP) refers to a feedforward artificial neural network. Arguably one of the simplest forms of an artificial neural network, an MLP consists of at least three layers of neurons, an input layer, a hidden layer and an output layer. Based on the biological neural networks that make up the brain (Minsky & Papert, 1988), MLPs are one of the oldest methods for supervised machine learning. Despite this they are still used for a large number of problems, and serve as a foundation for deep learning (Lin, Liang, 2020).

Figure 1 shows a simple graph representation of the MLP algorithm, which can be briefly described at follows. For an input vector of length *I* feeding into a hidden layer of *J* neurons, we define a set of weights $w_{i,j}$, where *j* refers to the neuron in question and *i* refers to the neuron in the previous layer to which *j* is connected. Formally for input vector *X* defined as:

$$X = \{x: x \in \mathbb{R}^I\} \quad (11)$$

The weights *W* with *J* rows and *I* columns can be defined as:

$$W = \{w: w \in \mathbb{R}^{J \times I}\} \quad (12)$$

and the set of biases *B* defined as:

$$B = \{b: b \in \mathbb{R}^J\} \quad (13)$$

The net inputs to a given neuron *j* are then calculated as the sum of the inputs multiplied by their respective weights plus the bias value:

$$net(j) = \sum_{i=1}^{I} x_i . w_{ji} + b_j \quad (14)$$

The net output is then calculated using an activation function *F*. In this paper we use a rectified linear activation function defined as:

$$F(x) = \max(0, x) \quad (15)$$

Therefore, the output for a given neuron j can be expressed as:

$$y_j = F\big(net(j)\big) \quad (16)$$

For a network with more than one hidden layer, the output from the previous layer is used as the input for the next layer. Thus, each hidden layer has its own set of biases and weights. The final layer of the network is the output layer and outputs the prediction of the network. A SoftMax activation function was used on the final layer to determine the prediction.

In order to train the network, the weights and biases are updated via backpropagation, using a stochastic gradient decent optimizer in order to minimize the network loss function. In this paper we use a sparse categorical cross entropy loss function defined as:

$$CCE(y, \hat{y}) = -\frac{1}{N}\sum_{j=0}^{N} y_j \cdot Log\big(\hat{y}_j\big) \quad (17)$$

where *N* is the number of elements in the training set, y is the ground truth, $\hat{y}$ is the estimate, log is the natural log and · is the inner product. The network is trained over a maximum of 500 epochs, with early stopping used to prevent overfitting.

3. EXPERIMENTAL SETUP

For the first experiment, a coarse representation of the keypoint changes was used, as defined in equation 8. The value used for λ was 0.2 seconds. This results in an overall temporal history of 3 seconds, and the resulting feature vector with 750 features.

For the second experiment, the fine-grained representation defined in equation 9 was used. The value used for λ was 0.2 seconds and the value used for ε was 0.04. This enabled a finer grained representation of the instantaneous change of the keypoint, while maintaining a feature vector of 750 features.

For the final experiment, the combined approach described in equation 10 was used. Again, the value for λ was 0.2 seconds and the value for ε was 0.04 seconds. The resulting feature vector contained a total of 1500 features. This enabled a more robust representation of the keypoint changes over the given time period.

In each of the three experiments, these features were subsequently used to train an MLP for activity recognition. The network has four hidden layers, each containing 450 neurons with a rectified linear activation function. These parameters were optimized using a grid search in order to maximize classification accuracy. As discussed in Section 2, the network was trained using a stochastic gradient descent optimizer to minimize a sparse categorical cross entropy loss function.

For our experiments, the data were split using leave-one-out cross validation as recommended by (Gao, Z., Chen, M. Y., Hauptmann, A. G., & Cai, 2010), where the set of videos for one individual is used for testing and the rest are used for training. The task therefore is to classify the activity exhibited by an unknown individual. The model was trained over a maximum of 500 epochs. In order to prevent overfitting, early stopping was used if the training accuracy failed to increase after 10 epochs.

4. ACCURACY EVALUATION

We evaluated the approaches on two simple but well-known datasets, the KTH dataset (Schüldt et al., 2004) and Weizmann dataset (Gorelick et al., 2007). The KTH dataset contains short video clips of 6 distinct actions: Walking, Jogging, Running, Boxing, Clapping and Waving. For each activity there are 25

sets of videos each containing a different individual. Each video set contains 4 videos, each with a different background: outdoors, outdoors with a different scale, outdoors with different clothes and indoors. This results in a total of 600 video clips, with an average length of 4 seconds, recorded at a rate of 25fps. The videos have a resolution of 160x120 pixels. Figure 2 shows example frames from the dataset. The results were validated using "leave one out" cross validation, where 24 of the video sets were used for training and one set was used for testing. The OpenPose library (Cao et al., 2017) was used for keypoint extraction as it provides a high degree of accuracy with real time performance.

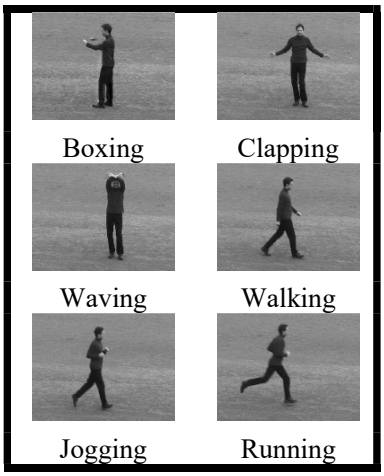

Figure 2 Example frames of the six activities from the KTH dataset

The confusion matrix for the first experiment (coarse approach), where the keypoint changes were each calculated over a time period of 0.2 seconds, is presented in Table 1. These results show that this approach achieves a classification accuracy of > 93% for four of the six activities. The average accuracy across all activities for this approach was 92.7%. The approach did struggle to differentiate between the jogging and running activities as these activities appear to be quite similar. However, this proposed approach was still able to separate these two classes with over 70% accuracy.

The results for the second experiment (fine grained), where the keypoint changes were calculated over a 0.04 second time period, are presented in Table 2. As can be seen from Table 2, the accuracy of the approach decreased when the changes were calculated over this shorter time period and the approach again struggled to differentiate between the running and jogging activities. However, the accuracy of the three non-locomotion activities (Boxing, Clapping and Waving), while lower than the coarse 0.2 second approach, remained over 93%. The average accuracy of this approach was 89.8%, a reduction of ~3% compared to the coarse approach in table 1. We postulate that this slight reduction in accuracy may be due to the fine grain keypoint changes not encoding as much temporal information about the movement as the coarse representation.

Results for the third experiment (combined approach) where the keypoint changes were calculated over both a 0.2 second time period and 0.04 seconds are presented in Table 3. These results show that using both sets of keypoint changes resulted in an increase in classification accuracy for all six classes. The classification accuracy for the three non-locomotive classes is now over 96%.

Additionally, the accuracies for the three locomotive activities are all over 80%. It is still difficult to differentiate between the jogging and running classes. However, this combined approach is more accurate than both the coarse and fine grained approaches individually. The average accuracy for this approach was 94.2%. This is an increase of ~1.6% over the coarse approach, and ~4.6% over the fine-grained approach.

Table 1 (Top) Coarse Approach, Table 2(Middle) Fine Grained approach and Table 3 (Bottom) the combined approach accuracy results for the KTH dataset

Coarse Approach						
	Walk	Jog	Run	Box	Clap	Wave
Walk	94.0	4.7	0.57	0.45	0.18	0.11
Jog	10.5	78.7	10.1	0.41	0.12	0.14
Run	4.02	22.4	72.8	0.35	0.17	0.18
Box	1.23	0.23	0.12	96.8	1.14	0.52
Clap	0.83	0.13	0.1	0.83	94.9	3.2
Wave	0.63	0.12	0.08	0.41	2.68	96.1
Total Accuracy 92.7%						

Fine Grained approach						
	Walk	Jog	Run	Box	Clap	Wave
Walk	87.9	13.0	2.33	0.88	0.36	0.12
Jog	8.05	67.5	22.8	0.34	0.26	0.07
Run	1.25	17.5	72.3	0.24	0.24	0.05
Box	1.62	1.07	0.93	94.8	1.32	1.31
Clap	0.62	0.38	0.74	1.88	93.9	3.02
Wave	0.60	0.58	0.87	1.87	3.89	95.4
Total Accuracy 89.8%						

Combined approach						
	Walk	Jog	Run	Box	Clap	Wave
Walk	94.3	4.49	0.65	0.37	0.12	0.08
Jog	7.61	82.6	9.41	0.29	0.08	0.06
Run	2.44	17.0	80.1	0.31	0.1	0.11
Box	1.1	0.12	0.06	97.4	0.76	0.58
Clap	0.2	0.03	0.02	0.55	96.4	2.81
Wave	0.15	0.02	0.01	0.35	2.34	97.1
Total Accuracy 94.2%						

It is worth noting that the approach of (Reid et al., 2020) achieved a total accuracy of 90.2% on this dataset. Our coarse approach detailed above outperforms their method by >2% and

the combined approach improves upon that by ~4%, while still using a reduced sample rate for action recognition.

To demonstrate that our proposed approach is not dataset dependent, we also evaluated it using the Weizmann dataset (Gorelick et al., 2007). This dataset contains short video clips of 9 distinct actions: walking, running, jumping, stepping sideways, bending, waving with one hand, waving with two hands, jumping in place, jumping jack and skipping. For each activity there are 10 sets of videos, each containing a different individual. This results in a dataset of 90 videos, recorded at a rate of 50fps interlaced. The videos had a resolution of 180 x 144 pixels. Again, "leave one out" cross validation was used to verify the results, with one set used for testing and nine sets used for training.

The results from the first experiment (coarse approach), where the keypoint change was measured over 0.2 seconds are presented in Table 4. As can be seen, the accuracy of the approach for the Weizman dataset was significantly lower than for the previous dataset. This may be due to two factors: firstly the size of the dataset was significantly smaller, only 90 videos as opposed to 600 in the KTH dataset, and secondly the duration of the videos was much shorter, averaging ~2 seconds per activity rather than the ~4 seconds for the KTH dataset. This makes it difficult for our approach to build a complete history of the keypoint changes for the action. However, the approach still achieved an accuracy of ~70%, with the majority of classes being classified correctly.

Like the KTH dataset, it was difficult for the coarse approach differentiate between activities which were similar in appearance. The two waving activities, waving with one hand and waving with two hands, were very similar, with below 50% accuracy for both activities. Additionally, the three locomotive activities, skipping, running, and walking had a large degree of similarity with each other, as with the KTH dataset. The skipping activity also had a large degree of similarity to the jumping activity.

The results for the second experiment (fine grained approach) where the keypoint change was measured over 0.04 seconds, are presented in Table 5. As can be seen, the average accuracy of this approach was approximately 2% higher than for the coarse 0.2 second approach. The confusion between the two hand waving classes was significantly lower than with the coarse approach. However, the confusion between the skipping and running classes was significantly higher. These results indicate that the effectiveness either coarse or fine grained keypoint changes depends on the activities which are being classified.

The results for the third experiments (combined approach), where the keypoint changes were calculated over 0.2 seconds (coarse) and 0.04 seconds (fine) are presented in Table 6. As can be seen, this approach outperformed both other approaches by a significant margin. The average classification accuracy was ~7% higher than the fine-grained approach, and ~9% more accurate than the coarse approach. The accuracy for every activity was significantly higher than for either method individually. There is still some confusion between classes which are similar in appearance, with the skip class having high confusion with both the running and jumping classes. However, this was significantly lower than for the other two approaches individually.

These results show that using the combined key point changes can result in a significant improvement in classification accuracy while still maintaining a sparse representation of the video frame. This may be because certain activities are easily identifiable when observed over a long period, whereas other activities are more easily identified over a shorter period. For example, the two hand waving activities were more easily identified when keypoint changes are calculated over a shorter time period, whereas the locomotion activities were more easily identified over a larger time period. By calculating the changes over both short and long time periods, the MLP can more easily differentiate between both sets of activities, thus improving the average accuracy.

	Bend	Jack	Jump	P.Jump	Run	Slide	Skip	Walk	Wave 1	Wave 2
Bend	85.3%	0.3%	6.3%	0.2%	0.2%	0.9%	1.9%	0.3%	1.7%	3.0%
Jack	0.3%	92.0%	0.5%	5.5%	0.0%	1.0%	0.0%	0.0%	0.4%	0.3%
Jump	1.7%	0.7%	77.9%	0.0%	0.7%	2.2%	13.3%	2.4%	0.4%	0.7%
P.Jump	0.0%	11.2%	0.7%	76.4%	0.0%	1.1%	0.0%	0.0%	0.4%	10.2%
Run	0.0%	0.2%	6.6%	0.0%	51.2%	4.4%	18.0%	19.0%	0.2%	0.2%
Slide	0.0%	0.5%	3.6%	0.0%	4.1%	84.0%	2.3%	2.9%	2.0%	0.7%
Skip	1.0%	0.2%	19.2%	0.0%	13.9%	3.1%	46.3%	10.6%	5.3%	0.4%
Walk	0.0%	0.0%	2.7%	0.0%	10.5%	1.8%	5.2%	79.7%	0.1%	0.0%
Wave 1	2.5%	1.2%	1.7%	1.5%	0.0%	2.0%	1.4%	0.5%	46.4%	42.9%
Wave 2	5.4%	3.8%	1.1%	5.1%	0.0%	2.1%	1.1%	0.0%	32.9%	48.4%
	Total Accuracy 69.6%									

	Bend	Jack	Jump	P.Jump	Run	Slide	Skip	Walk	Wave 1	Wave 2
Bend	86.5%	0.9%	0.0%	6.4%	0.0%	0.2%	1.3%	3.0%	0.5%	1.3%
Jack	0.4%	90.5%	0.0%	7.8%	0.0%	0.1%	0.7%	0.0%	0.0%	0.4%
Jump	0.0%	0.0%	60.9%	8.7%	2.0%	0.0%	24.5%	3.9%	0.0%	0.0%
P.Jump	0.6%	6.1%	0.2%	91.4%	0.0%	0.0%	0.9%	0.0%	0.0%	0.7%
Run	0.0%	0.0%	7.1%	9.8%	40.5%	0.7%	35.1%	6.8%	0.0%	0.0%
Slide	0.0%	8.8%	0.9%	9.0%	0.5%	77.9%	1.8%	1.1%	0.0%	0.0%
Skip	0.0%	0.0%	20.0%	8.2%	24.7%	0.4%	37.6%	9.2%	0.0%	0.0%
Walk	0.1%	0.0%	3.1%	5.6%	1.4%	0.6%	4.7%	84.2%	0.1%	0.1%
Wave 1	0.6%	0.0%	0.0%	6.3%	0.0%	0.6%	0.8%	2.3%	73.4%	16.1%
Wave 2	3.7%	1.9%	0.2%	8.2%	0.0%	0.3%	1.0%	0.8%	29.8%	54.2%
	Total Accuracy 71.9%									

	Bend	Jack	Jump	P.Jump	Run	Slide	Skip	Walk	Wave 1	Wave 2
Bend	90.1%	0.3%	2.2%	0.0%	0.3%	1.4%	2.3%	0.3%	1.6%	1.4%
Jack	0.3%	95.9%	0.1%	1.6%	0.0%	1.2%	0.0%	0.0%	0.4%	0.4%
Jump	1.3%	0.4%	78.6%	0.2%	2.4%	2.4%	14.0%	0.4%	0.0%	0.2%
P.Jump	0.0%	2.8%	0.2%	94.4%	0.0%	1.7%	0.0%	0.0%	0.0%	0.9%
Run	0.2%	0.0%	0.7%	0.0%	56.1%	3.2%	22.9%	16.6%	0.0%	0.2%
Slide	0.0%	0.2%	2.3%	0.0%	2.3%	92.1%	0.9%	1.1%	0.7%	0.5%
Skip	1.2%	0.0%	17.8%	0.0%	18.6%	2.4%	49.8%	9.6%	0.0%	0.6%
Walk	0.0%	0.0%	1.1%	0.0%	8.9%	2.0%	2.4%	85.6%	0.0%	0.0%
Wave 1	0.8%	0.0%	0.5%	0.2%	0.0%	3.2%	0.5%	0.2%	75.8%	19.0%
Wave 2	2.1%	2.4%	0.8%	0.3%	0.0%	3.2%	1.6%	0.0%	30.6%	59.0%
	Total Accuracy 79.0%									

Table 4 (Top) Coarse approach, Table 5(Middle) Fine grained approach and Table 6(Bottom) the combined approach accuracy results for the Weizman dataset.

5. RUNTIME EVALUATION

We computed the computation time of the combined approach using the Weizmann dataset which consists of 5701 frames. Experiments were conducted on an Intel XeonE5-1620 PC running Ubuntu version 18.04.3. The GPU used was a Nvidia Titan Xp with 16GB RAM. This is consistent with other approaches such as (Camarena et al., 2019) who also used GPU accelerated hardware when testing the runtime of their approach. The time taken for the OpenPose library to compute the key points for the entire set was 227.3 seconds. This is a rate of 39.8ms per frame and represents the most significant bottleneck of this approach. The time taken to compute the set

of keypoint changes for the entire dataset is 1.7 seconds; approximately 0.3ms per frame. Additionally, it takes the MLP algorithm 1 second to classify the activities for the test set, which consists of 701 frames. Therefore, classification is performed at a rate of 1.39ms. The total computation time for the entire pipeline is 41.5ms per frame; 24.0 frames per second. The runtime for the KTH dataset was also calculated and found to be the same. Hence, the approach is fast enough to perform activity recognition in real time.

Table 7 presents comparative results for the proposed approach and other state of the art approaches using the KTH dataset. Table 7 shows that the approach of Wang et al., (Wang et al., 2013) achieves an accuracy of 95.7%. While this is higher than the proposed approach, the computational cost of this method prevents it from running in real time. We also compare our approach with that in (Reid et al., 2020) who used a reduced sample rate and sample size to achieve real time performance using body keypoints. The proposed approach performs significantly better, indicating that the use of keypoint changes is a more robust alternative to simply reducing the sample rate and sample size while maintaining the real-time performance.

Table 7 Comparison of approaches on the KTH dataset

Performance evaluation using the KTH dataset		
Approach	*Accuracy*	*Speed/FPS*
(Wang et al., 2013)	95.7%	3
(Reid et al., 2020)	90.2%	24
Keypoint Changes	94.2%	24

6. CONCLUSION

We have presented a method for human activity recognition based on calculating the key points changes (Euclidean distance and angle). We have shown that this approach achieves accuracy on par with current state of the art methods, while using a sparse representation. Further, we have conducted run-time experiments and shown that this method is sufficiently fast enough for real time applications. In future work we will investigate how this approach performs for multi-person activity recognition and adapt this approach for more complex activities and scenes involving one or more people.

7. REFERENCES

Cai, Y., Wang, Z., Yin, B., Yin, R., Du, A., Luo, Z., Li, Z., Zhou, X., Yu, G., Zhou, E., Zhang, X., Wei, Y., & Sun, J. (2019). Res-steps-net for multi-person pose estimation. *Joint COCO and Mapillary Workshop at ICCV 2019: COCO Keypoint Challenge Track*.

Camarena, F., Chang, L., & Gonzalez-Mendoza, M. (2019). Improving the dense trajectories approach towards efficient recognition of simple human activities. *2019 7th International Workshop on Biometrics and Forensics (IWBF)*, 1–6.

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7291–7299.

Choutas, V., Weinzaepfel, P., Revaud, J., & Schmid, C. (n.d.). *PoTion: Pose MoTion Representation for Action Recognition*.

D'Sa, A. G., & Prasad, B. G. (2019). An IoT Based Framework For Activity Recognition Using Deep Learning Technique. In *ArXiv Preprint*. http://arxiv.org/abs/1906.07247

Dollar, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 65–72.

Efros, A. A., Berg, A. C., Mori, G., & Malik, J. (2003). Recognising Action at a distance. *Proceedings Ninth IEEE International Conference on Computer Vision*, *2*, 726–733. https://doi.org/10.1017/s1358246107000136

Gao, Z., Chen, M. Y., Hauptmann, A. G., & Cai, A. (2010). Comparing evaluation protocols on the KTH dataset. *International Workshop on Human Behavior Understanding*, 88–100.

Gorelick, L., Blank, M., Shechtman, E., Member, S., Irani, M., & Basri, R. (2007). Actions as space time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247–2253. https://doi.org/10.1109/TPAMI.2007.70711

Guo, K., Ishwar, P., & Konrad, J. (2010). Action recognition using sparse representation on covariance manifolds of optical flow. *Proceedings - IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2010*, 188–195. https://doi.org/10.1109/AVSS.2010.71

Jain, M., Jégou, H., & Bouthemy, P. (2013). *Better exploiting motion for better action recognition*. https://doi.org/10.1109/CVPR.2013.330

Ke, Y., Sukthankar, R., & Hebert, M. (2005). Efficient Visual Event Detection Using Volumetric Features. *Tenth IEEE International Conference on Computer Vision (ICCV'05)*, 166–173. https://doi.org/10.1109/CVPR.2007.383137

Laptev, I. (2004). *Local Spatio-Temporal Image Features for Motion Interpretation*.

Lee, D. G., & Lee, S. W. (2019). Prediction of partially observed human activity based on pre-trained deep representation. *Pattern Recognition*, *85*, 198–206. https://doi.org/10.1016/j.patcog.2018.08.006

Lin, Liang, et al. (2020). The Foundation and Advances of Deep Learning. In *Human Centric Visual Analysis with Deep Learning* (pp. 3-13.).

Matikainen, P., Hebert, M., & Sukthankar, R. (2009). Trajectons: Action recognition through the motion analysis of tracked features. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009*, 514–521. https://doi.org/10.1109/ICCVW.2009.5457659

Minsky, M. L., & Papert, S. A. . (1988). *Perceptrons: expanded edition*.

Reid, S., Vance, P., Coleman, S., Kerr, D., & O'Neill, S. (2020). Towards real time activity recognition. *Procedings of the Fourth IEEE International Conference on Image Processing, Applications and Systems (IPAS 2020)*, In Press.

Schüldt, C., Laptev, I., & Caputo, B. (2004). Recognizing human actions : A local SVM approach. *Proceedings of the 17th International Conference on Pattern Recognition (ICPR)*, *3*(September 2004), 32–36. https://doi.org/10.1109/ICPR.2004.1334462

Sheeba, P. T., & Murugan, S. (2019). Fuzzy dragon deep belief neural network for activity recognition using hierarchical skeleton features. *Evolutionary Intelligence*, *0123456789*, 1–18. https://doi.org/10.1007/s12065-019-00245-2

Subedar, M., Krishnan, R., Meyer, P. L., Tickoo, O., & Huang, J. (2019). Uncertainty-aware Audiovisual Activity Recognition using Deep Bayesian Variational Inference. *Proceedings of the IEEE International Conference on Computer Vision*, 6301–6310.

Sun, J., Mu, Y., Yan, S., & Cheong, L. F. (2010). Activity recognition using dense long-duration trajectories. *2010 IEEE International Conference on Multimedia and Expo, ICME 2010*, 322–327. https://doi.org/10.1109/ICME.2010.5583046

Wang, H., Kläser, A., Schmid, C., & Liu, C. (2013). *Dense Trajectories and Motion Boundary Descriptors for Action Recognition*. 60–79. https://doi.org/10.1007/s11263-012-0594-8

Wang, H., Kläser, A., Schmid, C., & Liu, C. L. (2011). Action recognition by dense trajectories. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3169–3176. https://doi.org/10.1109/CVPR.2011.5995407

Xiu, Y., Wang, H., & Lu, C. (2018). Pose Flow : Efficient Online Pose Tracking. *British Machine Vision Conference*, 1–12.

Yi, Y., & Wang, H. (2018). Motion keypoint trajectory and covariance descriptor for human action recognition. *Visual Computer*, *34*(3), 391–403. https://doi.org/10.1007/s00371-016-1345-6

Zelnik-Manor, L., & Irani, M. (2001). Event-based analysis of video. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2*(1229), 123–130. https://doi.org/10.1109/cvpr.2001.990935