Interpretable Classification of Human Exercise Videos through Pose Estimation and Multivariate Time Series Analysis

Ashish Singh (✉), Binh Thanh Le, Thach Le Nguyen, Darragh Whelan,
Martin O'Reilly, Brian Caulfield, Georgiana Ifrim (✉)

Insight Centre for Data Analytics, University College Dublin, Ireland
Output Sports Limited, NovaUCD, Dublin, Ireland
{ashish.singh, thanh.binh, thach.lenguyen, brian.caulfield, georgiana.ifrim}@insight-centre.org
{darragh, martin}@outputsports.com

Abstract. In this paper we present an approach for the classification and interpretation of human motion from video data. Our work builds upon the state-of-the-art advances in the area of Human Pose Estimation for video and Multivariate Time Series Classification and Interpretation. Our goal is to facilitate physiotherapists, coaches and rehabilitation patients by providing feedback after the execution of physical exercises. Recent work in sports science focuses on data collection with sensor devices, followed by a feedback step to the user. For example, the participant executes an exercise, and an application tells them whether the exercise was executed correctly or not, and what part of the movement was not executed correctly. Using sensors for collecting motion data has its challenges, for example, sensor devices require careful calibration, may not capture the full richness of the movement and are not easily accepted by users. Instead, we work with video data captured via mobile cameras, transform the video into time series via human pose estimation, train time series classifiers, and deliver feedback to the user through a time series classification and explanation step. We evaluate our approach on a real-world Crossfit Workout Activities dataset collected by the Personal Sensing Group at the Insight Centre for Data Analytics, University College Dublin, Ireland. We show that, although data capture with video and pose estimation is noisy, we obtain encouraging results with this approach and can provide useful feedback to the users.

1 Introduction

The majority of research conducted in the area of Human Activity Recognition (HAR) is based on motion sensor data [1, 25, 26, 27, 14]. This commonly involves extracting domain-specific or pre-defined statistical features from sensor data and applying supervised machine learning methods. However, using sensors to collect human activity data has some notable limitations [45]. For example, the sensors are costly and require careful positioning on the body, as well as calibration for the specific task. Furthermore, tracking multiple body segments requires multiple sensors, leading to issues with participant comfort and ease of movement, particularly when applied over long time periods. Instead of sensors, in this paper, we propose to use video recordings for data capture in HAR. Using video data helps to alleviate some of these problems, as videos can be easily captured through mobile phones that are widely available and accepted by users. Moreover, for some HAR tasks, it is easier to capture the differences between different types of movement by using video instead of sensor data. This is particularly the case when the task involves tracking multiple sensors placed on multiple body segments and locations. There are many video benchmark datasets for HAR which are relatively large and capture data over longer duration than

is typically done with sensors [15]. However, unlike many existing benchmarks where the activities are fairly easy to differentiate, in this work we focus on human exercise classification and interpretation, which is a harder task, because the differences between correct versus incorrect movement are more subtle. In particular, we use video recordings of participants executing the Military Press (MP) exercise. This is a commonly used exercise in strength and conditioning, injury risk screening and rehabilitation [45]. Incorrect execution of the Military Press may lead to musco-skeletal injuries and impede performance [2]. Therefore, feedback on the execution is important to avoid injuries and maximize the performance of the participants. Our research aims to use pre-recorded videos of the Military Press exercise to develop methods that can assists coaches and physiotherapists by automatically providing feedback about the execution of the exercise.

Our research methodology consists of two main steps. The first step uses human pose estimation [7] to extract multivariate time series data from videos. Given an input video, the time series data is obtained by applying pose estimation which tracks the location coordinates of multiple body parts over the video frames. The second step employs explainable time series classification methods [24, 14, 44]. This approach has many advantages: video data is easier to capture when compared to sensor data, all we need is a good camera which is widely available in smartphones, ipads, laptops, etc. Although video data is expensive with regard to storage and post-processing computation, through the pose estimation step, we reduce the data size by many orders of magnitude, since the data is now captured as a sequence of numeric values representing the location coordinates of body points, in the video frames. This can reduce the data size from Gb to Mb scale. Furthermore, recent work on time series classification also provides techniques to interpret the classifier results [14, 24]. For example, techniques such as saliency maps [35, 38, 41] can be helpful in pointing out the informative region of the time series which was considered by the classifier for making a prediction. In our work, we map the informative parts of the time series back to the frames of the original video, and give these important frames as feedback to the participant. We evaluate our approach on a real dataset collected at University College Dublin, Ireland. Figure 1 shows the overall flow of our proposed approach.

There are many approaches for HAR that use directly the video, without the pose estimation and time series analysis steps. Most of these techniques are based on complex deep learning architectures [17, 39, 42, 15] and require several hours of training. They also require expensive compute infrastructure, such as high end GPUs, and lack the classifier interpretation step. Furthermore, these models are trained and tested on datasets such as UCF-101 [40] and Kinetics-400 [18] which contain long duration videos and a wide range of activities. For instance, in Kinetics-400 the average duration of a clip is 10 seconds and the number of samples is around 300k. In our setting, the videos are of shorter duration (about 3 seconds, recorded at 30fps) and the differences between the classes are subtle, which makes the classification task more challenging. Our dataset is also small (a few thousand samples for training and validation), when compared to these large benchmarks.

Our main contributions in this paper are as follows:

- We propose a complete pipeline for the classification and interpretation of human motion captured via video. To achieve this, we build on state-of-the art human pose estimation and time series classification approaches. The data capture with video, combined with pose estimation to extract time series, is noisy and imperfect. Nevertheless, we show that by employing advanced time series classification methods we can already obtain encouraging accuracy for classifying the execution of a strength and conditioning exercise. The Military Press exercise studied in this paper is typically challenging for data capture with sensors [28, 45].

- We build on recent work for explaining the predictions of time series classifiers [14, 41] to provide an interpretation step as feedback to the user. For interpretation, we use saliency maps to select the most informative time series regions for the classification decision, and relate these back to informative frames in the video, which are then presented to the user.

- We provide an empirical study of recent multivariate time series classification and interpretation approaches on a real-world strength and conditioning dataset collected by sport science researchers at University College Dublin[1]. We discuss the benefits and challenges of these approaches, and propose future directions to further strengthen the benefits of exercise motion capture and analysis using video.

The rest of the paper is organized as follows. Section 2 presents an overview of related work, Section 3 describes the data collection procedure, Section 4 describes our approach for human exercise classification and interpretation and Section 5 evaluates the proposed approach on real-world data. Finally, Section 6 concludes and outlines directions for future work.

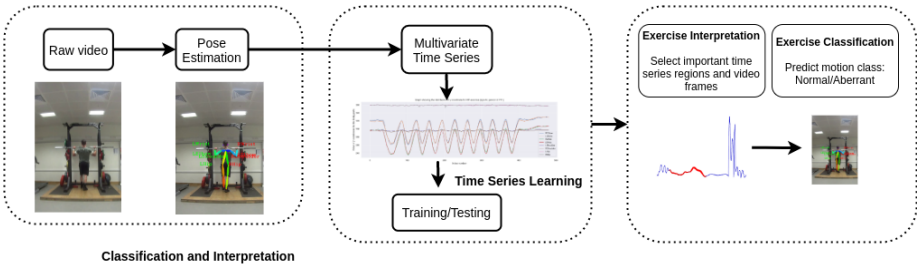

Fig. 1 Overview of our proposed approach for the Military Press exercise. Going from raw video to extracting and tracking body points using human pose estimation, and preparing the resulting data for time series classification and interpretation.

2 Related Work

We present a brief overview of existing approaches for human activity recognition from video, and discuss the latest advances in pose estimation and multivariate time series classification methods and their interpretation.

Human Activity Recognition (HAR). Video-based HAR is one of the core challenging problems in the area of computer vision. Some challenges for this task include: high data dimensionality, viewpoint changes, noisy background and complex motion dynamics. HAR methods can be broadly classified into two categories. The first category consists of methods based on handcrafted features such as bags of visual words [43, 11, 30, 31]. These include finding local spatio-temporal features such as motion boundary histograms [11] and trajectories [43], followed by feeding them to a classifier. These methods have been shown to provide competitive performance on benchmark datasets [8, 36, 40] before the emergence of deep learning methods. The second category includes methods based on deep learning, in particular convolutional neural networks. The recent success of 2D-CNN [19] in image classification has motivated researchers to employ these models for action recognition in video. Several models like 3D-CNN [17], two stream convolutional networks [39], I3D [9] and Slowfast [15] have been shown to achieve

[1] We released the anonymised time series dataset: `https://github.com/mlgig/video-pose-tsc`

state-of-the-art performance on benchmark datasets. However, these models suffer from modeling the long term dependencies and are computationally expensive to handle long duration videos. Approaches based on representing videos as space-time regions [49] and temporal layer [50] are able to overcome these limitations. Other architectures employ attention based mechanism [52, 51, 53] to focus on important regions of the video.

Human Pose Estimation. Pose estimation refers to recognizing the postures of humans from images. It is considered one of the hardest problems in computer vision due to challenges such as occlusion, complex motion dynamics, interactions, background, etc. Pose estimation has many applications, such as video surveillance and action recognition. The task becomes even more difficult when dealing with multiple persons in the same image or video frame. Earlier approaches were based on top-down methods which use a person detector over the whole image, whereas more recent bottom-up approaches, such as OpenPose [7], work by first finding the body joints and associating them using affinity fields. Other architectures based on deep learning [46, 10] and hourglass architectures [23] have been proposed and have been shown to give competitive performance on benchmarks. The latest pose estimation approaches can detect and track multiple body points in real-time and with high accuracy, for example OpenPose [7, 37], which is the approach we use in our work, can detect 18 body points in an input image in under 1 second and has an average accuracy ranging from 75.6% to 79% on recent 2D pose estimation benchmarks. While the output of pose estimation methods is still noisy, in this paper we model this output with the latest time series classification methods which are designed to be more robust to noise.

Multivariate Time Series Classification and Interpretation. Time series classification is a form of supervised classification where the data is ordered. Each sample in the data can have multiple features with ordered values. There are many proposed approaches for univariate time series classification (UTSC) which deal with a single feature. However, most of the phenomena in the real-world exhibit a multivariate nature, such as the signals from ECG, EEG and HAR, where for a single subject we capture multiple time series. It is thus important to develop effective multivariate time series classification (MTSC) methods. Most of the methods for MTSC are based on extending UTSC techniques. We can group existing methods for MTSC as follows [29, 13]: distance-based, shapelet-based, symbol-based, ensembles and deep learning. In our work we select a few effective time series classifiers from these groups as described below. We base our selection on the accuracy of the classifier, as well as the availability of code for interpreting the classifier prediction. As a baseline for classifier accuracy, we use 1NN-DTW [5]. This is a **distance-based** method built on dynamic time warping distance and a one nearest neighbor classifier, and is widely employed as the main baseline for TSC, due to its simplicity and accurate results. **Shapelet-based** methods such as the Shapelet Transform [6] and Learning Shapelets [16] are also very popular due to the fact that shapelets allow some level of interpretation of the classification model. Nevertheless, these methods were recently outperformed by more accurate approaches, and we do not consider them in our evaluation. In the **symbol-based** group we have symbolic classifiers such as MrSEQL [24] and WEASEL+MUSE [34]. These methods first transform the time series into sequences of symbols using transforms such as SAX [20] in the time domain or SFA [33] in the frequency domain, then learn a classifier on this data. MrSEQL [24, 41, 13] was recently shown to achieve high accuracy and also provides a step for interpreting the prediction via a saliency map. However, while MrSEQL and WEASEL + MUSE can achieve high accuracy for MTSC, they suffer from long running time, in particular for long time series and many dimensions. In the **ensembles** group we have HIVE-COTE v1.0 [21, 4], an ensemble of four classifiers and the current state-of-the-art with regard to classifier accuracy. However, this method suffers from high time complexity and memory consumption and does not provide any means for interpreting the classifier. Recent methods such as ROCKET [12] and **deep learning** methods such as Fully Convolutional Networks (FCN) and Resnet [14]

were shown to achieve state-of-the-art accuracy without suffering from high time and memory complexity. Apart from high accuracy, MrSEQL and deep learning methods also provide software for interpreting the classifier prediction which helps in identifying the informative regions in the input time series. Other MTSC methods such as [47, 48] can handle variable length and multivariate data by learning the embeddings of the time series. To recap, we select 1NN-DTW, ROCKET, FCN and Resnet to evaluate the accuracy and explanation capability of recent MTSC methods for our application.

3 Data Collection

Crossfit Workout Dataset. The data used for evaluating our approach are video recordings of the execution of the Military Press (MP) exercise. During this exercise the barbell is lifted to shoulder height and then smoothly lifted overhead by extending the elbows. The amount of weights lifted and time taken for each repetition may vary from participant to participant. Participants were asked to complete fixed repetitions of normal and induced forms for this exercise. The induced forms refer to predefined deviations as defined by the National Strength and Conditioning Association (NSCA) [2].

Participants. 56 healthy volunteers (34 males and 22 females, age: 26 +/- 5 years, height: 1.73 +/- 0.09 m, body mass: 72 +/- 15 kg) were recruited for the study. Participants did not have a current or recent musculo-skeletal injury that would impair performance of multi-joint upper limb exercises. The Human Research Ethics Committee at University College Dublin approved the study protocol and written informed consent was obtained from all participants before the study start.

Experiment Protocol. The testing protocol was explained to participants upon their arrival at the laboratory. Participants completed 10 repetitions of the normal form and 10 repetitions of induced forms. In order to ensure standardisation, technique was considered acceptable if it was completed as defined by NSCA guidelines. These were chosen based on common deviations listed in the NSCA guidelines [2] and through discussion with sports physiotherapists and strength and conditioning coaches.

Participants were allowed to familiarize themselves by completing practice repetitions. Two cameras (Sony Action Camera, Sony, Tokyo, Japan) were set up in front and to the side of the participants to allow for recording in the frontal and lateral planes simultaneously. The data is recorded at a rate of 30 frames per second with 720p resolution. Each of these individual video clips were then labelled according to participant number, exercise completed and if they were completed in an acceptable or aberrant manner. Each participant completed the set at their desired tempo.

Exercise Technique and Deviations. The induced forms were further sub-categorised depending upon the exercise. Below we describe the four classes of normal and deviated execution forms for the MP exercise.

Normal (N). This class refers to the correct execution of the exercise. The participant starts by lifting the bar from near shoulder to all the way above the head until the arms are fully stretched and then bringing it back to shoulder level with no arch in the back. The bar must be stable and parallel to the ground and the back should be straight.

Asymmetrical (A). This form refers to the execution when the bar is lopsided and asymmetrical.

Reduced Range (R). This form refers to the execution when the bar is not brought down completely to the shoulder level.

Arch (Arch). This type of execution indicates that the participants arches their back.

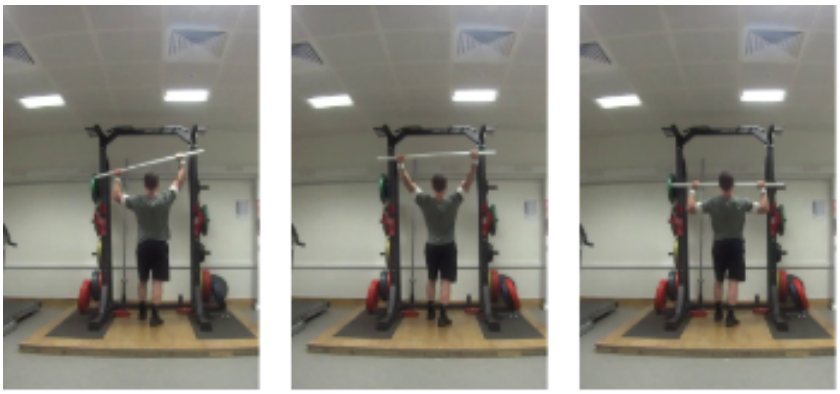

Fig. 2 Single frames depicting the induced MP deviations for class A, Arch and R (left to right).

4 Methods

We propose a two-step approach to classify the video recordings of the MP exercise. The raw data collected has four long videos per person corresponding to each class. Each video has 10 repetitions for that class. Currently, we only use the front-view camera recordings, but plan to study the side-view recordings in our future work. The proposed approach consists of applying the following steps for each video: (1) human pose estimation to obtain multivariate time series that track the location coordinates of body points; (2) segmentation of the long time series into individual exercise reps, (3) data pre-processing (eg resampling) and (4) classification and interpretation using multivariate time series methods. We describe each step in detail below.

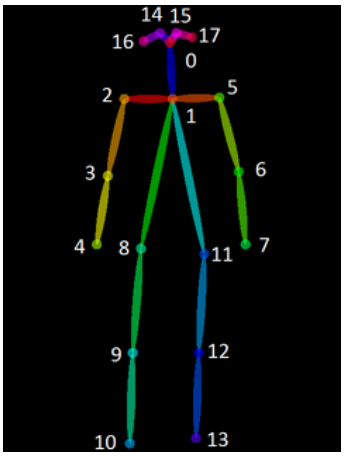

Fig. 3 OpenPose tracks 18 points on the human body. Figure taken directly from OpenPose repo[2].

0 Nose	6 Left Elbow	12 Left Knee
1 Neck	7 Left Wrist	13 Left Ankle
2 Right Shoulder	8 Right Hip	14 Right Eye
3 Right Elbow	9 Right Knee	15 Left Eye
4 Right Wrist	10 Right Ankle	16 Right Ear
5 Left Shoulder	11 Left Hip	17 Left Ear

Table 1 Body parts detected by OpenPose in a given video frame (index in Fig. 3).

Pose Estimation. We use OpenPose[3][4] [7] to extract the location coordinates of 18 body parts from the videos. OpenPose uses a pre-trained model based on the COCO[5] dataset. OpenPose uses part affinity fields (PAF) which encode the location and orientation of the body parts. Figure 3 shows the location of tracked body points and their corresponding names are given in Table 1. The frames are first cropped to remove unnecessary background and to centralize the participant. Cropping involves removing the extreme left and right portion of the image. This step does not risk cropping the participant as the camera settings remain unchanged for all the participants. The cropped frames are then given as input to OpenPose to obtain a sequence of x and y location coordinates for each body part and each video frame. Each frame is then considered a single time point in the output time series data. The above step is repeated for all the videos. The original

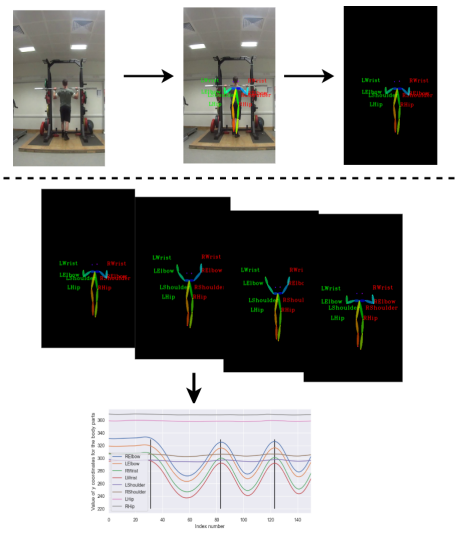

Fig. 4 Extraction of time series data from video. Each frame in the video is considered as a single time point in the resulting time series. Each body point results in a single time series.

videos amount to 4.5GB in storage, but after the pose estimation step and extracting the body points time series, the data size reduces to 22Mb, hence roughly a reduction of 200 times in data size. Figure 4 shows the use of pose estimation to track the coordinates of body parts over all the video frames. Figure 5 shows the raw y-coordinates for 8 body parts for a single video from class N. The time series obtained for body parts such as ankles, hips, etc. do not show much variability throughout the whole clip.

Segmentation. Due to the way the original data was recorded, each video records 10 reps, resulting in a time series capturing the body points movement for 10 reps. Since each rep is the record for a single exercise execution, segmentation of the long time series is required to obtain the sequence for a single rep. Each rep forms a single time series sample for training and evaluating a classifier. We use peak detection methods to

[3] `https://github.com/CMU-Perceptual-Computing-Lab/openpose`

[4] `https://github.com/michalfaber/keras_Realtime_Multi-Person_Pose_Estimation`

[5] http://cocodataset.org/

segment the data. First, we use the SavGol filter [32] to smooth the raw coordinates and get rid of smaller fluctuations which may hinder in finding the peaks. Second, we use a simple peak detection method to find the peaks. The body parts that are considered for finding the peaks are elbows or wrists as these are the only body parts showing regularity in the patterns as shown in Figure 5. The sequence from one peak to another consecutive peak constitutes the sequence for a single rep. Since there are 10 reps per video, the number of peaks obtained should be 11. However, due to the limitations of peaks detection (redundant peaks) and pose estimation (stability of coordinates) we may end up obtaining more than 11 peaks in some cases. Currently, we deal with this issue by removing sequences with a length less than 20, since due to the video sampling rate, we expect to have 30 frames per second, and the participants require around one second or more, to execute the MP exercise. This result in loss of few samples in the data. Each sample obtained after this step has variable length since the time taken to complete each rep differs from participant to participant. We can easily eyeball this data to cross verify the results of peak detection. The segmentation step is performed on a copy of the original data. Figure 5 shows the smooth coordinates and segmented sequences.

Data Pre-processing. The data is pre-processed to remove the body points time series with low variability such as ankles, knees, etc. Additionally, the sequences of x-coordinates are ignored due to low variability. The data for nose and eyes are also ignored as OpenPose fails to track these body parts, because the person is not facing the camera. Each sample is a multivariate time series, with variable length and 8 dimensions corresponding to 8 body parts being tracked (left and right shoulders, elbows, wrists and hips). Time series classification methods such as those based on deep learning and ROCKET cannot handle variable-length time series. Therefore, the data is resampled to the length of the longest time series (with a length of 161).

Train/Test Split. We perform a 70:30 split on the full data set to obtain training and testing data. The splitting is done based on the unique participant id to avoid leaking information into the testing data. By splitting on the id level we make sure all the samples from a particular participant go into either the training or testing data. The data is overall balanced across all the classes. Table 2 shows the number of samples across the four classes, for a single train/test split. There are roughly 1300 and 600 samples in training and testing data respectively. Each data sample is a multivariate time series data with a shape of 161x8 (161 length and 8 dimensions).

Definition: A multivariate time series with M dimensions can be defined as X = $[X^1, X^2, X^3, ...X^M]$ where each X^i is a univariate time series with $X^i \epsilon R^T$ where T is the length of the time series. In our case, the value of M is 8 and the value of T is 161.

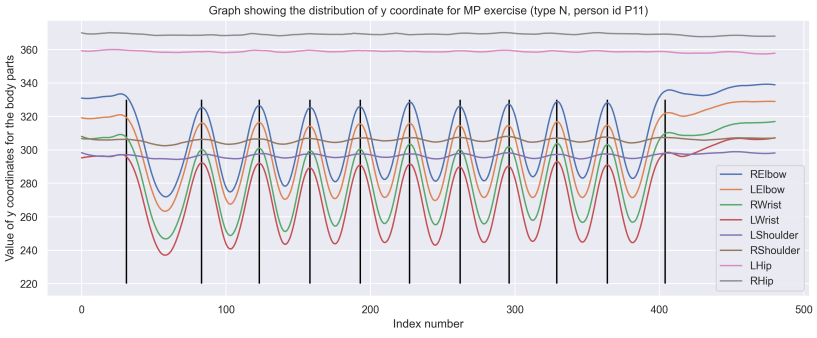

Fig. 5 Time series for different parts from a single participant. The sequence from one vertical line (solid black) to the consecutive vertical line constitutes one rep execution.

Class	Training	Test	Total
N	341	150	491
A	340	138	478
R	349	150	499
Arch	333	141	474
Total	**1363**	**579**	**1942**

Table 2 Total number of samples per class in the training and test datasets.

Time Series Classification. We use state-of-the-art multivariate time series classification methods to classify and interpret this data. We use 1NN-DTW as the baseline model and ROCKET and deep learning models for their accuracy and interpretability. Deep learning methods support interpretability through saliency maps. A saliency map is a numeric vector which highlights the discriminative (i.e., informative for classification) regions in the time series. We also include ROCKET as it was shown to be very effective on MTSC benchmarks. It is the fastest and much more efficient method than existing ensemble models. Though ROCKET does not provide code for interpreting the classifier, it is helpful to compare its accuracy to the other models. We present a brief overview of these classification methods.

1NN-DTW [3] is the accepted baseline in the time series classification community. Though this method suffers from high running time, it is one of the strongest benchmarks on multivariate time series classification in terms of accuracy [3, 29].

ROCKET [12] is a very recent time series classification method which was shown to achieve high accuracy, with low runtime. It transforms the data by projecting it into a higher dimensional space through random convolutional kernels. These kernels were shown to capture relevant features such as shape, frequency or variance without the need for handcrafted features. The transformed features are then fed to a linear classifier, e.g., Ridge Regression. This method is very fast as it does not require learning the kernels, in contrast to deep learning methods. We set the number of kernels to 10,000 as recommended in the original paper.

Deep Learning methods work by creating a hierarchy of features through multiple layers and using non-linear activation functions. We select the Fully Convolutional Network (FCN) and Residual Network (Resnet) based on their performance on TSC benchmarks [14]. The FCN architecture removes the pooling layer after each convolutional layer. This helps in keeping the data the same size as the original data. Resnet differs from other deep learning architectures by having a skip connection. This helps in training the network by reducing the vanishing gradient effect. In addition, both of these models replace the traditional fully-connected layer with a Global Average Pooling (GAP) layer which helps in drastically reducing the number of parameters. We use the default architectures for these models as described in [14]. Both models support a post-hoc model explanation step via the class activation map (CAM) which produces a saliency map explanation [35]. CAM was initially proposed for explaining image classification. It does so by highlighting the discriminative regions that contributed the most to a particular classification. ROCKET, FCN and Resnet only work with equal-length time series. Therefore, as discussed in the data pre-processing section, we resample all time series to equal length.

5 Experiments

All the time series classification methods are with the default hyper-parameters given in the original papers [14, 3, 12]. We use the sktime software[6] [22] for ROCKET and the original implementation[7] in [14] for FCN and Resnet. We use accuracy, recall, precision and F1-score to evaluate the classifier performance. We create three train/test splits (based on participant ids) and report the average score of these measures over these splits. Although, we did not observe much variance across the result over different data splits, we intend to do more extensive experiments using leave-one-out cross validation in future work.

5.1 Classifier Accuracy

Table 3 shows the average accuracy and standard deviation obtained on the test data over the three data splits. We first discuss the results using the un-normalized (middle column) data. All the recent MTSC methods outperformed the 1NN-DTW baseline in terms of accuracy. We did not present results of MrSEQL as the data gets normalized during the SAX transformation, which adversely affects the results for this application (since magnitude effects such as the full range of the movement cannot be captured if the data is normalised). Additionally, MrSEQL takes more training time than the other methods mentioned. ROCKET achieved the highest accuracy (0.81), followed by the deep learning models. The standard deviation values are generally insignificant. Nonetheless, DTW and FCN classifiers seem to be more affected by the different splits.

Classifier Name	Accuracy (Unnormalized data)	Accuracy (Normalized data)
1NN-DTW	0.58 (±0.04)	0.50 (±0.047)
ROCKET	**0.81** (±0.03)	**0.68** (±0.005)
FCN	**0.72** (±0.043)	**0.65** (±0.041)
Resnet	**0.73** (±0.028)	**0.65** (±0.025)

Table 3 Average accuracy on test data over three train/test splits. Normalising the time series significantly reduces the accuracy of all classifiers, due to losing information about the range and magnitude of the signal capturing the exercise movement.

Table 4 and 5 show the average precision, recall and F1-score for ROCKET and FCN respectively (Resnet results are similar to FCN). For detailed results, we also present the confusion matrix for a single split for ROCKET and FCN in Figure 6. Both models show similar performance, with higher precision and recall for class A (Asymmetrical) and R (Reduced Range) compared to the other classes. Class A has the highest precision and recall. This is likely because class A is easily discernible from the other classes.

Both methods struggle in detecting the Arch samples (where participants arched their backs). This is likely due to the fact that this subtle difference is hard to capture by the front camera. This suggests that using multiple cameras at different angles may help the classifiers. Samples from class R (Reduced Range) also confuse the models although to a lesser extent. This is where ROCKET shines in comparison with the other methods, as it detects 90% samples from this class. Note that the only execution error in this class is the range of the movement, which is reflected in the distances between the peaks and the valleys in the time series. In the TSC community data normalisation is typically considered a default step before training classifiers. As we discuss below, normalising

[6] https://github.com/alan-turing-institute/sktime

[7] https://github.com/hfawaz/dl-4-tsc

the data actually hurts the classifier accuracy in this application, so care is needed during data pre-processing.

Class Name	Precision	Recall	F1-score
A	0.99	0.88	0.93
Arch	0.73	0.80	0.76
N	0.79	0.67	0.72
R	0.77	0.90	0.83

Table 4 ROCKET: Average precision and recall per class on the test data.

Class Name	Precision	Recall	F1-score
A	0.93	0.87	0.90
Arch	0.64	0.63	0.62
N	0.59	0.58	0.59
R	0.79	0.80	0.79

Table 5 FCN: Average precision and recall per class on the test data.

Discussion. To check if data normalization has an effect on class R, we re-ran all the models on the normalized data and observed that there is a high reduction in precision and recall for this class. There is a decrease of 44% in average precision and recall for class R. However, normalization seems to have the opposite effect on class Arch with FCN. There is a reduction in all the metrics score when using the normalized data. A positive effect of using unnormalized data is that this may have helped the models in retaining the magnitude information. Additionally, other classes also seem to benefit from the unnormalized data. For ROCKET and deep learning methods we have disabled the normalisation step in the original code of the respective classifier. We think that in this application, the data should not be normalised in order to preserve the important information of each class. The rightmost column in Table 3 presents the accuracy using the normalized data. These results confirm that normalization in this case hurts the accuracy. We also experimented with other variants of data such as including the x-coordinates and padding the data with zeros (instead of resampling, to create equal-length time series). But none of these steps achieved better accuracy than the current approach.

5.2 Classifier Runtime

We used an Ubuntu 18.04 based machine with 72 cores and 503Gb RAM for all our experiments. We only utilized a single core for training and testing. In terms of training time, ROCKET was the fastest and 1NN-DTW was the slowest method. Table 6 shows the average training time over 3 splits.

Method	Training Time (minutes)
1NN-DTW	27
Rocket	1.2
FCN	6.3
Resnet	6.1

Table 6 Average training time for all the time series classification methods.

5.3 Classifier Feedback

The second contribution of our proposed approach is to provide feedback to the participants through visualisation. This can be done by finding and mapping the discriminative regions of the time series, back to the frames in the original video. The discriminative regions are the regions that distinguish samples from different classes

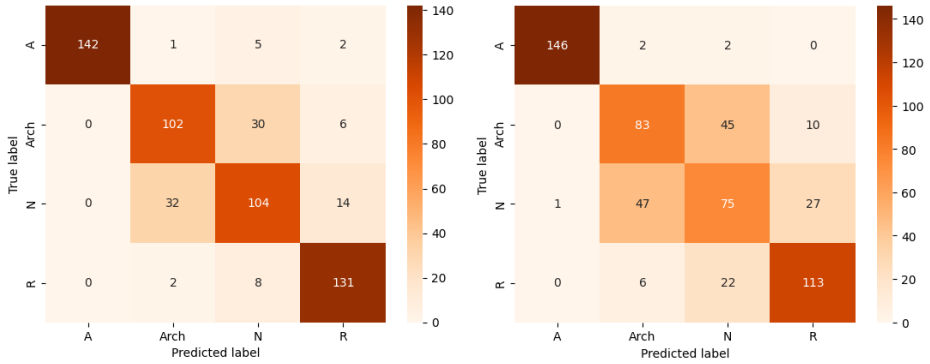

Fig. 6 Confusion matrix for a single test split. ROCKET (left) and FCN (right).

according to the classifiers. Explainable time series classifiers can identify such regions with saliency maps (e.g., Figure 7). It is worth mentioning that not all state-of-the-art time series classifiers are explainable. Among the evaluated classifiers, only deep learning based methods provide code for this support. Although the 1NN-DTW method can also provide feedback to a certain extent by showing the nearest example, it is unable to pinpoint where the execution is wrong, thus less useful in our opinion. A saliency map is a numeric vector indicating the importance of regions in the original time series. Both FCN and Resnet can compute the saliency maps using the CAM method. The visualisation is obtained by using a heatmap to highlight the discriminative regions. To demonstrate our feedback approach, we show two examples from class A (Asymmetrical) and class R (Reduced Range). Because with the current setup it is difficult to detect the errors in the Arch samples, we leave this issue for future work.

Class A. Figure 7 shows the discriminative region and the corresponding frames for FCN. The frames are taken from regions marked with a red ellipse. The FCN show the discriminative region on the left side which confirms the asymmetry (either on the left or right side) in the execution of this class. This confirms that the model was able to distinguish class A from other classes. The discriminative region is shown in cyan color. We also normalized the data from different saliency values of different samples to bring it under the same scale and observed that the relative position of the discriminative regions remains same.

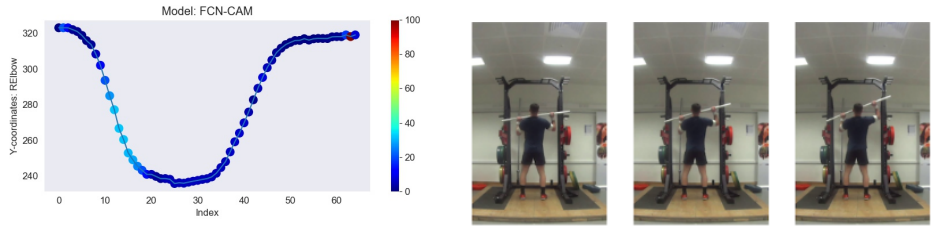

Fig. 7 FCN-CAM: Discriminative region (left) and the corresponding frames (right) for class A.

Class R. Figure 8 shows the discriminative regions and the corresponding frames for FCN-CAM. The discriminative regions are near the starting or the ending of the rep which refers to the action of not bringing the bar down completely to the shoulder level. The model was able to identify these regions as verified from the shown frames.

We also present the discriminative part per class by plotting the heatmaps for 50 correctly predicted samples from the training data set. Only those samples are selected where the predicted probability for the correct class is greater than 0.90. These samples may belong to a varied number of participants. This can help in analyzing the overall discriminative

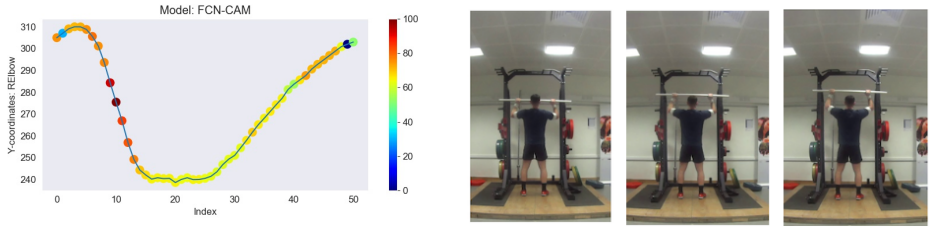

Fig. 8 FCN-CAM: Discriminative region (left) and the corresponding frames (right) in original clip for class R.

part per class chosen by the model. Figure 9 shows the visualizations using FCN-CAM for the discriminative parts for each class. As we see, the discriminative parts identified by the model for class A lie on the left or the right side which confirms the asymmetry in the execution. The discriminative part for class Arch generally lies in the bottom, whereas it lies on the extreme left or right for class R.

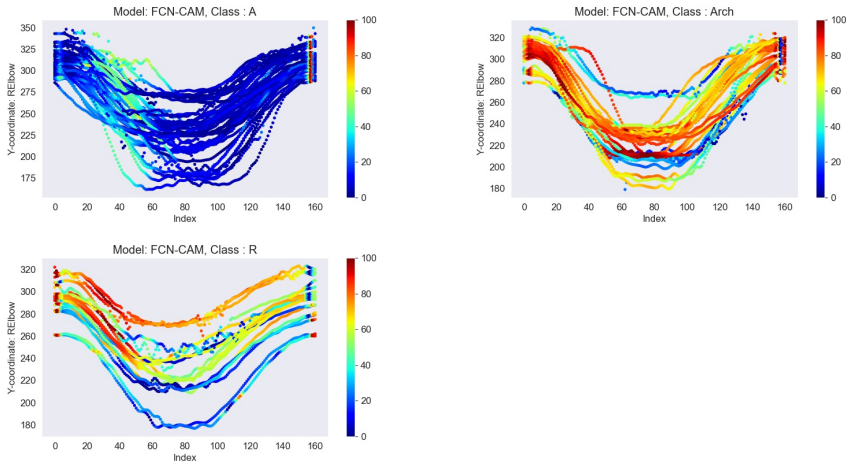

Fig. 9 FCN-CAM: Discriminative part for each class.

6 Conclusion

In this paper we have proposed a two step approach to classify and interpret human exercise motion captured in video recordings, through pose estimation and multivariate time series classification. We have evaluated our approach on a real-world dataset for the Military Press exercise and have achieved encouraging results. Additionally, we have also shown first steps towards providing interpretation for the classification decisions, which can serve as the basis for automated feedback to users. In our future work, we plan to investigate further methods for addressing the effect of noise in video capture and pose estimation, as well as the use of multiple video streams (e.g., front and side videos). We also plan on comparing our approach to classification and interpretation on sensor-based data capture, as well as direct video classification approaches.

Acknowledgments

This work was funded by Science Foundation Ireland through the Insight Centre for Data Analytics (12/RC/2289_P2) and VistaMilk SFI Research Centre (SFI/16/RC/3835).

References

[1] Ahmadi, A.; Mitchell, E.; Destelle, F.; Gowing, M.; O'Connor, N. E.; Richter, C.; and Moran, K. 2014. Automatic activity classification and movement assessment during a sports training session using wearable inertial sensors. *IEEE BSN*.

[2] Baechle, T. R.; and Earle, R. W. 2008. *Essentials of strength training and conditioning*. Human kinetics.

[3] Bagnall, A.; Dau, H. A.; Lines, J.; Flynn, M.; Large, J.; Bostrom, A.; Southam, P.; and Keogh, E. 2018. The UEA multivariate time series classification archive, 2018. *CoRR* abs/1811.00075.

[4] Bagnall, A.; Flynn, M.; Large, J.; Lines, J.; and Middlehurst, M. 2020. A tale of two toolkits, report the third: on the usage and performance of HIVE-COTE v1.0.

[5] Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; and Keogh, E. 2016. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *DAMI*.

[6] Bostrom, A.; and Bagnall, A. 2017. A Shapelet Transform for Multivariate Time Series Classification.

[7] Cao, Z.; Hidalgo Martinez, G.; Simon, T.; Wei, S.; and Sheikh, Y. A. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE TPAMI.*

[8] Carreira, J.; Noland, E.; Banki-Horvath, A.; Hillier, C.; and Zisserman, A. 2018. A Short Note about Kinetics-600. *CoRR* abs/1808.01340.

[9] Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *IEEE CVPR 2017*. 10.1109/CVPR.2017.502.

[10] Chen, Y.; Wang, Z.; Peng, Y.; Zhang, Z.; Yu, G.; and Sun, J. 2018. Cascaded Pyramid Network for Multi-Person Pose Estimation. In *IEEE CVPR 2018*.

[11] Dalal, N.; Triggs, B.; and Schmid, C. 2006. Human Detection Using Oriented Histograms of Flow and Appearance. In Leonardis, A.; Bischof, H.; and Pinz, A., eds., *ECCV 2006,*.

[12] Dempster, A.; Petitjean, F.; and Webb, G. I. 2020. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Min. Knowl. Discov.*.

[13] Dhariyal, B.; Nguyen, T. L.; Gsponer, S.; and Ifrim, G. 2020. An Examination of the State-of-the-Art for Multivariate Time Series Classification. In *LITSA, ICDM 2020*.

[14] Fawaz, H. I.; Forestier, G.; Weber, J.; Idoumghar, L.; and Muller, P.-A. 2019. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.*.

[15] Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. SlowFast Networks for Video Recognition. In *IEEE ICCV 2019*.

[16] Grabocka, J.; Schilling, N.; Wistuba, M.; and Schmidt-Thieme, L. 2014. Learning Time-series Shapelets. In *ACM SIGKDD 2014*.

[17] Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2010. 3D Convolutional Neural Networks for Human Action Recognition. In Fürnkranz, J.; and Joachims, T., eds., *ICML 2010*.

[18] Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The Kinetics Human Action Video Dataset.

[19] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Bartlett, P. L.; Pereira, F. C. N.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *NeurIPS*.

[20] Lin, J.; Keogh, E.; Wei, L.; and Lonardi, S. 2007. Experiencing SAX: a novel symbolic representation of time series. *Data Min. Knowl. Discov.*.

[21] Lines, J.; Taylor, S.; and Bagnall, A. J. 2018. Time Series Classification with HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles. *ACM Trans. Knowl. Discov. Data*.

[22] Löning, M.; Bagnall, A.; Ganesh, S.; Kazakov, V.; Lines, J.; and Király, F. J. sktime: A Unified Interface for Machine Learning with Time Series. In *NeurIPS 2019*.

[23] Newell, A.; Yang, K.; and Deng, J. 2016. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV 2016*.

[24] Nguyen, T. L.; Gsponer, S.; Ilie, I.; O'Reilly, M.; and Ifrim, G. 2019. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. *Data Min. Knowl. Discov.*

[25] O'Reilly, M.; Whelan, D.; Chanialidis, C.; Friel, N.; Delahunt, E.; Ward, T.; and Caulfield, B. 2015. Evaluating squat performance with a single inertial measurement unit. In *2015 IEEE BSN*.

[26] O'Reilly, M. A.; Whelan, D. F.; Ward, T. E.; Delahunt, E.; and Caulfield, B. M. 2017. Classification of deadlift biomechanics with wearable inertial measurement units. *Journal of biomechanics*.

[27] O'Reilly, M.; Caulfield, B.; Ward, T.; Johnston, W.; and Doherty, C. 2018. Wearable inertial sensor systems for lower limb exercise detection and evaluation: a systematic review. *Sports Medicine*.

[28] O'Reilly, M. A.; Whelan, D. F.; Ward, T. E.; Delahunt, E.; and Caulfield, B. 2017. Classification of lunge biomechanics with multiple and individual inertial measurement units. *Sports biomechanics*.

[29] Pasos-Ruiz, A.; Flynn, M.; and Bagnall, A. 2020. Benchmarking Multivariate Time Series Classification Algorithms. *CoRR abs/2007.13156*

[30] Peng, X.; Wang, L.; Wang, X.; and Qiao, Y. 2014. Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice.

[31] Sánchez, J.; Perronnin, F.; Mensink, T.; and Verbeek, J. J. 2013. Image Classification with the Fisher Vector: Theory and Practice. *Int. J. Comput. Vis.*.

[32] Savitzky, A.; and Golay, M. J. E. 1964. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*.

[33] Schäfer, P.; and Högqvist, M. 2012. SFA: A Symbolic Fourier Approximation and Index for Similarity Search in High Dimensional Datasets. In *ACM EDBT 2012*.

[34] Schäfer, P.; and Leser, U. 2017. Multivariate Time Series Classification with WEASEL+MUSE. In *CIKM 2017*.

[35] Selvaraju, R. R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; and Batra, D. 2016. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization.

[36] Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *ECCV 2016*.

[37] Simon, T.; Joo, H.; Matthews, I.; and Sheikh, Y. 2017. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. In *CVPR 2017*.

[38] Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.

[39] Simonyan, K.; and Zisserman, A. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *NeuRIPS 2014*.

[40] Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild.

[41] T. Nguyen, T. L. N.; and Ifrim, G. 2020. A Model-Agnostic Approach to Quantifying the Informativeness of Explanation Methods for Time Series Classification. In *AALTD, ECML-PKDD 2020*.

[42] Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE ICCV 2015*.

[43] Wang, H.; and Schmid, C. 2013. Action Recognition with Improved Trajectories. In *IEEE ICCV 2013*.

[44] Wang, Z.; Yan, W.; and Oates, T. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *IEEE IJCNN 2017*.

[45] Whelan, D.; O'Reilly, M.; Huang, B.; Giggins, O.; Kechadi, T.; and Caulfield, B. 2016. Leveraging IMU data for accurate exercise performance classification and musculoskeletal injury risk screening. In *IEEE EMBC 2016*.

[46] Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. In *IEEE CVPR 2016*.

[47] Lingfei Wu, Ian En-Hsu Yen, Jinfeng Yi, Fangli Xu, Qi Lei, and Michael Witbrock. Random warping series: A random features method for time-series embedding. In *AISTATS 2018*

[48] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In *NeurIPS 2019*

[49] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV 2018*

[50] Noureldien Hussein, Efstratios Gavves, and Arnold W. M. Smeulders. Timeception for complex action recognition. In *IEEE CVPR 2019*

[51] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees G. M. Snoek. Videolstm convolves, attends and flows for action recognition. *Comput. Vis. Image Underst.*, 2018.

[52] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE CVPR 2019*

[53] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015.