Reference Method for the Development of Domain Action Recognition Classifiers: The Case of Medical Consultations

Sabine Molenaar(✉), Laura Schiphorst, Metehan Doyran, Albert Ali Salah, Fabiano Dalpiaz, and Sjaak Brinkkemper

Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
{s.molenaar,l.a.f.schiphorst,m.doyran,a.a.salah, f.dalpiaz,s.brinkkemper}@uu.nl

Abstract. Advances in human action recognition and interaction recognition enable the reliable execution of action classification tasks through machine learning algorithms. However, no systematic approach for developing such classifiers exists and since actions vary between domains, appropriate and usable datasets are uncommon. In this paper, we propose a reference method that assists non-experts in building classifiers for domain action recognition. To demonstrate feasibility, we instantiate it in a case study in the medical domain that concerns the recognition of basic actions of general practitioners. The developed classifier is effective, as it shows a prediction accuracy of 75.6% for the medical action classification task and of more than 90% for three related classification tasks. The study shows that the method can be applied to a specific activity context and that the resulting classifier has an acceptable prediction accuracy. In the future, fine-tuning of the method parameters will endorse the applicability to other domains.

Keywords: Human action recognition · Interaction recognition · Reference method · Method engineering · Machine learning · Domain action recognition classifier · Computer vision

1 Introduction

Machine Learning (ML) and Artificial Intelligence (AI) have been introduced to many different industries and fields of research to automate many tasks, including the recognition and classification of human actions. Large, context-specific datasets are needed to train, validate and test the classifiers, but not every available dataset can be used for every purpose [24]. When a specific classification task needs to be executed, chances are that no relevant dataset exists.

We focus on human action recognition: the classification problem of "*labeling videos containing human motion with action classes*" [20]. Thanks to the advancements in action recognition, researchers are now able to analyze more

S. Nurcan et al. (Eds.): BPMDS 2020/EMMSAD 2020, LNBIP 387, pp. 375–391, 2020.
https://doi.org/10.1007/978-3-030-49418-6_26

complex tasks such as human-human interaction recognition, which considers actions between two or more subjects, rather than the movement of a single subject. Additional challenges exist regarding (i) how to distinguish multiple subjects, (ii) subjects who (partially) block each other, and (iii) the lack of large datasets for the different contexts [24]. For this kind of classifier, no systematic method exists for their development, training, and validation.

To overcome this gap, we introduce a reference method for the development of classifiers for actions and interactions in a particular domain, including a subprocess for the creation of a suitable dataset: the DARC-method. Our aim is to increase the maturity of action recognition processes through the proposal of a reference method that can be used by people who have limited expertise in ML, such as information systems engineers. Besides providing an easy-to-follow process, our method provides links to literature in the field that a user may want to check to customize the method for the specific case at hand.

We describe our research method in Sect. 2 and discuss related work in Sect. 3. We present the reference method in Sect. 4. We demonstrate its feasibility in Sect. 5, by applying it to healthcare through the use of videos that record domain actions. Finally, in Sect. 6, we discuss validity threats, present our conclusions, and outline future work.

2 Research Method

Techniques, methods and processes for data analysis and ML already exist, but are not tailored to the specific purpose of developing domain action recognition classifiers[1]. Typically, ML literature assumes the reader has some knowledge or experience with ML. While they tend to have an implicit method, they predominantly explain how certain algorithms can be implemented and domain understanding is assumed when a case study is described. Therefore, we use an assembly-based method engineering approach, resulting in the following research question: "*How can a reference method be assembled for the development of domain action recognition classifiers?*"

Ralyté *et al.* [21] distinguish three main activities in their assembly-based process model: (i) specify method requirements, (ii) select method chunks and (iii) assemble chunks. In our case, the method requirements are as follows. First, the method should provide guidance to practitioners in information systems, rather than ML experts. Second, we focus only on action recognition classifiers. Third, we are concerned with classifiers for a given domain: the method should cover the entire process from domain understanding to deploying the classifier. However, the method should be domain-independent, i.e., applicable to any domain in which action recognition is used.

The reference method aims to provide a structured overview of the activities and deliverables and consistent terminology [27]. We hope our method mitigates the risk of introducing errors throughout the process. Also, since all activities

[1] Throughout this paper, 'action recognition' stands for both action and interaction recognition.

and deliverables have been predefined, there is a lower chance of accidentally omitting any steps. Following the method should also aid auditing whether the classifier was developed correctly and solves the problem at hand. The generic activities in the method are extracted from existing processes and frameworks, see Sect. 3. More detailed activities and the concepts are extracted from literature on this topic and assembled and described in Sect. 4.

3 Related Works

To the best of our knowledge, no systematic approach exists that describes the development of action recognition classifiers from problem statement to deployment. Thus, we start from data science frameworks and assemble a reference method for building effective classifiers for action recognition. We compare three processes: the Common Task Framework (CTF) [6], the Knowledge Discovery in Databases (KDD) [7] process, and the CRoss Industry Standard Process for Data Mining (CRISP-DM) [28].

CTF is meant for predictive modeling, making it suitable for classifier development. However, it only includes three main elements according to Donoho [6]: (i) a publicly available training dataset with feature measurements and labels, (ii) competitors that infer class prediction rules from the data and (iii) a referee that receives the predictions compares them to the test data and returns the prediction accuracy. This framework, however, requires an existing dataset. In addition, competing teams are needed to conduct the common task, which may not be readily available and/or willing to participate.

The KDD process describes the following steps: data selection, preprocessing, transformation, data mining and interpretation/evaluation. The process was designed for use in the data mining field [7]. KDD differentiates itself by focusing on the entire knowledge discovery process. Unfortunately, the process does not start with a specific problem that needs to be solved. Moreover, it does not address deployment of the process in another system, since the result of the process is knowledge that can be used.

Thirdly, CRISP-DM is selected, since it was designed for projects with large amounts of data and achieving specific business-related objectives. The CRISP-DM process model prescribes six phases [28]:

1. Business understanding: involves understanding the business requirements and objectives and formulating a data mining problem from this knowledge;
2. Data understanding: includes data collection, familiarization and data quality assessment;
3. Data preparation: focuses on converting the raw data into a usable dataset through cleaning and attribute selection among others;
4. Modeling: selection and use of modeling techniques, determining parameters for optimal results;
5. Evaluation: evaluate whether the selected techniques sufficiently address the initial problems and objectives;

6. Deployment: presenting the results in such a way that the customer can use them, for instance by writing a report.

The application to specific domains requires the classifier to be trained for each domain, which requires the method to be able to handle large amounts of data. Furthermore, the classifier should be deployed in the domain context or system so that it can be used for prediction tasks by the stakeholder(s). Therefore, the CRISP-DM process is used as a starting point, as it fits best the requirements for the reference method.

4 Reference Method for Domain Action Recognition Classifiers

The reference method for the development of Domain Action Recognition Classifiers (DARC-method) is visualized in a Process Deliverable Diagram (PDD) in Fig. 1. A PDD describes a method in a process side on the left based on UML activity diagrams, and the resulting (indicated with dashed arrows) concepts and deliverables on the right based on UML class diagrams. Open concepts are described via aggregation relationships in the PDD, while the elaboration of closed activities and concepts has to be obtained elsewhere [26]. Based on the CRISP-DM process model, the reference method consists of six consecutive phases with adapted names and purposes:

1. **Domain understanding:** documentation related to the domain needs to be gathered, in particular the actions the classifier should recognize. Also, the relevance and correctness of these actions should be discussed with domain professionals;
2. **Dataset creation:** if no datasets containing the relevant domain actions exist, it is necessary to create one (if a suitable dataset is available, this phase can be skipped);
3. **Dataset preparation:** videos in the dataset need to be prepared before they can be utilized for classifier training, testing and validation;
4. **Classifier modeling:** the classifier is modeled, the feature sets are created and the experimental protocol is determined;
5. **Classifier training:** the classifier is trained using the main part of the dataset, then validated using another part of the dataset, parameters are changed in an attempt to improve the performance (paying attention to avoiding overfitting) and finally the classifier is tested using the remaining data;
6. **Deployment:** once the classifier achieves good results, it needs to be deployed in the domain system, which changes per domain, and feedback from domain users can be taken into account for further development and/or improvement.

Firstly, in the **domain understanding** phase, information related to the domain is gathered. The CRISP-DM process model describes tasks such as 'business objectives' and 'determine data mining goals' [28]. In the context of action

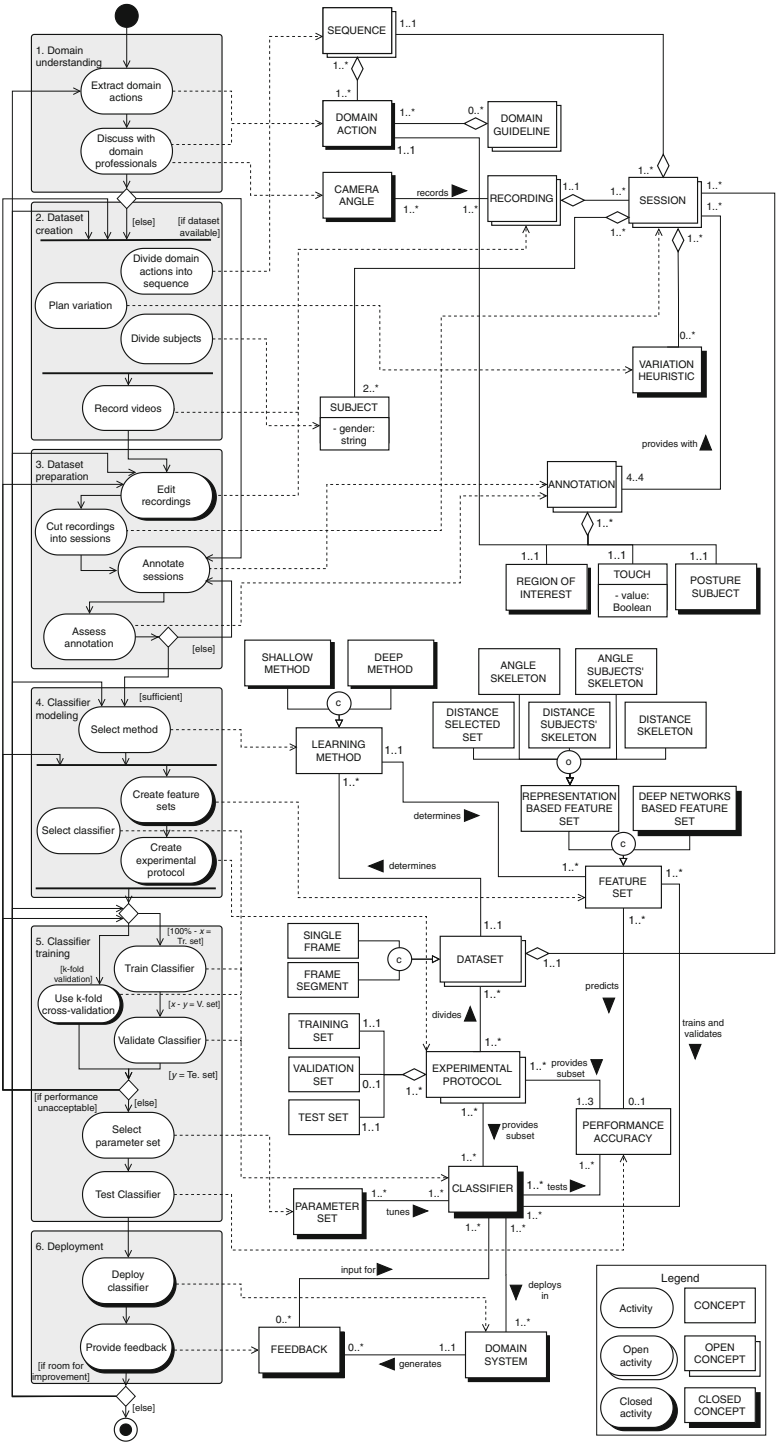

Fig. 1. Reference method for Domain Action Recognition Classifier (DARC) development.

recognition, these tasks can be translated to determining the domain objective: what action recognition problem or need should be addressed by the classifier? For the latter, the prediction goals can be determined, for example which actions the classifier should be able to classify. Domain actions that need to be recognized need to be identified, for instance by extracting them from available domain guideline repositories. In addition, domain professionals should give insight into which actions are performed most frequently, in which order they are performed and which camera angles might be needed to be able to record all actions to avoid occlusion. Intra-class variation and inter-class variation should be included in the actions, since subjects perform actions slightly differently every time and some actions may seem very similar to others and be classified incorrectly as a result. These variations should also include fluctuating duration, referred to as temporal variation [20]. Supporting variation is becoming easier: recent publicly available datasets for action recognition often include hundreds of actions, and datasets have become more realistic in terms of their settings [20]. However, **dataset creation** is needed if datasets containing actions related to a specific domain are not readily available, which is often the case [24].

Based on knowledge gathered in the first phase, the domain actions can be divided into sequences. For instance, actions that are nearly always performed together will be recorded in sequence, along with variations and combinations of other actions. Previously developed algorithms have, at times, been trained on segmented videos, meaning that the annotations have clear boundaries. In case of real-time action recognition, such boundaries are less evident. The classifier should be able to detect when an action starts and ends, also referred to as action spotting [4], and then classify said action. Therefore, the videos, with or without sequences, should not exclusively include actions [20].

In addition, to mitigate the risk of the classifier focusing on specific, irrelevant features, dataset variation (i.e. videos) is needed. Variation can be introduced by using different locations, clothing, camera angles, hairstyles, etc. Also, different subjects should be used to ensure that the classifiers are not trained to recognize actions performed by men and not by women or vice versa. The collection of these types of variation are referred to as *variation heuristic*s. In addition, multiple subjects need to be distinguished. The subjects may occlude each other, which is why multiple camera angles should be used to decrease the risk of occlusion [19]. The setting (fore- and back-ground) plays an important role in dataset creation too. Lighting conditions and additional movement in the recordings can affect the accuracy of the classifier. It is, however, ill-advised to create a 'clean' and static environment, since this is not comparable to the real-world context [20]. Variations in location may be desirable. When variation plans have been made, subjects have been divided and actions and sequences have been determined, the videos can be recorded. The minimal amount of data is difficult to ascertain, since this may vary between classification tasks, but more is nearly always better. Finally, statistical sparsity is a valid dataset quality concern. Even large datasets may be lacking in their inclusion of varied scenarios [8].

In the data preparation phase, the CRISP-DM model prescribes data cleaning and formatting. To prepare for supervised learning, the instances (actions) in the data should be labeled [13]. So, after recording, the videos should be prepared for training in the **dataset preparation** phase. If multiple cameras were used, the videos should be edited, such that they have the same dimensions and are uniform in terms of brightness, frame rate, etc. This is done to ensure that the variations in formatting do not affect the predictions. Besides, data augmentation can be applied during the editing activity, which means adding noisy versions of existing data to increase the size of the dataset. The augmentation approach depends on the data. For instance, when working with imagery, it is possible to mirror the existing images to create additional data. Alternatively, 3D synthesis can be utilized to generate synthetic data from real data, also with the intent of increasing the volume of data [25]. Then, recordings need to be cut into different sessions, since recordings can span over multiple sessions (to save time while recording). Subsequently, they can be annotated (supplied with the correct labels), so that the ground truth is available for training the classifier.

For the annotation, we distinguish four different labels: posture of the subject, region of interest, domain action and touch. Since the former three should be defined for a specific domain, they have not been extended with specific options. According to Moeslund *et al.* [16], concepts like action, activity and behavior are sometimes considered synonyms. In this method, however, we adopt their action taxonomy: (i) action primitives, which are atomic entities that comprise an action, (ii) actions, a set of action primitives that are needed to perform the particular action and (iii) activities, a collection of actions that describe a larger event. For example, 'playing tennis' is an activity, that contains action such as 'serve' and 'return ball', in which the latter consists of action primitives like 'forehand', 'run right' and 'jump' [16]. Preferably, the quality of the annotations is assessed. As an example, Mathias *et al.* [15] illustrated that re-annotating data with strict and consistent rules, as well as adding 'ignore' tags to unrealistically difficult samples, can have a significant impact on the assessment of classifiers. Incorrect annotations counted as errors created an artificial slope on the assessment curves and biased the results. Multiple labels for individual items, also referred to as repeated labeling, was also proven to be valuable [23].

Subsequently, in the **classifier modeling** phase, a learning method is selected. Kong and Fu distinguish two types of methods: *shallow* and *deep*. Action recognition predominantly makes use of deep methods, with shallow methods being better suitable when small amounts of training data are available [12]. Herath *et al.* divide action recognition in two similar categories, namely representation-based solutions and deep networks-based solutions [10]. For the former, some sort of representation, e.g., keypoints, silhouettes, 2D/3D models, are required in order to train, validate and test the classifier. If a deep method is selected, there are three different options: (i) if a large, labeled dataset is available, it is possible to perform end-to-end training of a deep neural net, (ii) if the dataset has some labeled data, data augmentation or transfer learning can be applied, i.e., employing large datasets to train deep networks and then using

intermediate layer representations as deep features for a different classification task [24] and (iii) if there is little labeled data available, active learning can be applied to select and annotate additional, relevant data to increase the size of the dataset [17].

Feature sets need to be extracted in order to recognize actions. Herath *et al.* distinguish between methods that are based on handcrafted features and those that use deep learning based techniques [10]. The former are referred to as representation-based feature sets in the method. An example of a spatial feature set is provided in Fig. 1, which includes distance between the keypoints within the person for all subjects, distance between keypoints of one subject and another, among others. Deep network solutions, on the other hand, implicitly extract features from the input data, which means they cannot be determined a-priori and are therefore not specified in the method. In addition, one or multiple classifiers need to be selected for the classification task. Multiple classification methods (e.g., Naive Bayes, Random Forest, k-Nearest Neighbor) can be trained to see which one yields the best results. Given that not every classifier may be appropriate for every classification task or dataset and classifier selection may be influenced by the available time and computing power, no classifiers are prescribed in the method.

To validate and test the classifier, we either rely on standard techniques or use k-fold cross-validation. In case of the former, the dataset is divided into subsets to be used for training, validation and testing the classifiers, these subsets are used in the training phase. To avoid overfitting on a specific angle (if multiple cameras are used), subsets should be created based on actions, rather than individual sessions. Overfitting can be described as the risk of "*memorizing various peculiarities of the training data rather than finding a general predictive rule*" [5]. If this occurs, the classifier might perform well on training data and less so on unseen data. Alternatively, k-fold cross-validation splits a single dataset into a training set (known data) and test set (unknown data) k times. For instance, if $k = 3$, the classifier is trained and tested on three different training and test sets, which are derived by slicing the dataset in different ways.

Furthermore, the dataset can be fed to the classifier using single frames or frame segments. Some domain actions may appear similar or nearly identical when single frames are considered, but can be distinguished when multiple frames are shown. Therefore, the classifier can be provided with multiple frames at once, for instance 30. If within those 30 frames 20 are identified as a specific action, all 30 frames will receive that label. The length of the sliding window (or skip length) refers to the number of frames the window moves through time (temporal direction) before a new segment is started [2].

The **classifier training** phase uses the subsets specified in the experimental protocol and the created feature sets to train, validate and test. The classifiers are trained and validated using several parameter sets. If parameters are only optimized for the training data, there is a risk of overfitting [5]. In case of limited data availability, training starts from a classifier trained for a similar, related task (i.e. transfer learning). Alternatively, using k-fold cross-validation, training and

testing are conducted k times. After training and validation is completed using all selected parameter sets, the parameter set that yields the best performance is selected for testing the classifiers. No specific parameter sets are described, since these may vary between domains. It should be noted that the subsets of the dataset should under no circumstances overlap. However, it is possible to join the training and validation sets for a final round of training before reporting results on the test set. In some cases, dataset creation and classifier training are performed iteratively, until a certain accuracy level is reached. If the accuracy level on the validation set is unacceptable, there are four options: (i) re-train the classifier using k-fold cross-validation, for example, (ii) creating different feature sets and/or selecting a different classifier, (iii) editing or augmenting the recordings in order to increase the size of the dataset or creating synthetic data and (iv) creating additional data from scratch.

If the classifier is sufficiently accurate, it can be used in its intended domain during the **deployment** phase. If not, the classifier should not be considered for deployment. Instead, an attempt can be made to identify more appropriate feature sets or a better suiting classifier. Context systems are systems used in the specific domain. The results of the classifiers might need to be included in an existing system, for instance. Finally, end-users of the classifier might want or need to provide feedback on its performance. If users notice that one specific action is often classified as another, additional data and training might be needed. This is in accordance with the monitoring and maintenance plan task included in the CRISP-DM process [28]. Since feedback cannot be predicted beforehand, it is assumed all phases in the process can be affected.

5 Case Study: Medical Consultations

The DARC-method was applied to a case study within the context of the Care2Report research program regarding the automated reporting of medical consultations [14]. The purpose of the case study is twofold: (i) to assess the feasibility of the method by demonstrating how the activities and deliverables can be applied, and (ii) to evaluate the effectiveness, by reporting on the prediction accuracy of the resulting classifier.

The method was applied to the healthcare domain, since General Practitioners (GPs) generally experience a high workload. A classifier that is able to recognize medical actions performed in a GP's office can support the administrative work of GPs, not by diagnosing disease or by detecting anomalies, but by serving the medical reporting of the consultation. The classifier and its results are also discussed in [22].

1. Domain understanding. In the Netherlands, GPs make use of clinical guidelines and standards developed by the Nederlands Huisartsen Genootschap[2] (transl. Dutch General Practitioners Society). First, all medical actions that occur in the clinical guidelines were extracted. Then, a selection was made based

[2] https://www.nhg.org/nhg-standaarden.

on the instruments available for this study and whether the actions are of a sensitive nature. For the former, actions such as performing an ECG were excluded, since we did not possess such instruments. For the latter, actions that require subjects to (partially) undress were excluded to preserve the privacy of subjects participating in the recordings. The following medical actions were included in the dataset, in decreasing occurrence order in the guidelines [22]: Blood pressure measurement (BPM), Palpation abdomen (PaA), Percussion abdomen (PeA), Auscultation lungs (AL), Auscultation heart (AH), and Auscultation abdomen (AA).

In addition, there is a 'no action' class, in case no action is occurring in a segment of the video. We also distinguish the following classes: 'sitting upright', 'laying down' and 'laying down with knees bent' (posture of patient), whether the GP touches the patient or not (distance to patient) and 'arm', 'chest', 'upper back' and 'abdomen' (region of interest). Actions, however, are not always performed in isolation, but may be part of a sequence. In discussion with medical professionals, and taking into account the guidelines, the most frequently occurring combinations and order within those combinations were used most often, with some slight variation to mitigate overtraining on specific sequences. Using the previously mentioned taxonomy [16], we distinguish the following examples: (i) 'pressing with the hand' and 'releasing pressure of the hand' as action primitives, (ii) 'palpation of the abdomen' as action and (iii) 'physical examination' or 'medical consultation' as activity.

2. Dataset creation. The videos [22] were recorded with four subjects (three female, one male), who all played the roles of both GP and patient. Since GPs examine one patient at a time, exactly two subjects appear in all videos. In 68% of the videos, both subject were female, in 15% only the GP was female and in 17% only the patient was female. In addition, they all changed clothes, hair and jewelry. The GP is required to wear their hair up, but the patient can have any hairstyle. The same is true for jewelry, since GPs are not allowed to wear any jewelry while patients are. It should also be noted that, in the Netherlands, GPs rarely wear white coats, which is why the GP role also changed clothing throughout the recordings. Glasses were worn by both GP and patients. During examinations, the patient is either sitting upright, laying down flat or laying down with their knees bent. The GP performed actions from either side of the patient and there is variation within actions. For instance, when listening to the lungs, sometimes they started on the right and sometimes on the left. Stethoscopes were used during recordings, both a red and black one to introduce variation. Finally, three cameras from three different angles were used, which is visualized in [22].

Since the GP (subject A) only performs medical actions using their hands, the lower half of their body is not of importance and can be hidden behind the examination table or gurney. The GP may have to move around to perform the appropriate medical actions, which is why three cameras were used. By placing them in different positions and thus acquiring three different angles, the chances of occlusion are reduced. A total of 451 videos were recorded, 73.6% of which

included a single action (those videos contained frames without any action as well), the rest contained sequences [22].

3. Dataset preparation. All recordings were cut into sessions, 192 in total [22], which were then annotated, meaning they were provided with the right labels. The online annotation tool ELAN [9] was used, since it allows for annotating multiple videos concurrently. An example of how part of a video is annotated is: 'sitting upright' (posture subject), 'arm' (region of interest), 'true' (touch) and 'BPM' (domain action).

4. Classifier modeling. In this case a representation-based solution was used, due to the small amount of data and since the human body structure and its movements are the focus of the classifier [10]. Either keypoints (also referred to as skeleton joints) can be extracted or images or videos can be used as input for the classifier [11]. Since distances between the subjects are important, to recognize interaction, and actions can be defined using distances and angles, keypoints are used. Information such as colors and background is present in images and videos, but is not required to recognize domain actions. Keypoints are used to extract a skeletal representation from the subjects in the videos. These representations can also be used to reduce the dimensionality of the dataset, by determining the minimum and maximum values of the feature set, as well as the average and variance, of multiple frames. This results in improved computational efficiency and accuracy [3].

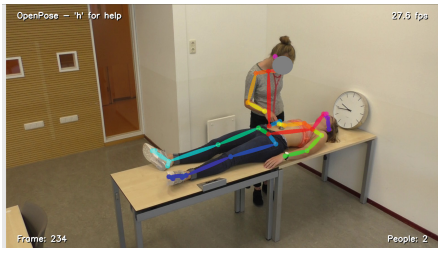

Fig. 2. 2D skeleton generated by OpenPose with patient laying down.

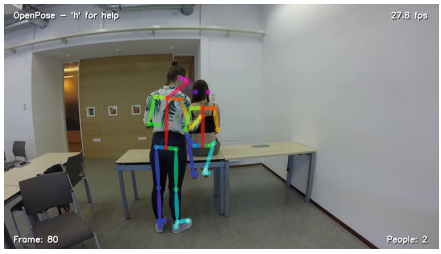

Fig. 3. 2D skeleton generated by OpenPose with patient sitting up. (Color figure online)

OpenPose was applied to all sessions to extract keypoints, since it is publicly available and able to recognize multiple subjects [1]. Examples of 2D skeletons generated by Openpose are depicted in Figs. 2 and 3. Note that in Fig. 2 the GP's face is blurred for privacy reasons. These keypoints are then utilized to create mathematical representations of the subjects and to determine the distances between skeleton joints and angles that each line between two neighbor

joints makes with the horizontal axis of each camera viewpoint [29,30]. We calculate angles differently from previous literature, where they use angles of two lines between three neighbor joints. Our approach on computing angles allows us to embed the body orientation of each subject both relative to camera viewpoint and therefore to each other as well. The 2D joint coordinates (Eq. 1) are determined in order to calculate the angles and distances:

$$J_c = J_c(J) = (J_x, J_y) \tag{1}$$

We use J for joint and J_c for 2D joint coordinate where J_x and J_y represent the x and y axis, respectively. Secondly, the angle between a line of two neighbor joints and the horizontal axis (Eq. 2) is calculated as follows [18]:

$$LL_a = LL_a(L_{J_1 \to J_2}, L_h) = arctan(\frac{J_{1_x} - J_{2_x}}{J_{1_y} - J_{2_y}}) \tag{2}$$

LL_a, $L_{J_1 \to J_2}$, L_h represent line-line angle, line between two joints, and horizontal line, respectively. Finally, the (Euclidean) distance between two or more joints of one subject and the distances between the joints of two or more subjects are determined (Eq. 3) using the following equation:

$$JJ_d = JJ_d(J_1, J_2) = \overrightarrow{||J_1 J_2||} = \sqrt{(J_{1_x} - J_{2_x})^2 + (J_{1_y} - J_{2_y})^2} \tag{3}$$

In Fig. 3 an example of how Eq. 3 is calculated can be seen. The distance between the right hand of the GP (in yellow) and the back of the patient (in red) can be used to determine whether the GP is touching the patient and which medical action (in this case AL) is being performed. The calculation of angles and distances was done as defined in Eqs. 1, 2 and 3.

Considering the medical actions, experiments were conducted using five different feature sets, to determine which set(s) offered the most valuable information. The following sets of features were included in experimentation:

Feature set 1: Pre-selected group of features: (i) angle between the neck and mid-hip of both subjects, (ii) distances between both hands of subject A and a specific body part (i.e. chest, abdomen, arm, left hand and right hand) of subject B and vice versa;
Feature set 2: Distances between all keypoints within subjects;
Feature set 3: Angles of keypoints relative to other keypoints within subjects;
Feature set 4: Distances between both hands of subject A and the upper body of subject B and vice versa;
Feature set 5: Angle of the hands of subject A relative to the upper body of subject B and vice versa.

Note that in these feature sets, the upper body is considered to range from the head to the lower abdomen (right above the keypoints of the hips).

5. Classifier training. We selected Random Forest (RF) to be used in the medical action recognition. Some experimentation was done with both k-Nearest Neighbors and Decision Trees, but results are left out for the sake of brevity. The classifiers were trained using 60% of the dataset (i.e. sessions). Then, 20% of the remaining 40% was used to validate the classifier. Based on the validation results, the parameter set with the highest accuracy was selected. Finally, the last 20% was used to test the classifiers. For all three subsets, the training set contained 60% of the sessions, distributed evenly over available angles. The RF classifier was trained, validated and tested on the aforementioned feature sets, as well as combinations of the sets, resulting in a predication accuracy of 0.697. An example of distance parameters that were used to classify the medical actions is shown in Fig. 4. Note that these distance parameters were fixed, but the decision tree was not, due to the nature of RF classifiers.

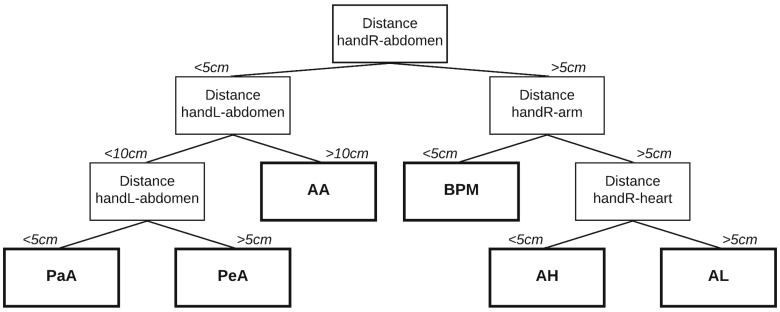

Fig. 4. Example of distance parameters in a decision tree.

At first, single frames were used to recognize actions. However, some medical actions are quite similar when comparing individual frames rather than a segment of a video, because they are performed on the same region of interest. For instance, during palpation of the abdomen, both hands are pressed on the abdomen, while during percussion one hand is not released from the abdomen, while the other is. Therefore, percussion is easily confused with palpation. The same is true for auscultation of the lungs and heart. In case of the heart, only the area of the chest around the heart is covered, while in case of the lungs the entire chest is examined. The two are more difficult to distinguish when only individual frames are considered. When taking into account frame segments of videos as opposed to single frames, the accuracy of the classifier increases by 0.059, using 120 frames and a sliding window of 20 frames, to 0.756.

The use of segments increased the accuracy of the classifier when distinguishing palpation and percussion of the abdomen and auscultation of the lungs and heart. Confusion matrices illustrating the prediction accuracy of the best performing feature set using both individual frames and segments of the classifier are shown in Figs. 5 and 6, respectively. While the prediction accuracy does not increase for all individual actions, the average prediction accuracy does. The

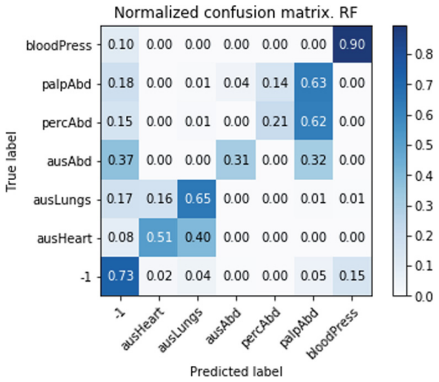

Fig. 5. Confusion matrix of feature set with best test accuracy (sets 3–5) of RF classifier.

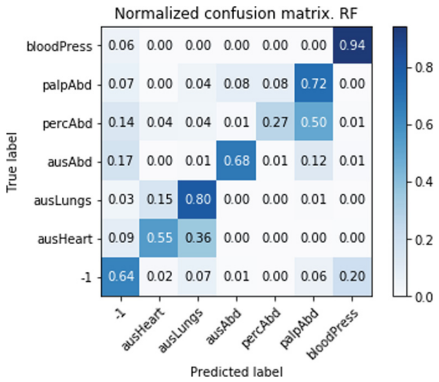

Fig. 6. Confusion matrix of best performing segment (120, 20) for RF (feature sets 3–5).

prediction accuracy of the other annotations, i.e. posture of patient, distance to patient and region of interest are 0.996, 0.910 and 0.908, respectively, using feature sets 3, 4 and 5 and frame segments of 30 frames with a sliding window of 15 frames. In [22] we report on the results of a 4-fold cross-validation.

6. Deployment. The Care2Report (C2R) system, first presented by Maas *et al.* [14], strives for fully automated medical reporting. Currently, the system is able to generate medical reports adhering to clinical guidelines by using audio of a consultation as input. However, the consultation with a patient often includes a physical examination as well, which requires video for analysis. The recorded medical actions need to be identified automatically, for which the described classifier can be used. The goal for the C2R system is recognition of medical actions performed by GPs in real-time, so that its results complement the report generated from the audio input. Therefore, the deployment context of the classifier is a GP's office. In addition, it will have to interact with the electronic medical record in the C2R system, but this is left to future work.

6 Discussion

In this paper, we presented the DARC-method, a reference method for developing classifiers for domain action recognition, based on existing methods and techniques. We then applied the method to a case study and reported on the prediction accuracy of the developed classifier.

Validity Threats. Firstly, the DARC-method was only applied to a single case and a single domain; it may need customization when applied to other domains. Secondly, the method was applied by people with some experience with both

ML and classifiers and the specific domain it was used for. Additionally, we were unable to conduct the deployment phase as of yet, meaning this part was never tested in a real-world situation and cannot be described in detail. Given that there are additional steps to developing the classifier, there is a risk of introducing errors throughout the process [24]. Finally, others less familiar with the subject matter and process may have a more difficult time applying the method to their own case and/or domain. However, the method is based on existing and validated techniques, which should improve its external validity.

Conclusion. Our research question "*how can a reference method be assembled for the development of domain action recognition classifiers?*" was answered by designing a reference method using a method assembly approach, combining existing methods and techniques. We expect the method to provide assistance, in line with the first requirement, when developing a classifier, by providing a step-wise process, related literature and consistent terminology. In accordance with the second requirement, the method is tailored to action recognition learning methods and relies on video input. Also, the DARC-method introduces algorithmic restrictions by defining a standard set of representation-based features. The domain understanding and deployment phases support the third requirement. The developed classifier proves that the reference method can be applied to a case in the medical domain. Thanks to the method, the decisions made for each activity in the process were reported in a structured manner. The classifier was trained to perform four different classification tasks: identifying medical actions, the posture of the patient, the distance of the GP to the patient and the region of interest. The prediction accuracy of the tasks are 75.6%, 99.6%, 91.0% and 90.8% respectively.

Future Work. In order to evaluate the effectiveness of the method an experiment using students as test subjects will be conducted. Students with little to no experience with action recognition will be divided into an experimental and a control group. The former will make use of the method, while the latter will use no method. The effectiveness of the method can then be assessed by comparing the efficiency, number of errors and quality of the resulting work of both groups. Next steps include the classifier being trained using different datasets for additional contexts and purposes. For instance, the method and its resulting classifier can be applied to the orthopedics specialization. Thanks to the use of imagery, we should be able record a patient's range of movement over a period of time and analyze the data. Placing the method in a broader perspective, classifiers can be developed and trained for use in other contexts, such as police reporting. Furthermore, the deployment of the trained classifier in the automated reporting system C2R should be investigated to evaluate the final phase of the method.

References

1. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: real-time multi-person 2D pose estimation using part affinity fields. arXiv preprint arXiv:1812.08008 (2018)
2. Colleoni, E., Moccia, S., Du, X., De Momi, E., Stoyanov, D.: Deep learning based robotic tool detection and articulation estimation with spatio-temporal layers. IEEE Robot. Autom. Lett. **4**(3), 2714–2721 (2019)
3. Cunningham, P.: Dimension reduction. In: Cord, M., Cunningham, P. (eds.) Machine Learning Techniques for Multimedia. COGTECH, pp. 91–112. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-75171-7_4
4. Derpanis, K.G., Sizintsev, M., Cannons, K., Wildes, R.P.: Efficient action spotting based on a spacetime oriented structure representation. In: Proceedings of the CVPR, pp. 1990–1997. IEEE (2010)
5. Dietterich, T.: Overfitting and undercomputing in machine learning. ACM Comput. Surv. (CSUR) **27**(3), 326–327 (1995)
6. Donoho, D.: 50 years of data science. J. Comput. Graph. Stat. **26**(4), 745–766 (2017)
7. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: The KDD process for extracting useful knowledge from volumes of data. Commun. ACM **39**(11), 27–34 (1996)
8. Gudivada, V., Apon, A., Ding, J.: Data quality considerations for big data and machine learning: going beyond data cleaning and transformations. Int. J. Adv. Softw. **10**(1), 1–20 (2017)
9. Hellwig, B.: EUDICO linguistic annotator (ELAN) version 1.4-manual. Last updated (2003)
10. Herath, S., Harandi, M., Porikli, F.: Going deeper into action recognition: a survey. Image Vis. Comput. **60**, 4–21 (2017)
11. Jordan, M.I., Mitchell, T.M.: Machine learning: trends, perspectives, and prospects. Science **349**(6245), 255–260 (2015)
12. Kong, Y., Fu, Y.: Human action recognition and prediction: a survey. arXiv preprint arXiv:1806.11230 (2018)
13. Kotsiantis, S.B., Zaharakis, I., Pintelas, P.: Supervised machine learning: a review of classification techniques. Emerg. Artif. Intell. Appl. Comput. Eng. **160**, 3–24 (2007)
14. Maas, L., et al.: The Care2Report system: automated medical reporting as an integrated solution to reduce administrative burden in healthcare. In: Proceedings of the 53rd HICSS (2020)
15. Mathias, M., Benenson, R., Pedersoli, M., Van Gool, L.: Face detection without bells and whistles. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 720–735. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_47
16. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Comput. Vis. Image Underst. **104**(2–3), 90–126 (2006)
17. Nath, T., Mathis, A., Chen, A.C., Patel, A., Bethge, M., Mathis, M.W.: Using DeepLabCut for 3D markerless pose estimation across species and behaviors. Nat. Protoc. **14**(7), 2152–2176 (2019)

18. Noori, F.M., Wallace, B., Uddin, M.Z., Torresen, J.: A robust human activity recognition approach using OpenPose, motion features, and deep recurrent neural network. In: Felsberg, M., Forssén, P.-E., Sintorn, I.-M., Unger, J. (eds.) SCIA 2019. LNCS, vol. 11482, pp. 299–310. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20205-7_25
19. Park, S., Trivedi, M.M.: Understanding human interactions with track and body synergies (TBS) captured from multiple views. Comput. Vis. Image Underst. **111**(1), 2–20 (2008)
20. Poppe, R.: A survey on vision-based human action recognition. Image Vis. Comput. **28**(6), 976–990 (2010)
21. Ralyté, J., Deneckère, R., Rolland, C.: Towards a generic model for situational method engineering. In: International Conference on Advanced Information Systems Engineering, pp. 95–110 (2003)
22. Schiphorst, L., Doyran, M., Salah, A.A., Molenaar, S., Brinkkemper, S.: Video2report: a video database for automatic reporting of medical consultancy sessions. In: 15th IEEE International Conference on Automatic Face and Gesture Recognition, Buenos Aires (2020)
23. Sheng, V.S., Provost, F., Ipeirotis, P.G.: Get another label? Improving data quality and data mining using multiple, noisy labelers. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 614–622 (2008)
24. Stergiou, A., Poppe, R.: Analyzing human-human interactions: a survey. Comput. Vis. Image Underst. **188**, 102799 (2019)
25. Varol, G., et al.: Learning from synthetic humans. In: Proceedings of the CVPR, pp. 109–117 (2017)
26. van de Weerd, I., Brinkkemper, S.: Meta-modeling for situational analysis and design methods. In: Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, pp. 35–54. IGI Global (2009)
27. van de Weerd, I., de Weerd, S., Brinkkemper, S.: Developing a reference method for game production by method comparison. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) Situational Method Engineering: Fundamentals and Experiences. ITIFIP, vol. 244, pp. 313–327. Springer, Boston, MA (2007). https://doi.org/10.1007/978-0-387-73947-2_24
28. Wirth, R., Hipp, J.: CRISP-DM: towards a standard process model for data mining. In: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, pp. 29–39 (2000)
29. Yun, K., Honorio, J., Chattopadhyay, D., Berg, T.L., Samaras, D.: Two-person interaction detection using body-pose features and multiple instance learning. In: Proceedings of the CVPRW (2012)
30. Zhang, S., Liu, X., Xiao, J.: On geometric features for skeleton-based action recognition using multilayer LSTM networks. In: Proceedings of the WACV, pp. 148–157. IEEE (2017)