2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Extracting Action Hierarchies from Action Labels and their Use in Deep Action Recognition

Konstantinos Bacharidis
Computer Science Department, University of Crete, and
Institute of Computer Science, FORTH
Heraklion, Greece
Email: kbacharidis@csd.uoc.gr; kbach@ics.forth.gr

Antonis Argyros
Computer Science Department, University of Crete, and
Institute of Computer Science, FORTH
Heraklion, Greece
Email: argyros@csd.uoc.gr; argyros@ics.forth.gr

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412033

Abstract—**Human activity recognition is a fundamental and challenging task in computer vision. Its solution can support multiple and diverse applications in areas including but not limited to smart homes, surveillance, daily living assistance, Human-Robot Collaboration (HRC), etc. In realistic conditions, the complexity of human activities ranges from simple coarse actions, such as siting or standing up, to more complex activities that consist of multiple actions with subtle variations in appearance and motion patterns. A large variety of existing datasets target specific action classes, with some of them being coarse and others being fine-grained. In all of them, a description of the action and its complexity is manifested in the action label sentence. As the action/activity complexity increases, so is the label sentence size and the amount of action-related semantic information contained in this description. In this paper, we propose an approach to exploit the information content of these action labels to formulate a coarse-to-fine action hierarchy based on linguistic label associations, and investigate the potential benefits and drawbacks. Moreover, in a series of quantitative and qualitative experiments, we show that the exploitation of this hierarchical organization of action classes in different levels of granularity improves the learning speed and overall performance of a range of baseline and mid-range deep architectures for human action recognition (HAR).**

I. Introduction

Computer vision exhibits rapid advancement, benefiting from the developments in machine learning and, in particular, of deep learning approaches. One computer vision sub-field that has faced significant advancement is that of human activity recognition. This is considered as a challenging task due to the high-dimensionality of video data, appearance variability due to viewpoint changes, intra-class appearance variations, etc. In recent years, numerous deep learning-based approaches for action recognition have been proposed that benefit from the representational power and temporal modeling capabilities of deep neural networks (DNNs) [1]. These approaches can be grouped into two categories, on the basis of their end-to-end model learning capabilities. The first utilizes DNNs as frame-wise or segment-wise feature descriptors forwarding the representation to probabilistic [2], [3] or deterministic [4], [5] temporal models, separating the learning of spatial and temporal representations. On the contrary, the second class of methods employ end-to-end spatio-temporal DNN architectures that combine spatial feature extraction and temporal modeling in a unified framework. In both cases, the complexity and robustness of the temporal modeling depends on the complexity and temporal ordering variations of the modelled action/activity. A set of complex activities/actions require models capable of expressing both short- and long-term dependencies and variations. Usually, models with large state storing capacity and a large number of hyper-parameters are defined. However, for such models the learning process is hard and the adaptability to new activity sets is low, requiring the redefinition of the core model structure and a reassessment of the model's learning capacity to the new task [1].

The aforementioned issues become more challenging as the inter-class variability of the activities to be recognized decreases. High action complexity and low inter-class variability increases the complexity of the model designs since all possible dependencies and correlations among the classes in the feature learning and temporal modeling processes need to be considered. Instead of brute force solutions that employ models of increasing complexity, a common strategy is to exploit a priori knowledge from other information sources. One such source capable of providing knowledge about potential dependencies and correlations between action classes are the lexical descriptions of the actions as manifested in the action labels or script data. A key observation is that, in most cases, the size, the complexity and the semantic content of the label sentence or script size, increases with the action complexity. Moreover, the label contains implicit information on potential relationships between actions. Thus, distilling and comparing the semantic content of labels via natural language processing (NLP) methods can be helpful in extracting meaningful knowledge about action class attributes and similarity metrics that can assist in the class distinction.

In the present work, we propose a novel way to exploit NLP-driven knowledge on action class similarities extracted from their action labels, in order to define action hierarchies that represent coarse-to-fine action contents. Additionally, we show experimentally that this coarse-to-fine, two-level action hierarchy, when properly exploited by an action recognition DNN architecture, leads to faster learning and higher recognition accuracy scores for baseline designs, with potential benefits for state-of-the-art (SOTA) designs. Overall, the contributions of our work are the following:

- We present an NLP-based approach to identify and quantify similarities of action labels of different action classes, based on verb commonalities or similarities.
- We show that this verb-centered label grouping can be utilized to generate a two-level action hierarchy that allows the reformulation of the HAR problem into a coarse-to-fine multi-level activity classification problem.
- We provide DNN design directions for the utilization of the derived two-level granularity hierarchy in HAR-oriented deep model, illustrating the benefits of incorporating the learned coarse representation into the fine-grained representation learning problem.

The remainder of the paper is organized as follows. In Section II we review related work on action recognition, highlighting the novelties of the proposed approach. Section III describes the proposed methodology. Our approach is validated experimentally in Section IV. Finally, Section VI provides a summary of this work.

II. Related Work

Over the last couple of years, deep learning has dominated video-based human activity recognition. Convolutional Neural Networks (CNNs) are now considered a standard for spatial feature extraction due to their representational superiority over hand-engineered feature descriptors. HAR requires modeling of the temporal evolution of these spatial features. Thus, recently, HAR methods utilize temporal modeling neural network designs, such as Recurrent Neural Networks (RNNs).

Similarly to the hand-engineered HAR methods, the design complexity of deep learning-based methods is also interweaved with the activity complexity. Complex activity sets with low inter-class and high intra-class variations require DNNs with high representational capacity, leading to deep and wide DNN designs, with a variety of temporal convolution kernel sizes and memory mechanisms. To increase the information flow, recent methods exploit multi-modal DNN designs combining visual information with language, audio or other sensory data [6].

In particular, the linguistic analysis on script data has proven to be a useful information source for activity recognition that is utilized by both hand-engineered and deep learning-based methods [7], [8], [9]. Nevertheless, only a small portion of existing HAR datasets provide script data. Another source of linguistic activity-related information is the action/activity labels. Despite the small sizeof label sentences, they contain valuable information about motion motifs (expressed in the form of verbs), as well as about the presence of action-relevant objects (expressed in the form of nouns). Thus, a number of HAR approaches utilize the semantic content of action labels in their model design, in a variety of ways. A subset of methods [10], [11] highlighted the existence of potential semantic overlap between verb-centered labels in HAR datasets, due to different verb meaning interpretation by annotators and proposed joint multi-label classification and label correlation approaches, with the goal of exploiting label correlation as a complementary attribute for better activity distinction. Other works, utilize linguistic analysis of label sentences to derive semantic similarities. The similarity attribute is then used in the form of weights either for potential mis-classification penalization [12], or for cross-domain learning, to weight the source instances based on the similarity of labels of the source and target data [13].

Compared to these works, our work exploits label similarities to cluster activities based on motion motif commonalities, expressed in the form of verb-centered semantic similarities. Moreover, the derived label grouping is utilized to generate a two-level action hierarchy, enabling the reformulation of the HAR problem into a coarse-to-fine multi-task activity classification problem.

Coarse-to-fine and *multi-task* learning (MTL) for HAR, have already been exploited in the existing literature, in a number of hand-engineered (e.g. [14], [15]) and deep learning-based methods (e.g. [16], [17]). The main idea is to first aggregate feature representations of different granularities into a common representation, essentially learning latent tasks shared across action classes, and subsequently evaluate them on the fine-grained classification problem, without explicitly defining a coarse-grained label set. Regarding the deep learning-based approaches, the action granularity representation hierarchy is expressed with a series of sub-networks, one for every granualrity level. In the case of an uncropped input sequence the different representation levels are generated with sub-network parameter gradations [17], [18], where lighter/shallower networks lead to coarser representations and deeper/wider produce more fine-grained ones. Another way to achieve this is to utilize the different actor body part regions as inputs for the discrete sub-networks to produce coarse-grained representations and then use their aggregation to generate the fine-grained one [5], [16], [19]. In comparison to such DNN architecture designs, the proposed DNN design strategy follows the same principal as the first class, however, we allow for a direct evaluation of the learned multi-level granularity representations using the NLP-driven action hierarchy.

III. Proposed Method

The proposed approach consists of a two-stage action recognition DNN framework that exploits linguistic commonalities in action labels. In that direction, we define an action tree structure in which top levels express more abstract/coarse action classes that become more specific/fine-grained in bottom levels. We claim that redesigning the higher-levels of existing deep architectures to exploit this action hierarchy into a two-stage coarse-to-fine action classification task allows the network to learn faster, and to develop more robust representations for both coarser as well as for more complex activity sets. In more detail, the network functionality initially involves the extraction of estimates regarding the coarser action categories in an input sequence. The estimation is then refined by propagating the information deeper into the network with the goal of classifying the action into the set of complex action classes that have been grouped under each coarser action category. Intuitively, the idea is to let the network learn at its early levels feature representations that are capable of distinguishing distinctive classes. Then, at its subsequent layers, the network learns finer representations that enable it to distinguish actions that belong to the same coarser action category but have subtle differences.

A. Action tree hierarchy

The action tree hierarchy expresses the degree of similarity of the lexical descriptions of the action labels. We can construct a multitude of such tree structures by focusing each time on a particular part of the speech (e.g. verbs, nouns) or even part-of-speech combinations (e.g. verb + noun) increasing the semantic content being utilized. However, the semantic correlation of the classes that is expressed in each structure is different depending on the part of the speech or combination used. Typically, an action is largely characterized by the verb that is used to describe it in the linguistic representation. Moreover, the majority of existing complex activity datasets contain action labels that share common verbs and differentiate based on the objects used by users, which are expressed in the form of nouns. Thus, the most meaningful way to construct the action tree is to target verb-centered commonalities in action labels. To construct the tree, we first isolate the action verbs in the sentence using a combination of part-of-speech taggers and carefully defined syntax rules. Subsequently, we cluster the classes based on verb sharing or if the degree of similarity of the semantic content of the verbs is large (above a predefined threshold). Finally, we construct a square N-by-N binary matrix, with N denoting the number of fine-grained action labels. An entry of this matrix has the value 1 if the corresponding classes have a common motion motif expressed via verb commonality (or high verb semantic content similarity, explained in the forthcoming paragraphs) or 0, otherwise.

In more detail, following our previous work [12] on NLP-assisted label clustering, we utilize part of speech taggers trained on large sets of corpus readers (specifically, WordNet [20]) that are provided in the Natural Language ToolKit (NLTK) platform [21] to assign tags for each word. Moreover, to refine the number of potential verbs identified in each sentence, we define syntax rules regarding the verb position, the actual contribution of the verb in the semantic content and overall linguistic structure of an action label sentence. Specifically, the following grammatical rules are imposed to refine and correct the set of identified action verbs:

- **Syntax rule 1**: *A candidate verb can be followed by any number of particles (at, on, out, up, etc) or ad-positions (on, of, at, with, etc), delimiter or possibly by a noun.*
- **Syntax rule 2**: *If a verb is followed by a particle or an ad-position, then define the candidate verb as compound.*
- **Syntax rule 3**: *A sentence may start with a verb token. If, instead, a sentence starts with a noun token followed by another noun, ad-position or adjective, search the corpus if the starting word can be classified also as a verb. If yes, change the token label to verb.*

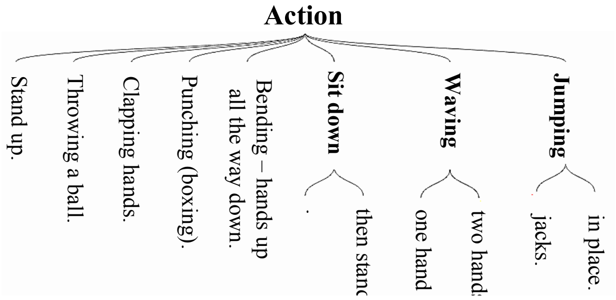


Fig. 1: Action hierarchy generated with the application of the proposed verb-centered lexical analysis on the class labels of the MHAD dataset, [*Jumping in place, Jumping jacks, Bending - hands up all the way down, Punching, Waving - two hands, Waving - one hand, Clapping hands, Throwing a ball, Sit down then stand up, Sit down, Stand up.*]

The first syntax rule corresponds to the general format of an action description, *do something with something*, which can be found in the majority of human action recognition datasets. Contrary, the second rule (initially presented in our previous work), corresponds to verb cases in which the juxtaposition of an ad-position/adverb after the verb changes entirely the semantic content of the verb, e.g. *take in* and *take out*. Finally, the role of the third rule is to deal with cases of words in the English language that can have more than one possible tag, due to multiple possible interpretations of the specific word. For example, the word *screw* can be interpreted as, (a) a **verb** referring to the act of putting something into its position by turning it, and, (b) a **noun** referring to a thin pointed piece of metal that is used for fastening one thing to another.

In order to impose these syntax rules in our text processing pipeline, we utilize Noun Phrase chuncking (NP-chunck) techniques [22] to segment and label multi-token sequences and identify the ones that follow the desired structure (grammar). The refined, tokenized label set is then clustered based on the similarity of verb tokens, with the cluster number being equal to the number of unique verbs identified in the entire label set. The clustered sets allow the construction of two-level action trees in which the first level nodes correspond to the verb-centroid of each cluster and the leafs to the fine-grained labels that have been assigned to this cluster. Figure 1, depicts the extracted action hierarchy for the MHAD dataset [23].

B. Incorporating the action hierarchy in DNN design

The previous stage allows the automatic generation of coarser action labels from the existing initial (finer) label set. The derived class set can serve as a more abstract, easier to solve formulation of the problem. One way to benefit from this formulation is to instruct the deep architecture to learn representations for this coarse-grained action set, and then, once it is able to discriminate the coarser action classes, use this representation as complementary information for the fine-grained classification problem.

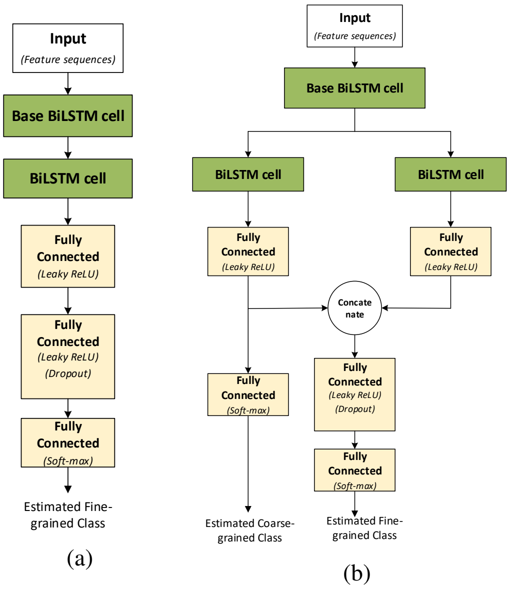
A deep architecture design approach that utilizes this dual problem formulation consists of a two-branch architecture, with the first sub-net being responsible for the coarser action classification, whereas the second sub-net is responsible for the finer classification. In order for the fine-grained sub-net to have access to the feature representations of the coarse-grained sub-net, an inter-connection can be utilized to skip-forward and connect (via concatenation) them with the existing fine-grained representation. One further design issue is the choice of the level at which we associate the two representations. In the present work, we consider two approaches regarding the feature-level of the coarse-grained sub-net. We should note that the fused maps are forwarded to the next layer set of the fine-grained sub-net.

- **Mid-level feature fusion:** fuse the feature maps of the sub-nets at the same high level of feature representation.
- **High to mid feature fusion:** fuse the learned coarse-grained action class probability distributions with the mid-level representations at lower layers of the fine-grained sub-net, and utilize the combined representation in the higher layers of the fine-grained modeling sub-net.

The first scheme is a common fusion strategy followed in the literature, combining the two representations at feature-level and attempting to map them into a more discriminative feature space. The second stems from the observation that, as the action complexity increases, so is the number of action motifs in the action sequence, expressed via verbs. For example, in the MPII Cooking Activities dataset [8], there exist cases of classes with more than one action verbs, such as *take and put in the fridge*. The action tree generation scheme proposed in this work only considers the first verb encountered, thus ignoring any additional verbs. This will lead to coarse-grained probability distributions that contain considerable probability values for the additional ignored verb cases. However, we can consider this probability distribution as an additional feature pattern, e.g. *take cup and put on the table* will lead to different pattern compared to *take cup and pour coffee*, despite the clustering of the two classes into the same coarse-grained class with the verb *take*.

With respect to the second approach, the probability vector corresponding to the coarse classes has much smaller dimensionality compared to the 1D feature vector from the FC layer of the fine-grained sub-net. In addition, this vector expresses assignment probability, thus, the value of each component ranges from 0 to 1, with the summation of all component values being equal to 1. To increase the contribution of the coarse prediction as a feature component in the fine-grained representation, we proceed with the following adjustments: (a) *batch normalization* in the feature vectors of the fine-grained FC layer to be concatenated with the coarse-grained probability vector, and, (b) *constrain the dimensionality* of the feature vector from fine-grained FC layer to be roughly D times (experimentally derived) the size of the probability vector from the coarse-grained part.

Overall, to enforce the symbiotic relation between the two action granularity levels in the learning process, a deep architecture that follows the aforementioned design strategy should minimize the following joint classification categorical cross-entropy-based loss function:


Input
(Feature sequences)
Base BiLSTM cell
BiLSTM cell
Fully Connected (Leaky ReLU)
Fully Connected (Leaky ReLU) (Dropout)
Fully Connected (Soft-max)
Estimated Fine-grained Class
Input
(Feature sequences)
Base BiLSTM cell
BiLSTM cell
BiLSTM cell
Fully Connected (Leaky ReLU)
Fully Connected (Leaky ReLU)
Concatenate
Fully Connected (Soft-max)
Fully Connected (Leaky ReLU) (Dropout)
Fully Connected (Soft-max)
Estimated Coarse-grained Class
Estimated Fine-grained Class

(a)

(b)

Fig. 2: (a) A simplified illustration of the custom baseline BiLSTM-based architecture, (b) its hierarchical variant.

$$L = \frac{1}{N}\sum_{n=1}^{N}\left[\sum_{k=1}^{K} T^{gn}_{n,k} log\left(Y^{gn}_{n,k}\right) + \sum_{l=1}^{L} w_l T^{fn}_{n,l} log\left(Y^{fn}_{n,l}\right)\right], \quad (1)$$

where w_l is a vector expressing label-centered associations between the fine-grained action classes, (T^{gn}_{n}, T^{fn}_{n}) are the ground-truth action labels for the coarse-grained and the fine-grained action classes respectively and, (Y^{gn}_{n}, Y^{fn}_{n}) are the estimated action classes.

Regarding the weight vector w_l, in the second part of the equation, it is derived from our previous work [12], concerning the extraction of misclassification penalization weights based on label description clustering using NLP. In more detail, we follow a verb-centered clustering of the action labels based on verb commonalities or verb with high semantic content similarity, with the verb cluster centers defining the coarser action class set. The weight vector values are uniformly defined based on the number of classes belonging in the same coarser action class.

IV. Experimental Setup

In the set of experiments that follow we assess the contribution of the proposed methodology to the performance of an action recognition pipeline. In order to illustrate the independence of the proposed approach to the specifics of the employed DNN, the evaluation is carried out on the basis of two deep architecture designs types. The designs involve (a) an architecture based on BiLSTM cells performing only temporal modeling of extracted frame-wise feature vectors serving as a baseline deep architecture design [12], and (b) the state-of-the art, end-to-end video classification architecture by Carreira

	Datasets		
	MHAD	***J-HMDB***	***MPII Cooking***
Num unique verbs	9 verbs	19 verbs	42 verbs
Avg num verbs/lbl	1.128 verb/lbl	1.0 verb/lbl	1.188 verbs/lbl
Avg lbl length	3.182 PoS/lbl	1.333 PoS/lbl	2.297 PoS/lbl
Avg asc via verb	0.545 asc/lbl	0.286 asc/lbl	1.656 asc/lbl
Max/min asc verb	1/0 asc	2/0 asc	5/0 asc
Num finer labels	11	21	64
Num Gen labels	8	18	36

TABLE I: **Dataset Label Statistics**. Abbreviations in the table contents refer to, *Avg*: average, *num*: number, *PoS*: part-of-speech, *lbl*: label, *Average asc*: average number of classes a single class is related to based on a PoS, *asc*: associations, is the amount of label lexical relations based on a specific PoS.

and Zisserman [24], based on 3D convolutions (Conv3D) and inception layers, with small variations to comply with the dual problem formulation.

Regarding the field of evaluation, we evaluate the proposed designs on three activity recognition datasets, namely Berkeley's MHAD [23], the J-HMDB [25] and Max Planck's Cooking dataset [26]. These datasets correspond to coarse-grained, mid-range, and fine-grained activity sets with the action complexity reflected on the size and complexity of the label descriptions. Moreover, the proposed two-level action tree hierarchy results in at least 20% reduction of the class amount in the coarser-level classification task, with increasing rates as the activity complexity and diversity increases. Table I presents statistical information regarding the number of unique verbs, average number of verbs per label, average verb associations between classes based on verbs, number of fine-grained (initial) classes and the generated number of coarser action labels.

A. Architecture modifications

For evaluating the proposed action hierarchy-based deep design directions, we modify the baseline architectures in order to introduce the hierarchical action format of the derived action tree into the deep design.

Custom baseline BiLSTM-based DNN: Compared to the initial single stream design (presented in [12] and illustrated in Fig. 2a), the modifications involve maintaining only the first BiLSTM layer and decoupling the two action granularity levels into discrete sub-nets. The coarse-level sub-net consists of a BiLSTM layer followed by a two-level Fully-Connected (FC) layer set, with Leaky ReLU and soft-max activation functions, respectively. The goal of this sub-net is to produce probability distribution estimates for the set of coarse-grained classes. Contrary, the fine-grained sub-net also consists of a BiLSTM layer followed by a three FC layer set, with the first two utilizing a Leaky ReLU and dropout sequence, whereas the last exploits a soft-max activation function, which generates the fine-grained class estimates. Moreover, the input of the second FC layer is defined to be the concatenation of the feature maps of the first FC layer of the coarse-grained sub-net and those of the corresponding level of the fine-grained network. Both sub-nets share the same initial BiLSTM layer, as depicted in Fig. 2b.

Architecture Design	**Datasets** (mAcc. (Coarse, Fine)%)		
	MHAD	***J-HMDB***	***MPII Cook***
NH-BiLSTM	(-, 64.17)%	(-, 36.28)%	(-, 29.45)%
H-BiLSTM	(82.50, 70.25)%	(45.68, 42.61)%	(60.70, 35.40)%
NH-I3D [24]	(-, 89.61)%	(-, 72.38)%	(-, 48.18)%
H-I3D	(98.75, 96.38)%	(78.47, 76.10)%	(70.47, 54.30)%

TABLE II: Action recognition performance for the MHAD, JHMDB and MPII datasets between hierarchical (H) and non-hierarchical (NH) deep architecture designs.

Custom I3D network: We maintain the original design up until the last receptive field up-sampling layer-block, using the pre-trained weights on ImageNet [27] and Kinetics [24]. For our hierarchical design modifications, the coarse-grained and fine-grained sub-networks maintain the same design as the previous architecture with the difference of the replacement of BiLSTM with Conv3D layers.

B. Training configurations

The batch size for MPII and J-HMDB was set to 32 samples per batch, whereas for MHAD to 16. The networks were trained for 20K iterations for MHAD and J-HMDB and for 38K iterations for the MPII dataset, in a Nvidia Quadro P6000 GPU. The loss minimization is performed using the Adadelta optimizer. For MPII Cooking, due to the large range of action segment sizes, we train the networks with video clips of 10 frames, sampled uniformly across the entire sequence. For MHAD, we only utilized data from only a single viewing angle. No data augmentation was introduced.

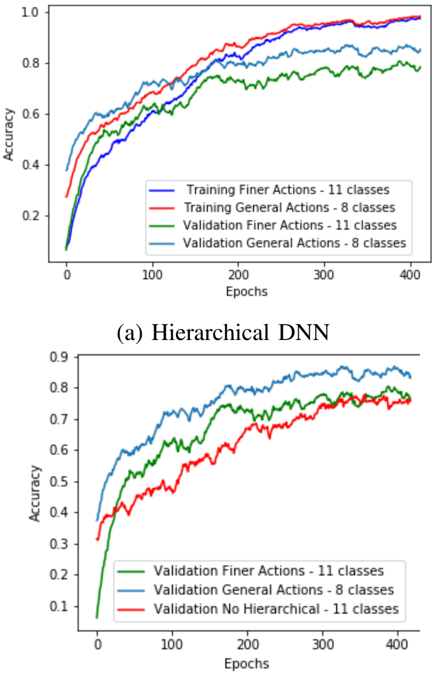
Regarding the learning process, we follow the standard experimental protocol described in the corresponding baseline dataset papers and report, for the case of multiple splits, the average accuracy across all splits.

V. Experimental Results

Our experiments are organized as follows. Initially, we compare the hierarchical variants of the custom and the I3D-centered action recognition architectures with their non-hierarchical counterparts in terms of recognition accuracy and learning performance. Subsequently, we assess qualitatively the contribution of the new formulation using visualizations of the learned feature mapping at different training stages of the two sub-networks in the proposed hierarchical architecture design. Finally, we explore different fusion schemes for the two action granularity modeling sub-nets, and assess the corresponding DNN performance.

A. Accuracy, label complexity, deep architecture evaluation

The first set of experiments aims at assessing the impact of adopting a hierarchical design on a deep architecture for action recognition. For this purpose we compare existing one-level granularity action classification networks with variants that



(a) Hierarchical DNN

(b) Hierarchical and Non-Hierarchical DNN

Fig. 3: (a) Training and validation accuracy change for the hierarchical action classification deep architecture for the MHAD dataset, (b) Training and validation accuracy change between the hierarchical and non-hierarchical architectures.

exploit the proposed two-level granularity hierarchy. Table II presents the test accuracy scores. The input source for both architectures are raw RGB frame sequences, which for the BiLSTM-based architecture are first passed from the InceptionNetv3 [28] network pre-trained on ImageNet, in order to be transformed into sequences of frame-wise feature vectors.

We observe that re-formulating our classification task into a two-stage optimization problem, which initially involves classifying actions into a set of coarser action categories and then refining our estimations to a set of fine-grained action classes, leads to an increase in accuracy in every dataset and architecture case. Specifically, an improvement in recognition accuracy in the range of 4 to 6% can be observed for both DNN architectures, for every dataset case.

Regarding the relation between the learning process of the two action granularity levels, as can be observed in Fig. 3a for the case of MHAD, the learning rate of the two granularity levels is simultaneous with similar gradation. The small margin between the performance of the two sub-networks during learning suggests that each refinement, at each epoch for the coarse-grained action representation, has a positive effect on the fine-grained representation learning as well, indicating the contribution of the coarse-grained representation. If this was not the case, we would observe higher performance margins and slower learning rates for the fine-grained learning problem. This would be indicative that the coarse-grained representation has no or little contribution, and the model struggles to learn meaningful representations. Overall, we can observe that the model generalizes well and does not exhibit under- or over-fitting behaviours.

Finally, as shown in Fig. 3b, the hierarchical architecture learns faster compared to the non-hierarchical one, regarding the first set of training epochs. However, we can observe that as the number of training epochs increases, the difference in learning speed between the two directions decreases, and eventually balances as we approach the limits of the network's learning capacity.

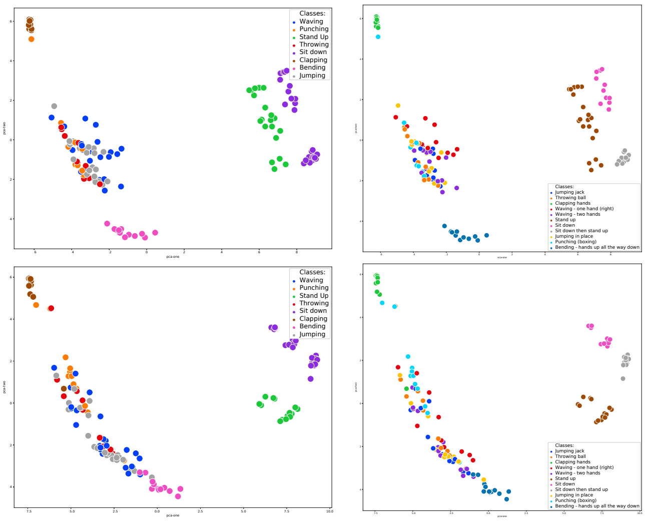


Fig. 4: Scatter-plots of the first two PCA components for the last BiLSTM layer of the coarse-grained action classification sub-network of the baseline hierarchical DNN architecture. First row plots refer to a training accuracy of 75% for the coarser action recognition sub-net and a 61% for the finer action recognition sub-net. Second row plots refer to a training accuracy of 84.2% for the coarser and a 71.4% for the finer action recognition sub-net.

B. Visualization of learned representations

In order to have a better understanding of the underlying representations that each action granularity sub-net learns and their relation, we visualize the high-dimensional feature space by mapping it on a 2D plane. Specifically, we employ learned feature maps of (a) the last FC layer and, (b) the dedicated BiLSTM layer of each sub-net in the custom baseline DNN architecture. A standard way to generate such visualizations is to employ dimensionality reduction techniques, such as Principal Component Analysis (PCA), or more recently the t-distributed Stochastic Neighbor Embedding (t-SNE) [29]. In this work we opted to use PCA on the learned feature maps for the MHAD dataset. Our selection stems from the small amount of samples and the fact that the first two components account for about 85% to 90% of the variation in the entire dataset. More over, PCA generated better and more easily distinctive visualizations in the 2-D space compared to t-SNE, providing a good intuition regarding the clustering of the samples and the separability of the classes based on the learned representation. The first visualization aims at validating the correct lexical clustering of the action classes. Specifically, we expect samples belonging to classes that were clustered together under a specific coarser verb-centered class to be mapped in near-by locations on the plane. To that end, Fig. 4 shows the positioning and underlying grouping of the dataset samples in the training set, based on the learned representation of the coarse-grained action sub-network, for the case of the BiLSTM layer that is dedicated to this granularity. Left-wise scatter-plots are coloured based on the corresponding coarse-grained action labels whereas right-wise scatter-plots are coloured based on the corresponding fine-grained action classes. Moreover, row-wise each sub-figure pair depicts the difference between the generated representation across different stages of training. The clustering formations verify the correct verb-centered lexical grouping of the action labels and indicates that visualization of learned representations at the initial training stages can be used to assess and refine the lexical grouping of the action labels.

To further confirm the contribution of integrating hierarchical action granularity analysis through lexical descriptions of action labels, we include a second illustration depicting the final representations learned from the DNN that leverages this additional knowledge in its design in relation to the one that doesn't. Figure 5 depicts the first two PCA components for both the hierarchical and non-hierarchical architectures, using the learned feature representations of the final FC layer of each architecture. It can be observed that the action hierarchy-inspired DNN architecture leads to more discriminative features, validating the overall accuracy gain.

C. Fusion strategies

We evaluate two different ways to combine the representations of the two granularities. The first combines the learned representations at a mid-level phase, concatenating the feature maps of the same representational level, illustrated in Fig. 2b. The joint feature maps are generated by concatenating the learned feature maps of the first FC layers. We should note that in our design the two layers have the same dimensionality. The concatenated representation is then forwarded deeper in the higher layers of the fine-grained sub-network.

Contrary, the second direction utilizes the learned coarse-grained probability distributions as a feature descriptor. For this, we combine the coarse-grained probability distributions (CPD) with mid level representations learned at the lower layers of fine-grained sub-net. Specifically, we fuse the coarse-grained probability distributions obtained from Softmax, and the feature maps obtained from the first FC layer of the fine-grained branch. The utilization of the learned coarse-grained probability distributions as a feature set is based on the assumption that such representation will be more beneficial for complex action sets, for which a single verb-centered grouping is not adequate, since such activities combine multiple action motifs and are usually expressed with more than one verbs in their label sentences.

In the second fusion scheme, one needs to be careful with the determination of the dimensionality of the fused feature vectors. The dimensionality of the CPD feature vector is smaller (equal to the coarse-grained class set) compared to that of the fine-grained mid level FC-generated representation. This could potentially limit the significance of the coarse-grained probability distribution set. To alleviate this, we experimentally found that concatenating the CPD with a feature descriptor that is 6 times larger, finds a good balance in performance, with smaller sizes leading to an under-fitting model.

Architecture Design	**Datasets** (mAcc. (Coarse, Fine)%)		
	MHAD	***J-HMDB***	***MPII Cook***
H-BiLSTM	(82.50, 70.25)%	(45.68, 42.61)%	(60.70, 35.40)%
HFP-BiLSTM	(86.35, 65.46)%	(42.41, 39.55)%	(36.84, 28.19)%
H-I3D	(98.75, 96.38)%	(78.47, 76.10)%	(70.47, 54.30)%
HFP-I3D	(91.35, 82.89)%	(67.17, 60.46)%	(60.34, 37.55)%

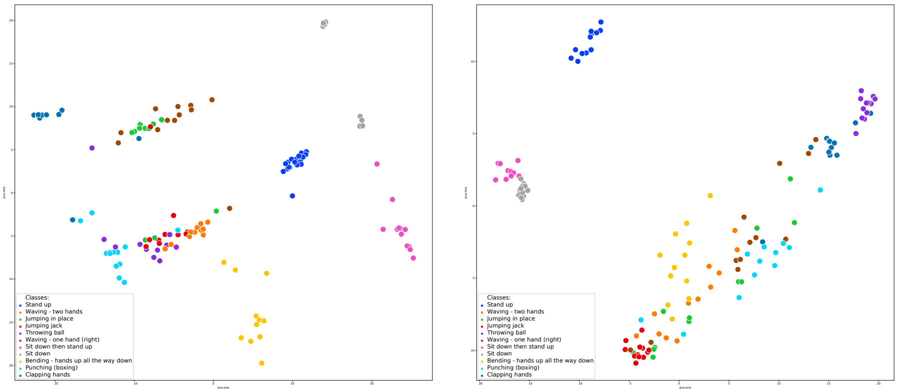
TABLE III: Accuracy variations due to (a) mid-level feature fusion (H), (b) coarse-grained probability distribution level to mid-level fine-grained feature fusion (HFP).

As can be observed in Table III, the mid-level fusion scheme allows for better exploitation of the learned coarse-grained action representation. The CPD-centered direction is not able to reach the performance of the previous fusion scheme, due to the limited information contained in this 1D coarse-grained class assignment vector. This fact, combined with the requirement to limit the dimensionality of the integrated vector, constrains the modeling capacity and leads to performance degradation as the complexity of the action increases.

VI. Conclusions

This paper dealt with a new approach for enhancing the performance of deep learning-based human activity recognition models by leveraging existing linguistic information on action labels. Specifically, we proposed a method involving a verb-centered vocabulary analysis of the label sentences, with the goal of organizing action classes into groups sharing the same action-related verb or containing verbs with very similar semantic content. This method allows for the extraction of a new action class set based on a generalized action motif, expressed solely with the action verb, and thus forming a two-level action hierarchy. Moreover, we introduced design directions that allow the exploitation of the developed two-level action tree hierarchy by a HAR deep learning architecture. The reformulation of the problem into a two-level coarse-to-fine optimization process enriches the model's discriminative power. This observation is backed with extensive quantitative evaluation on three datasets at various action granularity levels.

However, it is evident that the success of the proposed strategy is intertwined with the correct identification of the lexical correlations between the action labels. This requires more elaborate vocabulary analysis which will generate multi-level action granularity hierarchies based on the underlying motion motifs. Given that even a bi-level analysis is capable of enhancing the learning ability of HAR deep model, additional work on the lexical analysis of action labels appears to be a promising research direction.



(a) Hierarchical DNN architecture

(b) Non-hierarchical DNN architecture

Fig. 5: Scatter-plots from the application of PCA to the feature representations generated by the last fully-connected layer of the custom BiLSTM-based (a) hierarchical and (b) non-hierarchical architectures applied on the MHAD dataset.

Acknowledgments

This work was partially supported by the European Community through the project Co4Robots (H2020-731869). We also gratefully acknowledge the support of NVIDIA Corporation for the donation of a GPU.

References

[1] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and Vision Computing*, vol. 60, pp. 4–21, 2017.

[2] C. Liu, J. Liu, Z. He, Y. Zhai, Q. Hu, and Y. Huang, "Convolutional neural random fields for action recognition," *Pattern Recognition*, vol. 59, pp. 213–224, 2016.

[3] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*. IEEE, 2015, pp. 2625–2634.

[4] J. Wang, A. Cherian, and F. Porikli, "Ordered pooling of optical flow sequences for action recognition," in *WACV*. IEEE, 2017, pp. 168–176.

[5] G. Chéron, I. Laptev, and C. Schmid, "P-cnn: Pose-based cnn features for action recognition," in *ICCV*. IEEE, 2015, pp. 3218–3226.

[6] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019.

[7] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," 2008.

[8] M. Rohrbach, A. Rohrbach, M. Regneri, S. Amin, M. Andriluka, M. Pinkal, and B. Schiele, "Recognizing fine-grained and composite activities using hand-centric features and script data," *IJCV*, vol. 119, no. 3, pp. 346–373, 2016.

[9] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," in *ECCV*. Springer, 2016, pp. 852–869.

[10] M. Wray and D. Damen, "Learning visual actions using multiple verb-only labels," *arXiv preprint arXiv:1907.11117*, 2019.

[11] S. Khamis and L. S. Davis, "Walking and talking: A bilinear approach to multi-label action recognition," in *CVPR Workshops*. IEEE, 2015, pp. 1–8.

[12] K. Bacharidis and A. Argyros, "Improving deep learning approaches for human activity recognition based on natural language processing of action labels," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[13] V. W. Zheng, D. H. Hu, and Q. Yang, "Cross-domain activity recognition," in *Proceedings of the 11th international conference on Ubiquitous computing*, 2009, pp. 61–70.

[14] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 102–114, 2016.

[15] Y. Zhu and S. Newsam, "Efficient action detection in untrimmed videos via multi-task learning," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 197–206.

[16] W. Liu, C. Zhang, J. Zhang, and Z. Wu, "Global for coarse and part for fine: A hierarchical action recognition framework," in *ICIP*. IEEE, 2018, pp. 2630–2634.

[17] C. Avilés-Cruz, A. Ferreyra-Ramírez, A. Zúñiga-López, and J. Villegas-Cortéz, "Coarse-fine convolutional deep-learning strategy for human activity recognition," *Sensors*, vol. 19, no. 7, p. 1556, 2019.

[18] N. Hussein, E. Gavves, and A. W. Smeulders, "Timeception for complex action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 254–263.

[19] Z. Tu, W. Xie, Q. Qin, R. Poppe, R. C. Veltkamp, B. Li, and J. Yuan, "Multi-stream cnn: Learning representations based on human-related regions for action recognition," *Pattern Recognition*, vol. 79, pp. 32–43, 2018.

[20] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, p. 39–41, Nov. 1995. [Online]. Available: https://doi.org/10.1145/219717.219748

[21] E. Loper and S. Bird, "Nltk: The natural language toolkit," in *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ser. ETMTNLP '02. USA: Association for Computational Linguistics, 2002, p. 63–70. [Online]. Available: https://doi.org/10.3115/1118108.1118117

[22] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O'Reilly Media, 2009.

[23] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Berkeley mhad: A comprehensive multimodal human action database," in *IEEE WACV*. IEEE, 2013, pp. 53–60.

[24] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*. IEEE, 2017, pp. 6299–6308.

[25] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *ICCV*. IEEE, 2013, pp. 3192–3199.

[26] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *CVPR*. IEEE, 2012, pp. 1194–1201.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*. IEEE, 2009.

[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*. IEEE, 2016, pp. 2818–2826.

[29] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," 2008.