Expert Systems With Applications 139 (2020) 112829

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences

Abdessamad Elboushaki*, Rachida Hannane, Karim Afdel, Lahcen Koutti

Laboratory of Computer Systems and Vision, Faculty of Science, Ibn Zohr University, Agadir, 80000, Morocco

ARTICLE INFO

Article history:
Received 19 October 2018
Revised 10 June 2019
Accepted 19 July 2019
Available online 20 July 2019

Keywords:
Gesture recognition
Deep learning
Convolutional neural networks
Multimodal learning
Feature fusion
RGB-D video processing

ABSTRACT

Human gesture recognition has become a pillar of today's intelligent Human-Computer Interfaces as it typically provides more comfortable and ubiquitous interaction. Such expert system has a promising prospect in various applications, including smart houses, gaming, healthcare, and robotics. However, recognizing human gestures in videos is one of the most challenging topics in computer vision, because of some irrelevant environmental factors like complex background, occlusion, lighting conditions, and so on. With the recent development of deep learning, many researchers have addressed this problem by building single deep networks to learn spatiotemporal features from video data. However, the performance is still unsatisfactory due to the limitation that the single deep networks are incapable of handling these challenges simultaneously. Hence, the extracted features cannot efficiently capture both relevant shape information and detailed spatiotemporal variation of the gestures. One solution to overcome the aforementioned drawbacks is to fuse multiple features from different models learned on multiple vision cues. Aiming at this objective, we present in this paper an effective multi-dimensional feature learning approach, termed as MultiD-CNN, for human gesture recognition in RGB-D videos. The key to our design is to learn high-level gesture representations by taking advantages from Convolutional Residual Networks (ResNets) for training extremely deep models and Convolutional Long Short-Term Memory Networks (ConvLSTM) for dealing with time-series connections. More specifically, we first construct an architecture to simultaneously learn the spatiotemporal features from RGB and depth sequences through 3D ResNets which are then linked to a ConvLSTM to capture the temporal dependencies between them, and we show that they better combine appearance and motion information effectively. Second, to alleviate distractions from background and other variations, we propose a method that encodes the temporal information into a motion representation, while a two-stream architecture based on 2D-ResNets is then employed to extract deep features from this representation. Third, we investigate different fusion strategies at different levels for blending the classification results, and we show that integrating multiple ways of encoding the spatial and temporal information leads to a robust and stable spatiotemporal feature learning with better generalization capability. Finally, we perform different experiments to evaluate the performance of the investigated architectures on four kinds of challenging datasets, demonstrating that our approach is particularly impressive where it outperforms prior arts in both accuracy and efficiency. The obtained results affirm also the importance of embedding the proposed approach in other intelligent systems application areas.

1. Introduction

With the massive infusion of technologies (e.g. computers and machines) in todays society and their ubiquitous usage in our daily-life activities, facilitating human-computer interaction (HCI) has become one of the most interesting research topics in recent years. Hence, there has been a growing focus among researchers to develop and build new intelligent interface systems to achieve this goal. The ultimate objective in this process is to bridge HCI gap and make it as natural as human-human interactions. Among potential cornerstones of HCI systems, human gesture recognition based on computer vision is an active area of research, whose goal is to analyze and interpret the ongoing visible human behaviors

* Corresponding author.
E-mail addresses: abdessamad.elboushaki@edu.uiz.ac.ma (A. Elboushaki), rachida.hannane@edu.uiz.ac.ma (R. Hannane), k.afdel@uiz.ac.ma (K. Afdel), l.koutti@uiz.ac.ma (L. Koutti).

https://doi.org/10.1016/j.eswa.2019.112829
0957-4174/

that have meaning within a particular context. These behaviors may involve the movements of the whole body or some parts of it (i.e. upper body, lower body, hands, arms or head), depending on the application scenario. Vision-based human gesture recognition is thought to be one of the most natural, intuitive, friendly, and less intrusive form of interaction, particularly in noisy environments or where physical contact and/or speech are not possible. Because of the practical importance associated with this topic, it has drawn much attention from both industry and academic communities (Escalera, Athitsos, & Guyon, 2017).

The development of gesture-based HCI interfaces is beneficial to many expert and intelligent applications, ranging from surveillance events understanding (Dikmen et al., 2008; Kim et al., 2016) to entertainment (Seger, Wanderley, & Koerich, 2014) and many other aspects of daily life (Maqueda, del Blanco, Jaureguizar, & Garcia, 2015). In the first instance, human gestures can be used as a communicative tool for sign language translation systems (Almeida, Guimares, & Ramrez, 2014; Huang, Zhou, Li, & Li, 2015). This would help deaf and hard of hearing impaired people, speech-impaired people, and people with a learning disability to interact with computers or other electronic devices. Gaming is another interesting field of intelligent applications (Rautaray & Agrawal, 2011), where human gestures can be used as an input mode for interacting with computer games. For example, controlling the movement and orientation of interactive game objects, or navigating around a game environment. Human gestures are also famous in the domain of robot control (Tang, Yusuf, Botzheim, Kubota, & Chan, 2015) in which the robotic gestures can be used to control and interact with the robots in a normal way. The robotic gestures are most commonly seen in virtual reality applications (LaViola, Jr, & J., 2015) as the operator is controlling a robots actions while viewing the robots environment through a head-mounted display like Google glasses. In the domain of healthcare (Jacob & Wachs, 2014), the surgeons may use the gesture interfaces in the operating room to browse and display medical images of a patient during the operation, or to remotely control the surgical devices in aseptic environments. Eventually, human gestures are also being considered to build up intelligent interaction systems for vehicle driving assistance commands (Molchanov et al., 2016), as well as for smart houses (Diraco, Leone, & Siciliano, 2013).

Over decades, computer vision algorithms relied on only visual information to recognize gestures. The recent availability of cost-effective multimodal sensors (ex. Microsoft Kinect) have made simultaneous acquisition of depth and RGB data possible, thus widely increasing the motivation in developing methods for gesture recognition by taking advantages from both depth and RGB modalities (RGB-D). Depth cues provide rich 3D structural information of the scene that is more insensitive to illumination changes, clothing and, skin color, which make them act as a substantial complement to the original RGB data. Although the extensive research efforts that have been made in this context Escalera et al. (2017), accurate recognition of gestures in unconstrained videos remains an extremely challenging task, mostly when faced with viewpoint variation, cluttered scene backgrounds, occlusion, intra-class and inter-class variability as well as noises in the acquisition. Other difficulties involve inconsistent and non-standard behaviors among different performers, and complex variations in spatial and temporal scales.

Different from other visual classification tasks that rely on still images, dynamic gestures generally provide a rich communication channel because of the incorporation of both spatial and motion information from video data. Therefore, extracting effective spatiotemporal features to capture the spatial and temporal evolutions of different gestures is crucially important to accomplish the recognition task. This is challenging from a modeling perspective as we have to model complicated human movements in a three-dimensional structure. According to Tran, Bourdev, Fergus, Torresani, and Paluri (2015) and Zhang et al. (2017b), effective spatiotemporal features of gestures need to be (i) *robust*, (ii) *generic*, (iii) *compact*, (iv) *efficient* to compute, and (v) *simple* to implement. Up to date, it is still very difficult to build a practical intelligent gesture recognition system that strongly satisfies all these requirements.

Not surprisingly, much work has been done regarding gesture recognition. Traditional approaches (Choi & Park, 2014; Cirujeda & Binefa, 2014; Laptev, 2005; Liu & Liu, 2016; Tung & Ngoc, 2014; Zheng, Feng, Xu, Hu, & Ge, 2017) attempted to exploit specific handcrafted features to construct more representative spatial or spatiotemporal descriptors for video data. The constructed descriptors are then fed into a discriminative classifier (ex. linear SVM, PCA, Naive Bayes model, and so on) for gesture classification. Generally, handcrafted features are built on the pixel-level and can be either densely sampled or extracted from interest points (Wang, Ullah, Klaser, Laptev, & Schmid, 2009). This makes them more robust to the local geometric distortions of spatial body shape. However, in addition to their high computational complexity, most of the existing handcrafted features are dataset dependent, which makes them difficult to be generalized from one scenario to another scenario. Therefore, approaches based on these features alone may potentially fail to be efficient enough in some real-world environments.

Recently, there has been an increasing interest in a more advanced class of feature representation models, called Deep Neural Networks (DNNs) (Hinton, Osindero, & Teh, 2006). DNNs have the ability to learn complex hierarchies with increasing levels of abstraction while being end-to-end trainable. In view of the availability of large amounts of training data with appropriate regularization, deep neural models have quickly become a reference methodology for obtaining outstanding results in many computer vision and pattern recognition tasks (Girshick, Donahue, Darrell, & Malik, 2014; Jain, Tompson, LeCun, & Bregler, 2014; Ji, Xu, Yang, & Yu, 2013; Kang, Kim, & Kim, 2017; Krizhevsky, Sutskever, & Hinton, 2012). As a particular type of DNNs, Convolutional Neural Networks (CNNs) LeCun, Bottou, Bengio, and Haffner (1998) perform more intellectual learning by successively applying trainable filters with pooling operations on the input raw data, resulting in a hierarchical structure of high-level distinctive features. As in many other computer vision areas, researchers have also successfully exploited CNNs for gesture recognition, achieving outstanding results and outperforming the most prominent non-deep state-of-the-art methods Asadi-Aghbolaghi et al. (2017).

The first strategy that comes to mind for recognizing gestures with deep learning is applying 2D CNNs on individual frames to extract frame-based features and then conduct temporal fusion (Koller, Ney, & Bowden, 2016; Tompson, Stein, Lecun, & Perlin, 2014; Wu, Ishwar, & Konrad, 2016a). This type of approaches is easier to be fine-tuned on pre-trained models due to the availability of large annotated datasets (e.g. ImageNet). In spite of this advantage, they only use the appearance information to extract features, where there is no temporal encoding during the feature learning stage. In addition, the employed temporal fusion method tends to neglect the temporal order of the sequence. Another obvious strategy is to extend 2D CNNs to 3D CNNs, which allows learning motion features by 3D filters in their 3D convolutional and pooling layers (Huang et al., 2015; Ji et al., 2013; Liu, Zhang, & Tian, 2016a; Molchanov, Gupta, Kim, & Kautz, 2015). It has been shown that applying 3D CNNs to the entire video sequences is effective for modeling complex spatiotemporal patterns. However, because of the huge amount of parameters to learn, training such networks is a challenging task, especially when dealing with the long duration sequences. Another drawback of these networks is that they suffer from cluttered scene background and noise distortion as they

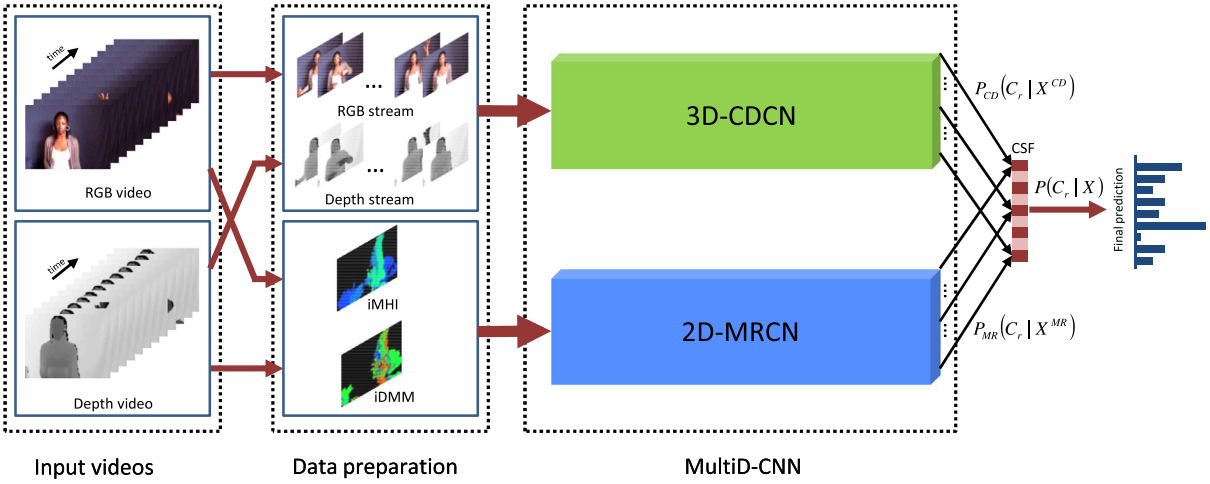


Fig. 1. Pipeline of the proposed MultiD-CNN framework. The inputs are depth and RGB data modalities. MultiD-CNN mainly consists of two sub-networks: 3D Color-Depth Convolutional Network (3D-CDCN) and 2D Motion Representation Convolutional Network (2D-MRCN). 3D-CDCN simultaneously learns spatiotemporal features from RGB and depth streams, while 2D-MRCN takes as inputs the improved Motion History Image (iMHI) and the improved Depth Motion Map (iDMM), which are generated from RGB and depth sequences, respectively. Predictions from both sub-networks are then fused in a class score fusion layer (CSF) to obtain the final gesture label.

directly operate on raw video data, making them confused between learning spatiotemporal relevant and irrelevant patterns. The third strategy uses what so-called motion-based input model for gesture recognition (Ijjina & Chalavadi, 2017; Wang et al., 2016b; Zhang, Wei, Song, & Zhang, 2017a). In this strategy, motion representations such as optical flow are computed from data before their usage and then fed to the deep models. Training models on pre-computed motion features is an effective way to save them from implicit learning of motion features. Moreover, motion representations are compact and robust to the gesture-environmental factors, making these models concentrated on learning only important features as these factors are filtered earlier. One of the major limitations of this strategy is that temporal information may inevitably be lost during the encoding of video into its motion representation. The fourth crucial strategy in the context of deep learning methods employs temporal networks like Recurrent Neural networks (RNNs) and Long Short-Term Memory (LSTM) to model dynamics of videos (Chai, Liu, Yin, Liu, & Chen, 2016; Liu, Shahroudy, Xu, & Wang, 2016b; Nishida & Nakayama, 2015). LSTM learns how to integrate information over time by using memory cells to store, modify, and access internal state. This allows it to better discover long short-range temporal relationships of videos. However, as memory cells utilize full connections in input-to-state and state-to-state transitions, the spatial correlation information is not encoded.

From the above analysis, it can be observed that each of these strategies has its strengths and weaknesses. This emphasizes the fact that models based on a single strategy lack sufficient robustness and have limited performance in gesture recognition since the extracted features cannot efficiently capture both relevant visual information and detailed spatiotemporal variation of gestures simultaneously. One possible solution to overcome these problems is to design multiple feature channels in a hybrid model learned on multiple vision cues. Such a combination can provide a compact, yet discriminative feature representation, which will obviously lead to an efficient gesture recognition system. Nevertheless, it is not an easy task given the heterogeneous nature of each strategy, which apparently requires an intelligent fusion mechanism between them.

Targeting at this objective, we propose in this paper a multi-dimensional deep learning-based framework, called MultiD-CNN, for the task of human gesture recognition in videos (shown in Fig. 1). Specifically, our main goal is to investigate several learning methodologies to automatically extract high-level features from RGB-D modalities and exploit them for gesture recognition. First, theoretically motivated by the fact that simultaneous encoding of both spatial and temporal information can better represent motion patterns in a three-dimensional structure than handling them separately, we propose a novel deep model called 3D Color-Depth Conventional Network (3D-CDCN) based on the idea of cooperative learning. It takes firstly the RGB and depth sequences as inputs and fed them into two separate 3D Convolutional Residual Networks (3D ResNets) to learn spatiotemporal gesture representation. Our inspiration to use ResNets (He, Zhang, Ren, & Sun, 2016) is due to two main advantages: (i) The convolutional layers in ResNets significantly reduce the number of trainable parameters using the concept of weights sharing, which help the model to overcome the usual problem of over-fitting. (ii) ResNets introduce shortcut connections that perform identity mapping, thus effectively avoid signal attenuation caused by multiple stacked non-linear transformations. Up to now, we ensure that the extracted features maintain the short time lags, whilst the global dynamics of video are not yet modeled. To compensate for this, these features are then given to a Convolutional Long Short-Term Memory (ConvLSTM) network (Xingjian et al., 2015) connected to the output of the 3D ResNets to further learn the spatiotemporal dependencies between them. Differently from traditional LSTM, ConvLSTM explicitly assumes that the input is a sequence of images and replaces the vector multiplication in LSTM gates by convolutional operations, in which the intermediate representations of the images preserve the spatial correlation information during the recurrence. The results from both streams are then fused for final prediction. Second, because the expressed gestures might be affected by different irrelevant factors (such as cluttered backgrounds, noise, and illumination variance), direct learning on raw data can distract the model from capturing useful information. So, it is highly desirable to eliminate these variations from data and make the learning only focus on the motion of performers. To achieve this, the theory of motion representation is employed. Specifically, we propose another deep model proclaimed 2D Motion Representation Conventional Network (2D-MRCN) that learns to recognize gestures from their motion representation. The design of motion representation in our approach aims to accumulate motion information into static images as better representatives of gesture spatiotemporal states. That is to say, two spatial motion maps (i.e. improved Motion History Image (iMHI) and improved Depth Motion Map (iDMM)) are computed from RGB

and depth sequences, respectively. Each of them provides an explicit encoding of gesture movements in form of specific appearances and shapes. Two 2D ResNets are then trained on these two motion images and the results are combined to obtain the final classification score. We demonstrate that our motion representation through its effective de-background process can reduce the influence of gesture-irrelevant factors, while enabling 2D-MRCN to learn effective spatiotemporal features in a two-dimensional structure. Third, it is theoretically demonstrated that ensemble learning is more conducive to further improve the recognition performance. This is also the main strength of this work. In this regard, we propose a set of comprehensive fusions between different components of MultiD-CNN to boost the recognition accuracy. We adopt feature level fusion between different modalities so that to investigate their complementary advantages. Meanwhile, linear fusion at decision level between 3D-CDCN and 2D-MRCN is exploited to obtain the final classification label. Lastly, by leveraging these three principles, we achieve the state-of-the-art performances on four different gesture classification tasks: Chalearn LAP IsoGD (Wan et al., 2016b), Shefeld KInect Gesture (SKIG) (Liu & Shao, 2013), NATOPS gesture (Song, Demirdjian, & Davis, 2011), and SBU Kinect interaction (Yun, Honorio, Chattopadhyay, Berg, & Samaras, 2012).

Based on our literature review, the proposed MultiD-CNN presents several advantages compared to other existing expert and intelligent systems in the field. First, we do not rely on handcrafted features that are time-consuming to design, especially when this has to be done independently for each data modality (e.g. Althloothi, Mahoor, Zhang, & Voyles, 2014; Zheng et al., 2017. Instead, we utilize convolutional networks to automatically extract the relevant spatiotemporal information from the data. Second, Regions-of-Interest (RoI) to detect human parts, e.g. upper body or hands, is not required for MultiD-CNN. Approaches based on RoI such as (Narayana, Beveridge, & Draper, 2018; Wang, Wang, Song, & Li, 2017b) need an extra pre-processing stage to crop the hands from the rest of the scene at each video frame (usually using Faster R-CNN), making the overall process computationally expansive. In addition, the performance of these approaches largely depends on the extent to which the hands are accurately detected, which is a challenging task in its own right. More than that, the networks focusing on the hand alone do not take the essence of generalization capability into consideration. This makes them work exclusively for gestures involving the movements of the hands. Differently from most others, our approach, including both 3D-CDCN and 2D-MRCN, takes into account all the moving parts of the body that represent the gesture by investigating the complete spatial and temporal information from RGB-D sensor. Third, instead of using score fusion between the networks that are architecturally similar such in Wang et al. (2017b), Zhu, Zhang, Shen, and Song (2017) and Molchanov et al. (2015), we exploit the high-level representation learned by each network and we adopt the intermediate fusion which processed at feature level. This is more effective due to the rich complementary information of each kind of the utilized features. In addition, the employed multi-dimensional fusion strategy helps to improve the performance of the overall system and makes it suitable to overcome the usual shortcomings of many other techniques proposed so far for gesture recognition. All of these extensions have shown clear advantages in our experimental comparison to previous methods on expert and intelligent systems.

In a nutshell, it is worth highlighting the key contributions below:

- We introduce a novel deep learning-based framework, termed as MultiD-CNN, to learn spatiotemporal features from RGB-D videos and exploit them for gesture recognition. In this framework, we incorporate the spatial and temporal information through two different recognition models: 3D-CDCN and 2D-MRCN. 3D-CDCN appends the temporal dimension as an extra learning channel and employs 3D ResNets and ConvLSTM to simultaneously learn spatiotemporal features. Meanwhile, 2D-MRCN accumulates the motion across the video sequences into a motion representation and uses 2D ResNets to learn on it. Such diversity leads to robust and stable gesture recognition with better generalization capability.
- We investigate different fusion strategies at different levels (i.e. feature level and decision level) to blend the outputs of various MultiD-CNN components. The results of these mechanisms show that multiple-channel fusions outperform individual modules.
- We apply the proposed MultiD-CNN to different kinds of human behaviors recognition and we found promising results for upper body gesture, hand gesture, and two-person interaction datasets. This demonstrates the effectiveness of the proposed approach and suggests more general applicability to other video classification tasks.
- Last but not least, our proposed MultiD-CNN for gesture recognition may serve as an expert system, or as a part or a component that builds other intelligent systems for different HCI application domains such as healthcare and smart houses.

The remainder of this paper is organized as follows: In Section 2, we review some related work for gesture recognition organized according to the type of feature representation used. In Section 3, we describe the overall structure of the proposed MultiD-CNN framework for gesture recognition. In Section 4, we report our experimental evaluation on several challenging datasets. The last Section 5 concludes the paper.

2. Related work

Because of the importance of gesture recognition in computer vision and pattern recognition fields, much research works have been done in this context. A comprehensive survey on gesture recognition techniques can be found in (Pisharady & Saerbeck, 2015) as well as detailed overviews in (Asadi-Aghbolaghi et al., 2017; Cheng, Yang, & Liu, 2016a; Escalera et al., 2017). Apart from the application domain, the existing mechanisms either utilize handcrafted features for gesture description or learn the discriminative features from input data using deep learning techniques. We, therefore, organize our discussion on related works along these two axes.

2.1. Handcrafted feature-based gesture recognition

Previous studies on computer vision focused mainly on exploiting spatiotemporal engineered features for video-based gesture recognition (Wang et al., 2009). Typically, these approaches operate in two main phases: feature detection and feature description. The popular feature detectors and descriptors are HOG/HOF (Laptev, 2005), Harris3D (Sipiran & Bustos, 2011), HOG3D (Klaser, Marszaek, & Schmid, 2008), Hessian3D (Willems, Tuytelaars, & Van Gool, 2008), Cuboids (Dollar, Rabaud, Cottrell, & Belongie, 2005), ESURF (Willems et al., 2008), 3D EMoSIFT (Wan, Ruan, Li, & Deng, 2013), HON4D (Oreifej & Liu, 2013) and VS-LBP (Maqueda et al., 2015). In particular, (Yang, Zhang, & Tian, 2012) employed what so-called depth motion maps (DMMs) to capture motion cues from different viewpoints (front, side and top), and then they adopted HOG descriptors for action representation. In a similar manner, (Chen, Jafari, & Kehtarnavaz, 2015) exploited local binary pattern (LBP) in DMMs to characterize local rotation invariant texture information. The input gestures are classified using kernel-based extreme learning machine (KELM) classifier. DMMs can be also generated at segment level such in

(Chen et al., 2016a), where the depth sequence is divided into a set of overlapping segments. Each segment is then represented by DMMs, leading to a multi-temporal DMMs representation that preserve more detailed motion and shape cues. Recently, (Wan, Guo, & Li, 2016a) proposed a feature descriptor, named mixed features around sparse keypoints (MFSK), specifically designed for gesture recognition in RGB-D videos. MFSK is computed from the local patch around every detected keypoint at predefined temporal scales. Histogram of 3D Facets (H3DF) (Zhang & Tian, 2015) is another interesting type of feature descriptors for human action and gesture recognition that explicitly encodes the 3D shape information from depth maps. The encoding is processed by projecting the normal vector onto the three Cartesian orthogonal planes. To account for the variations of gesture motion performed by different subjects, one popular approach is to use dynamic time warping (DTW) (Reyes, Dominguez, & Escalera, 2011), since it provides a simple yet effective temporal matching between sequences of different lengths. On this basic scheme, several extensions have been proposed such the one in (Hernndez-Vela et al., 2014) that uses a probabilistic model with Bag-of-Visual-and-Depth-Words (BoVDW) for gesture segmentation and recognition. A further approach was proposed by (Cheng, Dai, Liu, & Zhao, 2016b), where the DTW was successfully extended to compute the distance from image to class instead of image to image. In (Suk, Sin, & Lee, 2010), the authors proposed a dynamic Bayesian network model for continuous gesture recognition. Their model is based on capturing the gesture motion trajectories in RGB sequences and then converting them to time series signals. Finally, attempts have been made also to go beyond the motion of the 3D joint positions either of the hand (De Smedt, Wannous, & Vandeborre, 2016) or of the whole human body (Althloothi et al., 2014). The basic idea here is to analyze the trajectories of a set of geometric configurations extracted from 3D joint coordinates and exploit it for gesture recognition using a certain classification model like support vector machines (SVM).

Despite the achieved successes, it has been noticed that handcrafted features are unable to take into account all environmental factors simultaneously. Such approaches are designed for specific video sequences in specific application domains, which make them difficult to be generalized to other real-world scenarios (Wang et al., 2009). One more limitation of handcrafted features (especially local features) is the high computational complexity that is unavoidable in both the training and testing phases. Alternatively, deep learning-based methods have provided some solutions to overcome these shortcomings.

2.2. Deep learning-based gesture recognition

With the rapid growth of deep learning and powerful hardware like GPU, deep neural networks (DNNs), and in particular convolutional neural networks (CNNs) (LeCun et al., 1998), have demonstrated more and more superiority in many visual tasks, from image classification (Krizhevsky et al., 2012) or object detection (Girshick et al., 2014) to face recognition (Kang et al., 2017) and pose estimation (Jain et al., 2014). The secret behind the massive usage of CNNs in computer vision is attributed to their ability to learn rich mid-level image representations as opposed to handcrafted low-level features. Because of this, many researchers have successfully extended CNNs model for multimodal gesture recognition in videos (Asadi-Aghbolaghi et al., 2017). In this section, we briefly review some of the recent deep-based studies organized according to the way they treat the temporal information.

2.2.1. 2D convolutional networks

In the context of gesture recognition, 2D CNNs have been popularly exploited. One group of recent work concentrated on applying 2D CNNs on individual frames of gesture video and then aggregating the scores for classification, such the one carried out by Tompson et al. (2014) where a deep learning model was developed for instantaneous hand pose estimation. They extracted 3D joint positions from a set of generated heat-maps and then minimizing an appropriate objective function to align 3D model features to each heat-map position. Koller et al. (2016) proposed an iterative ED-based algorithm that integrates CNNs with Hidden-Markov-Models (HMMs) for weakly supervised learning. Their algorithm allows the labeling of vast amounts of data at the frame-level given only noisy video annotation. Instead of dealing with the whole video sequence for gesture classification, John, Boyali, Mita, Imanishi, and Sanma (2016) proposed to extract the representative frames from the video using deconvolution neural network (DNN). They employed tiled image and binary patterns to train the DNN. The final representative frames are fed into long-term recurrent convolution network for prediction. To obtain better performance, some other studies have exploited both temporal and spatial information and utilize them as a compact representation to train 2D CNNs. These temporal and spatial information can often be modeled as a set of pre-computed motion-based features, such as depth motion map (DMM) (Wang et al., 2016b; Zhang et al., 2017a), static pose map (SPM) (Zhang et al., 2017a), dynamic depth image (DDI) (Wang et al., 2016c), dynamic depth normal images (DDNI) (Wang et al., 2016c), and dynamic depth motion normal images (DDMNI) (Wang et al., 2016c). In particular, Ijjina and Chalavadi (2017) introduced a new temporal template representation that captures the motion information in various temporal regions of the video. This motion representation is computed from both RGB and depth sequences and given as input to 2D CNNs for action classification. In Wu et al. (2016a), a two-stream CNNs model containing spatial and temporal networks is developed for gesture learning. In this approach, depth map frames are mapped to spatial stream CNNs, whereas temporal stream CNNs takes colored optical flow frames as input. Chron, Laptev, and Schmid (2015) computed motion and appearance cues along tracks of human body parts instead of the whole video frames. They then used two distinct CNNs on each body part separately to obtain motion-based and appearance-based deep features, while aggregating them over time to form a video descriptor. Finally, Narayana et al. (2018) proposed a deep architecture composed of 12 residual networks processing different types of localized data. A fusion strategy based on sparse network was also proposed to boost the recognition performance.

2.2.2. 3D convolutional networks

As an alternative approach to directly inferring the temporal information from raw data, Ji et al. (2013) introduced a new model, called 3D CNNs, which perform 3D convolution along both spatial and temporal dimensions at pixel level, using stacks of multiple contiguous frames. On this basic model, several refinements have been proposed in the area of gesture recognition in videos. For instance, Molchanov et al. (2015) proposed a hand gesture recognition system that interleaves depth and intensity channels to build normalized spatiotemporal volumes. These volumes were then used to train two separate 3D CNNs. The authors also utilized an effective spatiotemporal data augmentation to reduce potential overfitting. A multimodal combination of RGB-D and skeleton streams was used in Huang et al. (2015) by developing a deep model that takes multiple data as input. The authors showed that their method outperforms the classical baselines such as Gaussian mixture model with HMM. The work carried out by Liu et al. (2016a) employed 3D CNNs to learn high-level features from depth sequences and the vector calculating component to extract joint features. Both feature sets are classified using SVM and the results are fused for final prediction. Camgoz, Hadfield, Koller, and Bowden (2016) adopted the model

of Tran et al. (2015) to train an end-to-end 3D CNNs and applied it to large-scale continuous user-independent gesture recognition. The model of Tran et al. (2015) was also used in the work of Zhu et al. (2016a), but this time with pyramid input to preserve the multi-scale contextual information of gestures. Miao et al. (2017) proposed a deep model that leverages the advantages of both ResNets and 3D CNNs models to extract deep features from different modalities. A fusion scheme based on canonical correlation analysis was also proposed to blend the extracted features. The classification of gestures was achieved using SVM classifier. Finally, Lin et al. (2016) proposed a deep structured model to adaptively decompose an activity instance into temporal parts using 3D CNNs and latent structured model. They also integrated a radius-margin regularization scheme with the deep model to effectively conduct the classification with good generalization performance.

2.2.3. Recurrent networks

The application of temporal sequence modeling techniques, such as Recurrent Neural network (RNN) and LSTM, for gesture recognition showed promising results in recent years. We are aware of Chai et al. (2016) where they proposed a multi-stream RNN for large scale gesture spotting from RGB-D modalities. The hand gesture is firstly isolated from the background based on a hand detector model trained from Faster R-CNN. The isolated gesture is feed into two paralleled simple RNN layers, which are then fused by a fusion layer. An LSTM layer is used right after the fusion layer to model the contextual information of the temporal gesture sequences. Nishida and Nakayama (2015) proposed another multi-stream model, called MRNN, which extends RNN capabilities with LSTM cells in order to facilitate the handling of variable-length gestures. A hierarchical RNN model is proposed in Du, Wang, and Wang (2015) for action recognition using skeleton data. The skeleton is divided into five parts, each of which is feed into different subnets. The outputs of the subnets are fused into higher-layer RNN and then feed into a single-layer perceptron for the final decision. Skeleton information was also used by Zhu et al. (2016b) to learn the co-occurrence features of skeleton joints using LSTM network. In Song, Lan, Xing, Zeng, and Liu (2017), the authors proposed an end-to-end spatial and temporal attention model for gesture recognition from skeleton data based on RNN with LSTM. They also proposed a regularized cross-entropy loss to drive the model learning process and developed a joint training strategy accordingly. Finally, Liu et al. (2016b) extended the traditional LSTM into two concurrent domains, i.e., spatiotemporal LSTM. To encode the spatiotemporal context, they developed a tree structure where each joint of the network receives contextual information from both neighboring joints and previous frame.

2.2.4. Heterogeneous networks

The idea of combining multiple models for gesture recognition has been extensively investigated by many other approaches. For instance, Wu et al. (2016b) proposed a semi-supervised hierarchical dynamic framework for continuous gesture segmentation and recognition on multimodal data, including color, depth and skeleton information. Deep belief networks were used for the processing of skeleton features, whereas, 3D CNNs were employed to learn from RGB-D data. The whole model was integrated with HMM to incorporate temporal dependencies. In Wang, Song, Han, and Cheng (2016a), CNN network is exploited to extract spatial features of each frame, and then these features are summed over by sequentially supervised LSTM network for temporal feature learning. Duan, Zhou, Wan, Guo, and Li (2016) combined two deep networks for multimodal gesture recognition, two-stream consensus voting network for modeling RGB stream and optical flow, while the saliency stream and depth stream are handled by depth-saliency network. The authors also adopted late fusion strategy for final scoring. In the recent work (Molchanov et al., 2016), the authors developed a deep model for online gesture classification by integrating 3D CNNs with recurrent mechanism. 3D CNNs are employed to extract local spatiotemporal features from depth, color, and stereo-IR data streams. These features are then fed into a RNN which aggregates transitions across several clips. 3D CNNs can be also combined with LSTM to model the long short-term spatiotemporal dependencies (Zhang et al., 2017b; Zhu et al., 2017), where short-term spatiotemporal features of gestures are first learned through 3D CNNs, followed by learning long-term spatiotemporal features using ConvLSTM network. Spatial pyramid pooling is then used to normalize the spatiotemporal features for gesture spotting. Not so far from combining CNNs with LSTM, the authors in (Wang et al., 2017b) adopted two types of networks, namely 3D ConvLSTMs to recognize gestures in videos and ConvNets to recognize gestures from dynamic images constructed by rank pooling. Both networks were applied at two spatial levels: body and hands.

The proposed gesture recognition approach in this paper can be also regarded as a heterogeneous model, where 3D-CDCN performs 3D ResNets and ConvLSTM to directly learn spatiotemporal features from both RGB and depth sequences, on the other hand, 2D-MRCN uses 2D ResNets to learn high-level features from motion representation based on iMHI and iDMM. Both models learn spatiotemporal simultaneously, but the extracted features are different. The multi-dimensional fusion strategy between them makes use of the full advantages embedded in different kinds of features, which allows considerably improving the recognition results compared to individual models.

3. Proposed MultiD-CNN framework

In this section, we describe our MultiD-CNN framework for multimodal gesture recognition. As depicted in Fig. 1, MultiD-CNN is composed of two main models, 3D Color-Depth Convolutional Network (3D-CDCN) and 2D Motion Representation Convolutional Network (2D-MRCN). Both models are based on Deep Residual Networks (ResNets) (He et al., 2016), which generally has two main advantages. From one aspect, the convolutional layers in the ResNets significantly reduce the number of trainable parameters using the concept of weights sharing. This is particularly useful to overcome the usual problem of over-fitting. Also, they have the capability of finding stable and invariant features, which is helpful in some situations where there are a lot of repeatable features. From another aspect, the ResNets introduces shortcut connections that perform identity mapping, and directly add the output of a particular layer to the output of later layers, thus effectively avoids signal attenuation caused by multiple stacked non-linear transformations. Consequently, deeper network can be constructed with ResNets while faster training speed can be achieved. As will become clear in later sections. 3D-CDCN performs 3D ResNets and ConvLSTM to simultaneously learn spatiotemporal features from both RGB and depth streams, whereas, 2D-MRCN uses 2D ResNets to learn high-level features from motion representation based on iMHI and iDMM. Scores from both models are later fused using a simple linear combination to get the final prediction results. Each of the above models is described in more detail in the following subsections.

3.1. 3D Color-Depth Convolutional Network

As already pointed out in the introduction section, we investigate the generalization capability of our system by treating the motion information as a feature channel during the learning process. Here we base our architecture of 3D-CDCN on three main components: 3D ResNets-based spatiotemporal feature learning,

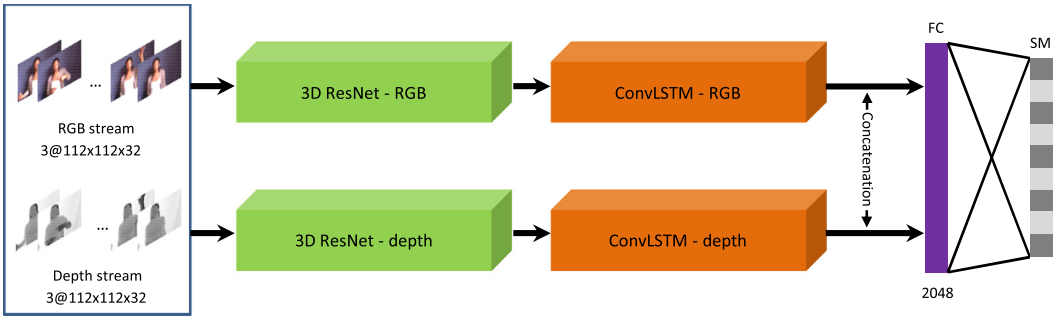


Fig. 2. Overview of 3D-CDCN model. The inputs to the model are $112 \times 112 \times 32$ fixed size cuboids originated from RGB and depth sequences. Each stream is composed of two consecutive networks: 3D ResNets for spatiotemporal features extraction and ConvLSTM to model the spatiotemporal correlation. Individual streams are trained for each data modality and then fused using a concatenation layer followed by a fully-connected layer (FC). The output of the network is class-membership probabilities $\mathcal{P}_{CD}(\mathcal{C}_r|\mathcal{X}^{CD})$ obtained using softmax classifier (SM).

ConvLSTM-based spatiotemporal correlation modeling, and a multimodal fusion model for score prediction. As depicted in Fig. 2, the inputs to the 3D-CDCN are cuboids generated from RGB and depth sequences after being preprocessed and normalized to the same benchmark volume size of $112 \times 112 \times 32$. Note that each individual stream in 3D-CDCN is responsible for a data modality (RGB or depth). We also recall that the channel of each data modality is set to be 3. Since depth frames are originally grayscale images, we simply copy the information of single-channel twice to construct a three-channel depth video, whereas, three-channel of color information are investigated for RGB stream. As a first step in our process, 3D ResNets are performed to simultaneously learn spatiotemporal features for each data modality. This is achieved by performing 3D convolution and pooling operations (Ji et al., 2013) on stacks of multiple adjacent video frames (video clips) in both spatial and temporal domains. To further learn the spatiotemporal dependencies between the extracted features, we employ a recurrent neural network with ConvLSTM cells which are connected to the output of the underlying 3D ResNets. In multimodal fusion stage, the output of ConvLSTM from both RGB and depth streams are then fused to estimate the class-membership probabilities of the gesture.

From now on, and for simplicity, we denote the input video clip with a volume size of $w \times h \times l$, where w and h are respectively the width and the height of the video frames, and l is the length of the video clip. We also refer 3D convolution and 3D pooling kernel size by $k \times k \times d$, where k is the spatial dimension of the kernel and d is the kernel temporal depth.

3.1.1. 3D ResNets architecture

Inspired by the recent successes of Deep Residual Networks (ResNets) (He et al., 2016) in numerous challenging visual recognition tasks, we adopt a model that combines the merits of 3D CNN and the residual concept, namely 3D ResNet17, to effectively extract the spatiotemporal features from input data. Compared to other feature learning schemes, 3D CNN has the ability to encode temporal information alongside with spatial information without requiring an extra temporal modeling. Given a fixed-size cuboid of $w \times h \times l$ video clip, denoted as $\mathcal{V}$. The 3D convolution is performed by spatiotemporally convolving the input volume $\mathcal{V}$ with a learned 3D filter $\mathcal{W}_s$, where s is the index over the set of learned filters. More formally, the feature value ϕ_s at position (x, y, t) computed using the learned 3D filter $\mathcal{W}_s$ is given by:

$$\phi_s(x, y, t) = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} \sum_{p=0}^{d-1} \mathcal{V}(x+m, y+n, t+p) \cdot \mathcal{W}_s(m, n, p) \qquad (1)$$

where x and y represent the pixel position within the frame t. It is worth emphasis to recall that the output of 3D convolution is still three-dimensional, allowing to maintain the spatiotemporal arrangement of the learned features. Basically, a ResNets network consists of a set of stacked residual blocks, and each residual block could be generally given by:

$$z_{g+1} = z_g + \mathcal{F}(z_g) \qquad (2)$$

where z_g and z_{g+1} denote respectively the input and the output of the g^{th} residual block. $\mathcal{F}(z_g)$ represents the result of performing the residual function $\mathcal{F}$ over the input z_g. The form of the residual function $\mathcal{F}$ is flexible and can represent multiple 3D convolutional layers. The shortcut connection $z_g + \mathcal{F}(z_g)$ performs identity mapping. This is done by adding the output of a residual block element-by-element to the output of the following residual block. The underlying principle of ResNets is to learn the additive residual function $\mathcal{F}$ with reference to the input unit z_g, thus enabling the following block to learn a residual mapping rather than a full mapping. This makes the learning process for deeper architectures easier, while better performance can be achieved.

The complete deep architecture of 3D ResNet17 employed in this study is illustrated in Fig. 3, which is similar in design to the Res3D model (Tran, Ray, Shou, Chang, & Paluri, 2017). The whole network is composed of 8 residual learning blocks and 5 individual 3D convolutional layers. First, the input to the network is spatially down-sampled in conv1 with 3D convolutional kernel of size $7 \times 7 \times 3$ and stride of $2 \times 2 \times 1$, while producing 64 feature maps of size $56 \times 56 \times 32$. The output of this layer is then fed into 8 sequential residual blocks. Each block consists of two consecutive convolutional layers with 3D kernels of size $3 \times 3 \times 3$ and stride of $1 \times 1 \times 1$. As in the Res3D (Tran et al., 2017), the numbers of filter response maps for both convolutional layers in the 8 residual blocks are 64, 64, 128, 128, 256, 256, 512 and 512, respectively.

In order to prevent data from gradient diffusion, the batch normalization layer is utilized right after each convolutional layer. The rectified-linear-unit (ReLU) activation function is applied next to boost the power of statistical modeling. A shortcut pass connects the top of the block to the layer just before the last ReLU in the block. Spatiotemporal down-sampling is performed by conv3a1, conv4a1, and conv5a1 with 3D convolutional kernels of size $1 \times 1 \times 1$ and stride of $2 \times 2 \times 2$. Note that different from Res3D (Tran et al., 2017), we do not include the spatial global average pooling layer in order to preserve the spatial information for the next learning stage. Finally, the output of the network is 512 3D spatiotemporal feature maps of size $7 \times 7 \times 4$. These resulting feature maps will be then fed into ConvLSTM network to further learn the spatiotemporal dependencies.

3.1.2. ConvLSTM architecture

From the fact that the variations between sequential data may encode additional information which could be useful in making

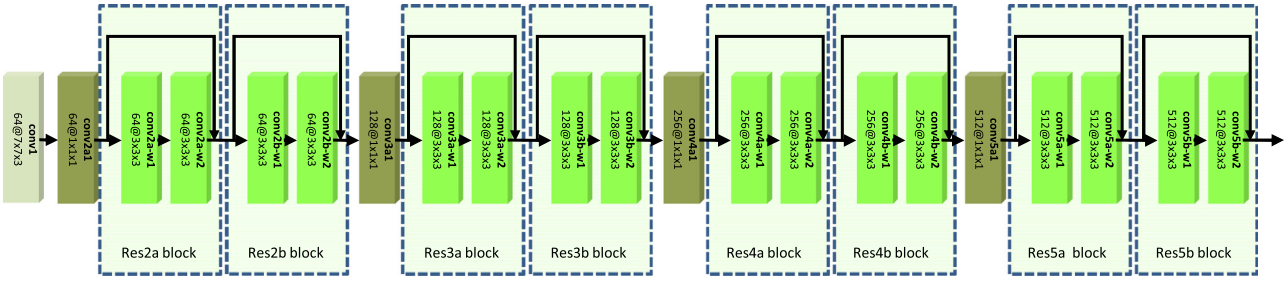


Fig. 3. Architecture of 3D ResNet17 network used for spatiotemporal features extraction. It consists of 8 3D residual learning blocks and 5 individual 3D convolutional layers. The design of 3D ResNet17 is similar to that of Res3D in (Tran et al., 2017).

more accurate predictions, we propose using Long Short-Term Memory network (LSTM), particularly Convolutional LSTM (ConvLSTM) (Xingjian et al., 2015), to model the spatiotemporal dependencies between feature maps of the input sequence. LSTM is a special type of Recurrent Neural Networks (RNNs) that defines a recurrent hidden state whose activation at each time is dependent on that of the previous time. The major innovation of LSTM is memory cell which essentially acts as an accumulator of the state information, allowing it to better discover complex long-range temporal relationships while maintaining short time lags. Differently from the traditional Fully-Connected LSTM (FC-LSTM) which takes the vectorized features as input to learn only temporal features, ConvLSTM explicitly assumes that the input is sequence of images and replaces the vector multiplication in FC-LSTM gates by convolutional operations, in which the intermediate representations of the images preserve the spatial correlation information during the recurrence. As a result, the convolution and recurrent operations in both input-to-state and state-to-state transitions can take full usage of the spatiotemporal relationships information.

More precisely, given an input sequence $\mathsf{x}_1, \ldots, \mathsf{x}_T$, let $\mathsf{c}_1, \ldots, \mathsf{c}_T$ be the cell activation states, and let $\mathsf{h}_1, \ldots, \mathsf{h}_T$ be the hidden states. The ConvLSTM unit is similar to the one of FC-LSTM, the only difference is that the fully connected layers in each gate are replaced by convolutions, as specified by the following update equations:

$$\begin{aligned}
\mathsf{i}_t &= \sigma\left(\mathcal{W}_\mathsf{i} * \mathsf{x}_t + \mathcal{U}_\mathsf{i} * \mathsf{h}_{t-1} + \mathsf{b}_\mathsf{i}\right)\\
\mathsf{f}_t &= \sigma\left(\mathcal{W}_\mathsf{f} * \mathsf{x}_t + \mathcal{U}_\mathsf{f} * \mathsf{h}_{t-1} + \mathsf{b}_\mathsf{f}\right)\\
\mathsf{o}_t &= \sigma\left(\mathcal{W}_\mathsf{o} * \mathsf{x}_t + \mathcal{U}_\mathsf{o} * \mathsf{h}_{t-1} + \mathsf{b}_\mathsf{o}\right)\\
\mathsf{c}_t &= \mathsf{f}_t \circ \mathsf{c}_{t-1} + \mathsf{i}_t \circ \tanh\left(\mathcal{W}_\mathsf{c} * \mathsf{x}_\mathsf{t} + \mathcal{U}_\mathsf{c} * \mathsf{h}_{t-1} + \mathsf{b}_\mathsf{c}\right)\\
\mathsf{h}_t &= \mathsf{o}_t \circ \tanh\left(\mathsf{c}_t\right)
\end{aligned} \tag{3}$$

Here, $*$ represents convolution operator and $\circ$ denotes element-wise product operator. $\sigma(\cdot)$ and $\tanh(\cdot)$ are respectively logistic sigmoid and hyperbolic tangent functions. $\mathsf{i}_t, \mathsf{f}_t$, and o_t represent respectively the input gate, forget gate, and output gate. Whereas, $\mathsf{b}_\mathsf{i}, \mathsf{b}_\mathsf{f}, \mathsf{b}_\mathsf{o}$, and b_c are the bias terms. The input weights $\mathcal{W}_\sim$ and hidden weights $\mathcal{U}_\sim$ represent the trained 2D convolutional kernels of ConvLSTM unit. In general, the data stored in the cell state c_t is maintained unless it is added to a new input by the input gate i_t or forgotten by the forget gate f_t. The output gate o_t controls the emission of the memory data from the cell state c_t to the final state h_t.

The deep architecture of ConvLSTM employed in the proposed system is depicted in Fig. 4. We use a deep architecture in which the output from one ConvLSTM layer is input for the next layer. We experimented with up to three ConvLSTM layers, and we found that with only one layer the proposed system works sufficiently well. The output from the final layer of 3D ResNet17 network is firstly decomposed into a sequence of four consecutive 2D feature maps of spatial size 7×7. We then use four ConvLSTM units, where each 2D feature map is given as input to the ConvLSTM at different time steps. The convolutional kernel spatial sizes of the input weights and hidden weights are set to 3×3 with stride of 1×1. The number of convolutional kernels is set to 512. The same padding of 1 pixel in both spatial dimensions is used in all the ConvLSTM units in order to preserve the same spatial size of spatiotemporal features the during convolution process. As we are concentrated in this paper on isolated gestures, only the output made by ConvLSTM after processing the complete sequence of 2D feature maps is considered as the final long short-term spatiotemporal features of the gesture. More specifically, the output of ConvLSTM network is spatiotemporal features with the spatial size of 7×7 which is the same as that of the input to ConvLSTM, whereas its temporal length is reduced to 1.

3.1.3. Multimodal fusion

Information fusion is a fundamental aspect in multimodal gesture recognition. We distinguish between three multimodal fusion strategies: fusion at data level (i.e. early fusion), fusion at feature level (i.e. intermediate fusion) and fusion at decision level (i.e. late fusion). Early fusion requires multimodal data to have some consistent characteristics, which is not the case in our approach as the input data have different structural properties. On the other hand, late fusion may lead to loss of information since it only concerns about voting. This can be more suitable if different classifiers are involved. In contrast, as both RGB and depth models are architecturally similar, we investigate the high-level representation learned by each module and we adopt the intermediate fusion which processed at feature level. Compared to other fusion schemes, fusion at feature level can be more effective owing to the rich complementary information of each kind of the employed features. To do this, the outputs from ConvLSTM of both RGB and depth streams are appended in a concatenation layer, as shown in Fig. 2. To further learn non-linear combinations of these features, the resulting feature map is propagated up to a fully-connected layer (FC) with 2048 neurons. Finally, a softmax layer (SM) with c output (i.e. number of classes) is used for gesture classification. The (SM) layer estimates a class-membership probability $\mathcal{P}_{CD}\left(\mathcal{C}_r|\mathcal{X}^{CD}\right)$ for each gesture label $\mathcal{C}_{r_{1\le r\le c}}$, which can be computed as follows:

$$\mathcal{P}_{CD}\left(\mathcal{C}_r\middle|\mathcal{X}^{CD}\right) = \frac{\exp\left(\mathcal{X}^{CD}_{\mathcal{C}_r}\right)}{\sum_{q=1}^{|\mathcal{X}^{CD}|}\exp\left(\mathcal{X}^{CD}_{q}\right)} \tag{4}$$

where $\mathcal{X}^{CD}$ is the output feature vector of a given gesture as predicted by 3D-CDCN.

3.2. 2D Motion Representation Convolutional Network

In this section, we describe our architecture for 2D-MRCN (shown in Fig. 5) which is based on spatiotemporal motion representation and 2D ResNets (He et al., 2016). Our motivation to use motion representation here is the ability to exploit the domain knowledge of the system by accumulating the whole motion characteristics of the sequence in a single image. Overall, 2D-MRCN first computes the improved Motion History Image (iMHI)

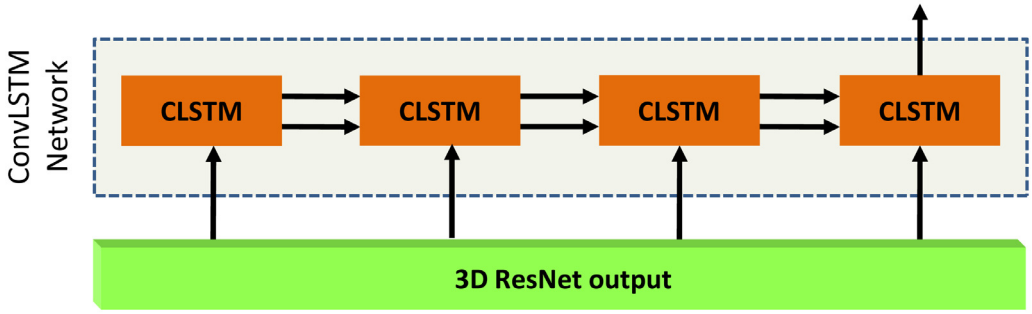


Fig. 4. Architecture of ConvLSTM network used for learning the spatiotemporal correlations. This network takes as input the output from the final layer of 3D ResNet17, which is a sequence of 4 consecutive 2D feature maps of spatial size 7×7. Each 2D map is given as input to the ConvLSTM at different time steps. Only the prediction made by ConvLSTM after processing the complete sequence of 2D feature maps is propagated up for multimodal feature fusion.

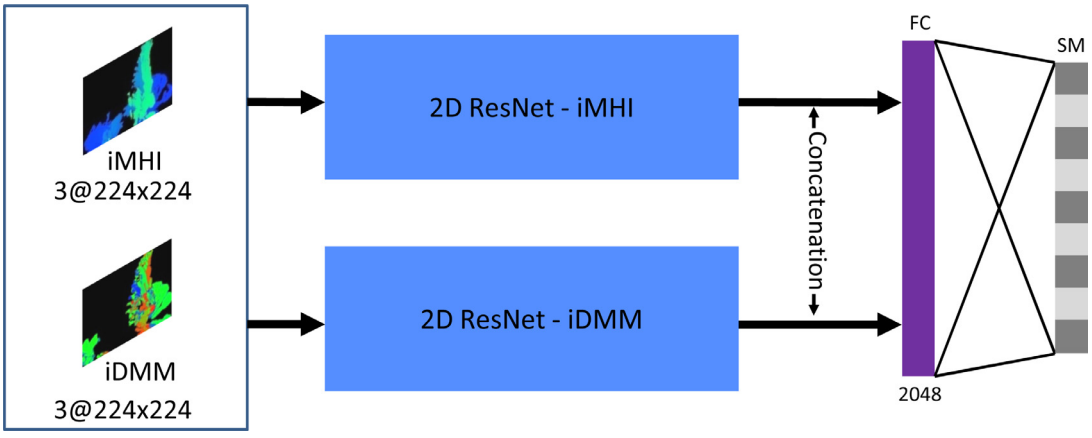


Fig. 5. Overview of 2D-MRCN model. The inputs to the network are the motion representation images iMHI and iDMM after being normalized to a fixed size of 224×224. Each input image is passed through a separate stream based on 2D ResNets to extract high-level deep features. To overcome the limitations of individual modalities, the features from all streams are then fused and fed into a softmax layer (SM) to estimate class-membership probabilities $\mathcal{P}_{MR}(C_r|\mathcal{X}^{MR})$ of the gesture.

and the improved Depth Motion Map (iDMM) from RGB and depth sequences, respectively. A two-stream architecture based on 2D ResNets is then employed to learn the deep features from these two motion images and combine them to exploit their complementary information.

3.2.1. Motion representation

Another interesting way to represent the gesture is by using the concept of motion representation which attempts to encode the motion information through the whole video sequence in a single 2D representation. Here, we base our motion representation on the well-known MHI and DMM images because of their high representativeness and reputability. However, these motion images suffer from noise issue caused by several factors such as low resolution of the cameras, jumbled objects, and cluttered backgrounds. We then propose in this subsection a method to remove these noises, hence improving the quality of both MHI and DMM images.

Improved Motion History Image: Motion History Image (MHI), initially introduced by (Bobick & Davis, 2001), is a visual-based temporal template method that records a history of temporal changes at each pixel location across a video sequence. MHI is simple but efficient in encoding the spatial distribution of movements by using the intensity of every pixel in a temporal order. In this work, we compute the MHI only for RGB modality. The procedure for computing MHI starts by condensing the RGB image sequence into grayscale images, while dominant motion information is preserved in a compact manner. This also would better make MHI less sensitive to light change and silhouette noises. Specifically, given a grayscale image sequence $I(x, y, t)_{t=0,\ldots,N-1}$ with N frames, and $m_I(x, y, t)_{t=0,\ldots,N-2}$ be the corresponding binary image sequence indicating the regions of motion (i.e. each binary frame $m_I(x, y, t)$ represents the motion between two consecutive frames $I(x, y, t)$ and $I(x, y, t+1)$). The MHI of $I(x, y, t)_{t=0,\ldots,N-1}$ can be computed as follows:

$$MHI = \sum_{t=0}^{N-2} \psi_t \cdot m_I(x, y, t) \tag{5}$$

where ψ_t is an assigned grayscale weight whose value varies between 0 and 255:

$$\psi_t = (t+1) \cdot \frac{255}{N} \tag{6}$$

Here, ψ_t is a linearly increased function with time, meaning that oldest motion information has the lowest significance, whereas recent motion information has the highest significance. Typically, the binary image sequence $m(x, y, t)_{t=0,\ldots,N-2}$ is computed using image subtraction between consecutive frames based on a pre-determined threshold δ:

$$m_I(x, y, t) = \begin{cases} 1 & if \ \left|I(x, y, t) - I(x, y, t+1)\right| \geq \delta \\ 0 & otherwise \end{cases} \tag{7}$$

However, identifying the region of motion using pixel-wise difference can bring some undefined moving regions caused by slight body shaking or objects boundaries. These undefined moving regions are considered as noises in the image sequence $m_I(x, y, t)_{t=0,\ldots,N-2}$, which will cause poor quality of MHI. To overcome this, a simple but efficient strategy is proposed to remove these noises from the sequence $m_I(x, y, t)_{t=0,\ldots,N-2}$, resulting in an improved version of MHI (iMHI). The proposed strategy relies on a 2D spatial sliding window placed at each pixel location (x, y, t) to

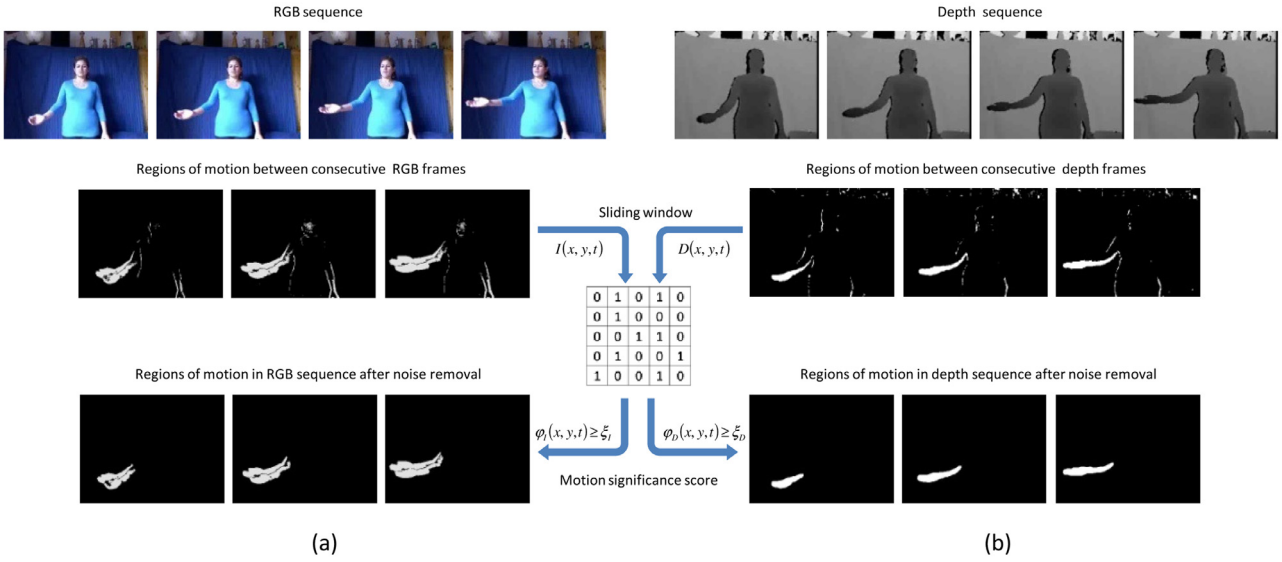


Fig. 6. Process of removing the fake moving regions from motion representation of (a) RGB video sequence and (b) depth video sequence. Pixels with low motion significance score are considered as noisy moving pixels and removed by assigning the background intensity value.

decide whether this pixel is a moving part of the human body or it is an unconnected fake moving pixel. Notice that in contrast to the suspect moving pixels, the real motion of human body generates a large region with non-zero pixels in the binary image $m_I(x, y, t)$. In the other words, as much as the movement at a pixel location (x, y, t) is considered significant as much as the number of other relative moving pixels within the 2D sliding window is higher, and vice-versa. We then compute a motion significance score $\varphi(x, y, t)$ for each moving pixel in $m_I(x, y, t)$ to evaluate how much this movement is significant as follows:

$$\varphi_I(x,y,t)=\frac{1}{(\Delta_I+1)^2}\cdot\sum_{i=x-\frac{\Delta_I}{2}}^{x+\frac{\Delta_I}{2}}\sum_{j=y-\frac{\Delta_I}{2}}^{y+\frac{\Delta_I}{2}} m_I(i,j,t) \tag{8}$$

where Δ_I defines the height and the weight of the 2D sliding window. In our experiments, we set $\Delta_I = 6$. Note that the motion significance score $\varphi_I(x, y, t)$ can take value in the range [0, 1], where the fake moving pixels will have values close to 0. Thus, we formulate the computation of binary image sequence $m_I(x, y, t)_{t=0,\ldots,N-2}$ by removing the noisy regions that have the motion significance score less than a pre-defined threshold ξ_I:

$$m'_I(x,y,t)=\begin{cases}1 & if\ m_I(x,y,t)=1\quad and\quad \varphi_I(x,y,t)\geq\xi_I\\ 0 & otherwise\end{cases} \tag{9}$$

We experimentally noticed that values of $\delta = 40$ and $\xi_I = 0.5$ are shown to work sufficiently well. Fig. 6(a) shows clearly the disparity between the old binary image sequence $m_I(x, y, t)_{t=0,\ldots,N-2}$ and the new one $m'_I(x, y, t)_{t=0,\ldots,N-2}$ for a sample gesture. Eventually, the iMHI is then obtained by using Eq. (5) taking into consideration the new binary image sequence $m'_I(x, y, t)_{t=0,\ldots,N-2}$. The result of this computation is a clean grayscale image where more recently moving pixels are brighter, and vice-versa. Examples of MHI and iMHI for some sample gestures from ChaLearn IsoGD dataset are illustrated in Fig. 7(a). It is clearly visible that iMHI return only real motion of moving body parts, whereas, the standard MHI contains a lot of noises that degrade its quality.

Improved Depth Motion Map: To further exploit the additional motion information from depth sequence, we use the concept of Depth Motion Map (DMM), which was originally introduced by (Yang et al., 2012). DMM is a visual representation of human activities generated by accumulating the motion energy through the entire depth sequence. The concept of DMM was also considered in (Chen, Liu, & Kehtarnavaz, 2016b) where the motion energy is calculated by accumulating the absolute difference between consecutive frames instead of taking the thresholded difference (Yang et al., 2012). This would better preserve subtle motion information. Therefore, in this work we use the same concept as in (Chen et al., 2016b). Specifically, let $D(x, y, t)_{t=0,\ldots,N-1}$ be a depth video sequence with N number of frames, its DMM is given by:

$$DMM=\sum_{t=0}^{N-2}\left|D(x,y,t+1)-D(x,y,t)\right| \tag{10}$$

DMM can effectively capture the shape and motion cues of a depth sequence, resulting in a spatial energy distribution map that discriminatively represents a gesture. However, because of low resolution and unstable reflection of depth cameras, most of depth data come with noticeable depth noises that significantly degrade the quality of the video and can cause some undefined energy regions in DMM. In particular, some regions are sometimes missing from one frame to another frame; there are also ghost shadows around object boundaries (pixels with undefined depth values). Besides, the occurred small body shaking between consecutive frames can also bring some false moving edges to DMM. These narrow moving edges are sensitive to body size and do not provide any useful information to help in distinguishing between gestures. In order to remove these noises, some image processing techniques including median filter and morphological operations can be applied to DMM such in (Zhang et al., 2017a), but at the risk of missing the original motion information contained in DMM. Since the noises accompanied with DMM share almost the same properties with that of MHI, we instead adopt the strategy of motion significance score described above to remove them. We call the resulting motion representation as improved DMM (iDMM). Specifically, for each difference map $Diff(x, y, t)$ between two consecutive depth frames $D(x, y, t)$ and $D(x, y, t + 1)$, we first build a binary image $m_D(x, y, t)$ indicating the regions of motion in the following way:

$$m_D(x,y,t)=\begin{cases}1 & if\ D(x,y,t)\neq D(x,y,t+1)\\ 0 & otherwise\end{cases} \tag{11}$$

Notice that the way we detect the regions of motion here is different from that of iMHI. Based on a sliding window of size $(\Delta_D + 1) \times (\Delta_D + 1)$, we then compute the motion significance

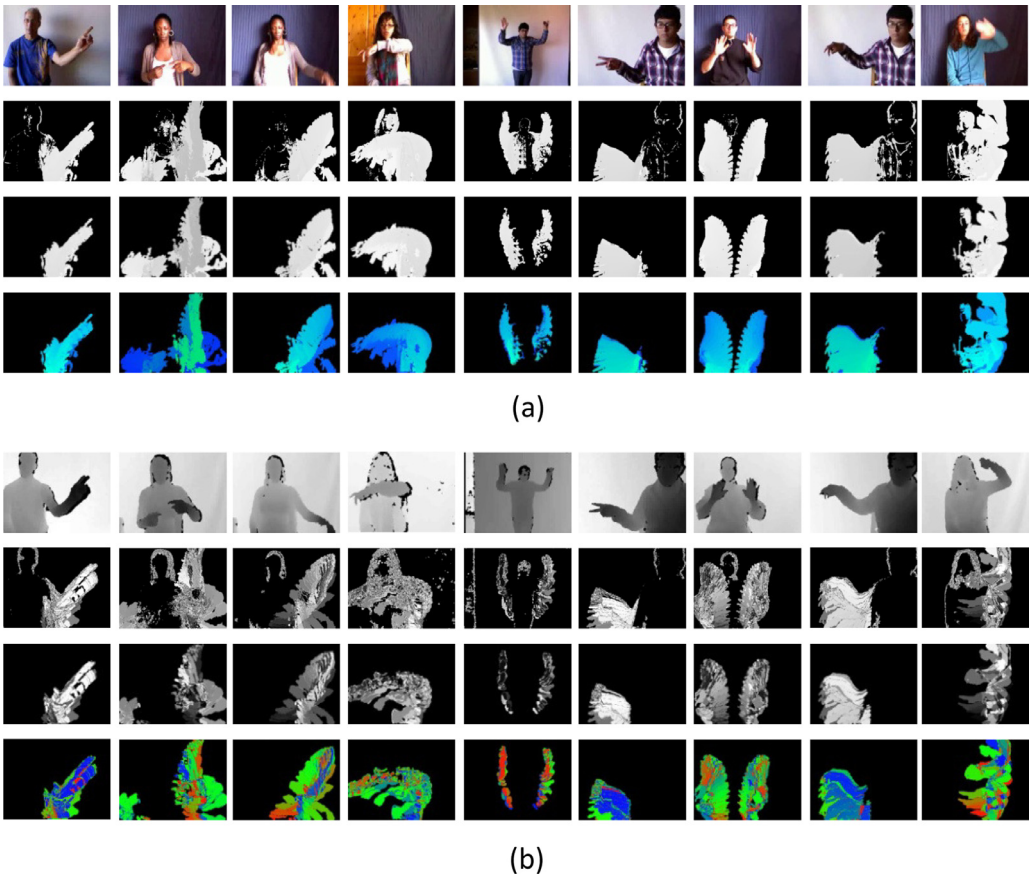


Fig. 7. Examples of motion representation for some sample gestures from ChaLearn IsoGD dataset, each column represents a gesture class: (a) from top to bottom: RGB gesture keyframe, traditional MHI, iMHI after noise removal, pseudo-colored iMHI; (b) from top to bottom: depth gesture keyframe, traditional DMM, iDMM after noise removal, pseudo-colored iDMM.

score $\varphi_D(x, y, t)$ for each moving pixel (pixel with non zero value) in the binary image $m_D(x, y, t)$ using Eq. (8). The pixels that have motion significance score $\varphi_D(x, y, t)$ less than the predefined threshold ξ_D are considered as noisy moving pixels and deemed as background in the corresponding difference map $Diff(x, y, t)$:

$$Diff(x,y,t) = \begin{cases} \left|D(x,y,t+1) - D(x,y,t)\right| & if\ \varphi_D(x,y,t) \geq \xi_D \\ 0 & otherwise \end{cases} \tag{12}$$

Throughout our experiments, we set $\Delta_D = 8$ and $\xi_D = 0.6$. An example of difference maps for a sample gesture after noise removal is illustrated in Fig. 6(b). At last, the calculation of iDMM can be expressed as:

$$iDMM = \sum_{t=0}^{N-2} Diff(x,y,t) \tag{13}$$

Fig. 7(b) shows some examples of traditional DMM and iDMM generated for sample gestures from ChaLearn IsoGD dataset. We can see that compared to traditional DMM, our iDMM provide a clean visually plausible spatial energy distribution map while preserving the original motion information.

Pseudo-coloring: From a human perceptual point of view, visualizing slight differences in gray values is very difficult and can be facilitated with the use of color space, which displays gray levels according to a set map of colors rather than a single intensity. While the human eye can only differentiate up to 100 grayscale values in a scene, it can easily distinguish a wide range of color hues (Johnson, 2012). It is thus of interest to use a color-coding scheme to enhance the perceptual capabilities of the human visual system and gather more information from gray images (Wang et al., 2016b). The result is that small differences in gray values can be more distinguished even across a wide dynamic range. Motivated by this, we propose in this paper to transform the iMHI and iDMM into pseudo-colored images before feeding them to 2D ResNets. We then adopt the well-known power rainbow transform (PRT), proposed by (Abidi, Zheng, Gribok, & Abidi, 2006), to perform this task. The PRT can display increasing segments of grayscale values as shades of red, green, and blue cues (RGB color space). This allows significantly intensifying and enhancing the texture of motion patterns in iMHI and iDMM images to better represent the spatiotemporal information of gestures. Formally, for a given gray intensity $\mathcal{I}$, its corresponding RGB normalized color code $(\mathcal{I}_R, \mathcal{I}_G, \mathcal{I}_B)$ can be obtained through PRT as follows:

$$\begin{cases} \mathcal{I}_R = \left[\frac{1}{2} \cdot \left(1 + \cos\left(\frac{4\pi}{3{\cdot}255} \cdot \mathcal{I}\right)\right)\right]^{\alpha} \\ \mathcal{I}_G = \left[\frac{1}{2} \cdot \left(1 + \cos\left(\frac{4\pi}{3{\cdot}255} \cdot \mathcal{I} - \frac{2\pi}{3}\right)\right)\right]^{\alpha} \\ \mathcal{I}_B = \left[\frac{1}{2} \cdot \left(1 + \cos\left(\frac{4\pi}{3{\cdot}255} \cdot \mathcal{I} - \frac{4\pi}{3}\right)\right)\right]^{\alpha} \end{cases} \tag{14}$$

where α is the power parameter, its value can be chosen to vary the image contrast. In our work we set $\alpha = 2$. To encode the iMHI and iDMM, we respectively multiply each normalized color channel of $\mathcal{I}_R, \mathcal{I}_G$, and $\mathcal{I}_B$ by a constant factor of $\frac{255}{MAX_R}$, $\frac{255}{MAX_G}$, and $\frac{255}{MAX_B}$, where MAX_R, MAX_G and MAX_B represent the maximum value of $\mathcal{I}_R$, $\mathcal{I}_G$, and $\mathcal{I}_B$, respectively. In this way, all intensity values of iMHI and iDMM are scaled within the range $[0, \ldots, 255]$. Some examples of the resulting pseudo-colored iMHI and iDMM images are illustrated in Fig. 7. We can observe that the motion patterns in

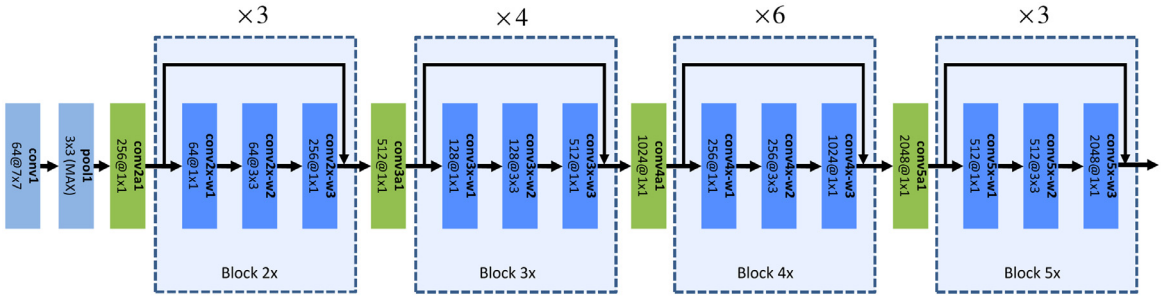


Fig. 8. Architecture of 2D ResNet49 network used for deep features extraction from pseudo-colored iMHI and iDMM images. It consists of 16 2D bottleneck residual blocks, five individual 2D convolutional layers and one 2D pooling layer. The layer configuration of 2D ResNet49 is similar to that of (He et al., 2016).

pseudo-colored images can be easily and effectively distinguished compared to the grayscale version.

3.2.2. 2D ResNets architecture

Once the motion representation (pseudo-colored iMHI and iDMM images) is computed from video observations, the next stage is to extract the features from this motion representation and use them for gesture prediction. We then propose a two-stream 2D architecture to learn high-level deep features from iMHI as well as the iDMM. Each individual stream is separately trained on a data modality after being resized to a fixed size of 224×224. We investigate the three-channel of RGB color information for each data modality. The layer configuration of each stream follows that of ResNet50 in (He et al., 2016). As depicted in Fig. 8, it consists of 16 2D bottleneck residual blocks, five individual 2D convolutional layers and one 2D pooling layer. Specifically, the first layer (conv1) in the network consists of 64 feature maps of size 112×112. This layer is obtained by convolving 2D kernels of size 7×7 and stride of 2×2 with the input image, followed by subsampling (pool1) each feature map with a 2D max-pooling kernel of size 3×3 and stride of 2×2. The output of these layers is then fed into 16 stacked 2D residual blocks. Each residual block contains three consecutive convolutional layers with respectively 2D kernels of size 1×1, 3×3 and 1×1. Similar to (He et al., 2016), the number of output feature maps for the blocks of types $(2\times)$, $(3\times)$, $(4\times)$ and $(5\times)$ are set to be 256, 512, 1024, and 2048, respectively. Each convolutional layer in each residual block is followed by batch normalization and a ReLU. A shortcut pass connects the top of the block to the layer just before the last ReLU in the block. To maintain less memory usage, spatial down-sampling is performed by conv3a1, conv4a1, and conv5a1 with a stride of 2×2. After multiple stages of convolution and pooling operations, we are now able to extract discriminative deep features with a 2D spatial size of 7×7, which are then considered as the ultimate spatiotemporal features of the input motion image.

3.2.3. Multimodal fusion

To exploit the complementary nature of iMHI and iDMM, we employ the same multimodal fusion strategy as described in Section 3.1.3. Namely, we fuse the two modalities in an intermediate fashion by concatenating the outputs from ResNet49 of both streams in a concatenation layer. We then feed the resulting feature map into a fully-connected layer (FC) with 2048 neurons, as shown in Fig. 5. Eventually, On the top of (FC), the softmax layer (SM) is used to predict the class-membership probability $\mathcal{P}_{MR}\left(\mathcal{C}_r|\mathcal{X}^{MR}\right)$ of the input gesture using Eq. (4), where $\mathcal{X}^{MR}$ is the corresponding output feature vector as predicted by 2D-MRCN.

3.3. Multi-dimensional late fusion for classification

After construction of 3D-CDCN and 2D-MRCN models, we propose to adopt one effective late fusion method at decision level to get the final recognition results. Our choice of using late fusion here allows the possibility that different models can be simultaneously trained according to the characteristics of the input data. Generally, combining individual models at decision level can be done either in supervised or in unsupervised fashions (Duin, 2002). The former requires additional training on the outputs of individual models (ex. SVM or softmax), while the latter does not require any training (ex. majority voting or average method). As mentioned in (Wu et al., 2016b), efficient training of an additional classifier requires additional distinct training sets and memory usage, which is not suitable for our condition. Meanwhile, unsupervised fusion can be more effective since it directly operates on predicted scores from individual models. We, therefore, base our late fusion in this paper on unsupervised fashion. Recall that, for given the gesture observation $\mathcal{X}$, each of the 3D-CDCN and 2D-MRCN networks produces for each class $\mathcal{C}_{r_{1\le r\le c}}$ a class-membership probability $\mathcal{P}_{CD}\left(\mathcal{C}_r|\mathcal{X}^{CD}\right)$ and $\mathcal{P}_{MR}\left(\mathcal{C}_r|\mathcal{X}^{MR}\right)$, respectively. We then perform a simple linear combination to compute the ultimate class-membership probabilities for the given gesture $\mathcal{X}$:

$$\mathcal{P}\left(\mathcal{C}_r|\mathcal{X}\right) = \beta \cdot \mathcal{P}_{CD}\left(\mathcal{C}_r|\mathcal{X}^{CD}\right) + (1-\beta) \cdot \mathcal{P}_{MR}\left(\mathcal{C}_r|\mathcal{X}^{MR}\right) \tag{15}$$

The coefficient β controls the contributions of each model to the classification. The best value of β is determined experimentally. Note that, in our experiments, we also evaluate the capability of the proposed system using several other late fusion schemes such as maximum fusion and product fusion. As for final prediction results, the gesture is assigned the class label $\mathcal{C}^*$ having the maximum class-membership probability:

$$\mathcal{C}^* = \underset{1\le r\le c}{argmax}\left(\mathcal{P}\left(\mathcal{C}_r|\mathcal{X}\right)\right) \tag{16}$$

4. Experiments and analysis

The purpose of this section is to extensively analyze and evaluate the performance of the proposed MultiD-CNN framework under various settings for the task of RGB-D gesture recognition. In what follows, we first introduce the benchmark datasets and experimental settings in Section 4.1. Then, we state the implementation details for the training of MultiD-CNN in Section 4.2. In Section 4.3, we present detailed experiments on the large-scale ChaLearn IsoGD dataset. We also report results on NATOPS, SKIG, and SBU-Kinect Interaction datasets in Section 4.4, 4.5 and 4.6, respectively. Following that, we present some statistical analysis of different components of MultiD-CNN in Section 4.7. In Section 4.8, we analyze the computational time of the proposed system. In Section 4.9, we discuss how MultiD-CNN is effective to complex background. Lastly, we end our experiments by presenting some limitations of MultiD-CNN in Section 4.10.

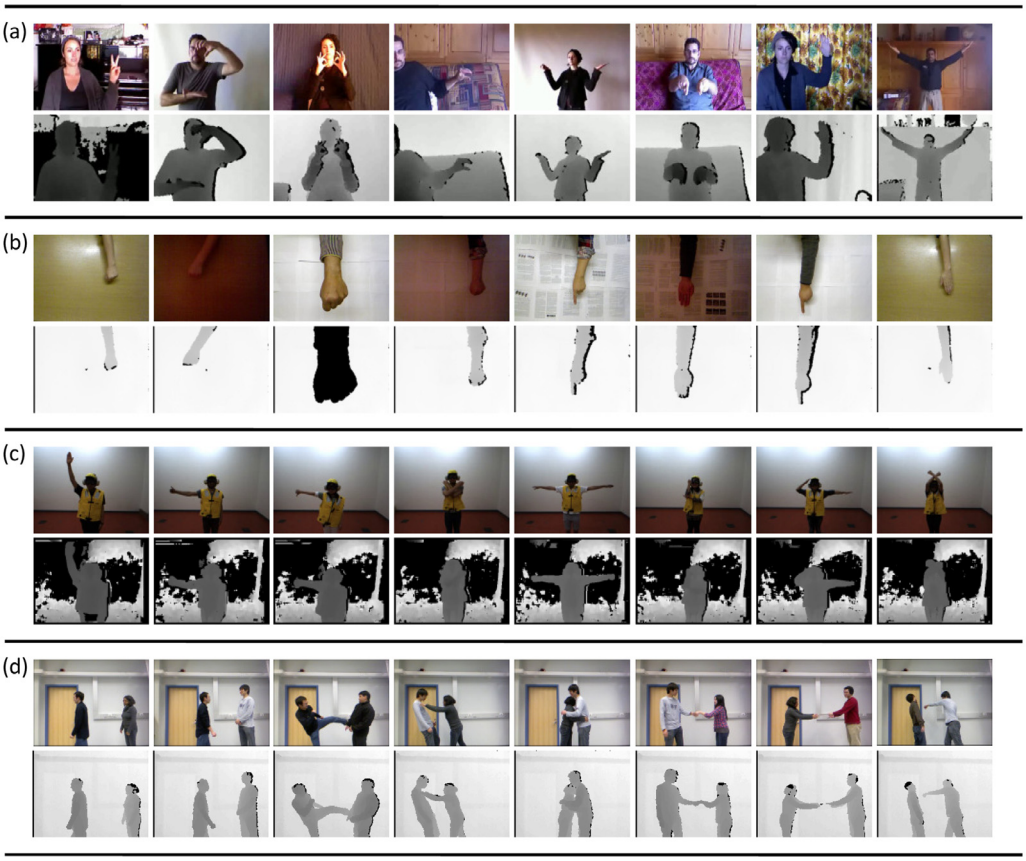


Fig. 9. Sample RGB-D frames of different gestures/actions from the datasets: (a) Chalearn LAP IsoGD, (b) Shefeld KInect Gesture (SKIG), (c) NATOPS gesture, and (d) SBU Kinect interaction.

4.1. Datasets and experimental setup

To validate the advantages of our MultiD-CNN model, we conduct experiments on four challenging public RGB-D gesture datasets: Chalearn LAP IsoGD (Wan et al., 2016b), Shefeld KInect Gesture (SKIG) (Liu & Shao, 2013), NATOPS gesture (Song et al., 2011) and SBU Kinect interaction (Yun et al., 2012).

IsoGD (Wan et al., 2016b) is a large-scale isolated gesture dataset derived from the Chalearn Gesture Dataset (CGD) (Guyon, Athitsos, Jangyodsuk, & Escalante, 2014). This dataset consists of 47,933 RGB-D two-modality gesture sequences performed by 21 different individuals and manually labeled into 249 gesture categories. Each RGB-D video depicts one single gesture instance. The dataset is divided into three mutually exclusive subsets: the training set contains 35,878 gesture sequences performed by 17 individuals, the validation set which contains 5784 gesture sequences performed by 2 individuals and the testing set which contains 6271 gesture sequences performed by 2 individuals. IsoGD is an extremely challenging dataset because it includes several types of gestures (like body/hand gestures, Italian gestures, pantomimes, actions, and so on), and no person performs in the training data will appear in the validation or testing data (user independence). Some samples of the IsoGD dataset are shown in Fig. 9(a). For the evaluation on this dataset, we follow the standard protocol: we use the training set to train our model and we examine on the validation set and the testing set. We report the results in term of accuracy.

SKIG (Liu & Shao, 2013) contains 1080 RGB-D video sequences collected from 6 subjects using Kinect sensor. The dataset includes 10 categories of hand gestures, namely: *circle (clockwise), triangle (anti-clockwise), up-down, right-left, wave, "Z", cross, comehere, turn-around*, and *pat*. All gestures are performed with 3 kinds of hand postures (i.e. fist, index, and flat) under 3 different backgrounds (i.e. wooden board, white plain paper, and paper with characters) and 2 illumination conditions (i.e. strong light and poor light), amounting to 180 RGB-D gesture sequences for each subject. Some examples from this dataset are shown in Fig. 9(b). We follow the same experimental conditions as in (Liu & Shao, 2013) for evaluation on this dataset, where we adopt the three-fold cross-validation for obtaining the average results.

NATOPS (Song et al., 2011) consists of 24 types of upper body gestures representing aircraft handling signals from the Naval Air Training and Operating Procedures Standardization (NATOPS) manual for the US naval aircraft. These gestures are performed by 20 different subjects. Each subject repeats each gesture 20 times, amounting to 400 samples for each gesture class (overall there are 9600 samples in total). The Kinect sensor is used to collect this dataset at 20 fps with 320×240 pixel resolution, providing both RGB and depth data. Videos are recorded in a closed room environment with a constant illuminating condition, and with positions of cameras and subjects fixed throughout the recording. Some typical examples of the NATOPS dataset are illustrated in Fig. 9(c). For the evaluation on this dataset, we follow the same experimental criteria as in (Song et al., 2011), where we select the samples corresponding to the last 10 subjects for training, the first 5 subjects for testing, and the remaining 5 subjects for validation.

SBU (Yun et al., 2012) contains about 282 RGB-D video sequences of humans performing interaction activities that are recorded by Kinect sensor using 21 different subject pairs. All videos in the datasets are divided into 8 types of two-person interactions, including: *approaching, departing, pushing, kicking, punching, exchanging objects, hugging,* and *shaking hands*, with approxi-

mately 40 sequences for each interaction. In most cases, one person is acting and the other person is reacting, and the videos are captured when both the left subject and the right subject initiate the action. Some samples of SBU dataset are shown in Fig. 9(d). Similarly to (Yun et al., 2012), the 5-fold cross validation is adopted for evaluation on this dataset. Namely, the dataset is divided into 5 sets, of which 4 sets are used for training and the remaining one set is used for testing. The evaluation process is repeated 5 times, changing the test set in each repetition.

4.2. Implementation details

In this section, we present the details of learning the MultiD-CNN model. Note that each of the 3D-CDCN and 2D-MRCN networks is trained separately, and they are merged only during the forward propagation phase employed for decision making. All the experiments in this paper are conducted on a PC with *Intel Xeon E5-1603 v4 @ 2.80 GHz* $\times$ *4 CPU, 16GB* RAM and *NVIDIA Geforce GTX 1080 Ti GPU* with *11GB* of memory.

4.2.1. Data augmentation

As a familiar problem in training deep networks, the risk of overfitting is likely to occur mostly due to the small size of datasets and especially in the field of video classification. To overcome this issue, we follow the common practices for CNN training, namely, we augment the training data by adding multiple perturbed copies of each video to cover diversity and variability. The perturbed copies are generated offline by applying some transformations which consist of rotation (-10° to 10°), random cropping, flipping along x and y-axes, scaling with a factor of 2, Gaussian smooth with $\sigma = 0.5$, and contrast adjustment. Besides that, noise and temporal translations (-5 to 5) are also employed. Note that data augmentation can also improve the low accuracy caused by either small or unbalanced number of samples per class.

4.2.2. Learning of 3D-CDCN

At the feature learning stage, we use C3D (Tran et al., 2015) framework to train the 3D-CDCN network, which is a modified version of Caffe (Jia et al., 2014) to support 3D convolution. Each input video is firstly re-sampled to 32-frames clip and spatially resized to 112×112. The sampling is done by removing or repeating frames around the central frame. As no pre-trained model is compatible with the proposed 3D-CDCN architecture, we train the networks from scratch on IsoGD dataset, and then we fine-tune the pre-trained models on the other small-scale datasets (SKIG, NATOPS, and SBU). In our implementation, we separately train the 3D ResNet17 and ConvLSTM to ensure fast convergence and avoid overfitting. Specifically, we first train the 3D ResNet17 part from scratch on each data modality without ConvLSTM. Then, we add the ConvLSTM layers, concatenation layer, and FC layer to the network and fine-tune them again. Note that during fine-tuning the weighs of the 3D ResNet17 are frozen to avoid catastrophic forgetting. This strategy is proved to work sufficiently well.

For optimization on IsoGD, we train the 3D ResNet17 using Stochastic Gradient Descent (SGD) algorithm with a fixed momentum-coefficient of 0.9 and a minimum batch size of 2 examples. The weight-decay is set to 0.0005. Due to the small batch size, the base learning rate is initialized to small value which is 0.001 and decreases by a factor of 0.9 every 5000 iterations. The optimization is stopped at 250,000 iterations. The weights of both RGB and depth streams are randomly initialized from a normal distribution with $\mu = 0$ and $\sigma = 0.05$. To further gain efficiency, we perform cross-modality fine-tuning strategy on IsoGD dataset. Namely, we fine-tune the 3D ResNet17 of depth stream based on the pre-trained model of the RGB modality, and vice versa. The base learning rate for fine-tuning is initialized at 0.001 and dropped to its 0.9 after each 5000 iterations. At most 150,000 iterations are needed for cross-modality fine-tuning. As for training of ConvLSTM, we adopt the implementation of S. Agethen[1] to build our ConvLSTM layer in Caffe (Jia et al., 2014). Adaptive Moment Estimation (Adam) algorithm is used for optimization of our network, with a learning rate fixed to 0.001 and the batch size of 60.

For fine-tuning on the SKIG, NATOPS, and SBU datasets, the learning rate is initialized to 0.0001 and drops at a rate of 0.9 every 5000 iterations. The weight-decay is set to 0.0005 and at most 20,000 iterations are needed for fine-tuning on SKIG and SBU datasets, whereas, 40,000 iterations is executed for NATOPS dataset. We freeze the weights of the pre-trained 3D ResNet17 after connecting to the ConvLSTM to avoid overfitting. The remaining parts of the networks are initialized using the pre-computed weights on IsoGD and fine-tuned using a fixed learning rate of 0.0001.

4.2.3. Learning of 2D-MRCN

The proposed 2D-MRCN network is implemented based on the Caffe framework (Jia et al., 2014). We follow the same training procedure as that of 3D-CDCN, namely, we train the networks on the motion representation generated from IsoGD dataset, and then fine-tune model parameters on SKIG, NATOPS, and SBU datasets. Notice that at the training stage, the motion representation (iMHI and iDMM) is computed for all RGB-D videos of the augmented version in Section 4.2.1. This allows preventing the risk of overfitting. Pre-training is another efficient way to prevent severe overfitting. We, therefore, use the pre-trained model on ImageNet (Deng et al., 2009) to initialize the weights of both 2D ResNets - iMHI and 2D ResNets - iDMM networks. The training on IsoGD is carried-out using SGD algorithm with a fixed momentum-coefficient of 0.9 and a minimum batch of size 16. The weight-decay is fixed to 0.00001. The initial learning rate is set to 0.001 and decreases by a factor of 0.1 every 15,000 iterations. The training process is stopped after 100,000 iterations. The weights of 2D ResNets - iMHI and 2D ResNets - iDMM models are frozen after connecting to the remaining parts of the network, while fine-tuning them again with a learning rate fixed to be 0.001.

After getting the pre-trained models of IsoGD, we then transfer their weights for fine-tuning on SKIG, NATOPS and SBU datasets. The initial learning rate is set to 0.0001 and dropped to its 0.1 every 5000 iterations. The training undergoes 20,000 iterations to update the weights on SKIG and SBU datasets, whereas it needs about 30,000 iterations for fine-tuning on NATOPS dataset.

4.3. Empirical analysis on IsoGD dataset

We begin by evaluating the performance of the proposed MultiD-CNN model on IsoGD dataset. To demonstrate the full potential of our framework, we extensively evaluate the contribution of each component as well as different fusion schemes to the overall system performance. We also present some qualitative results to give an intuitive analysis. We further compare the results of our method with the state-of-the-art approaches, available for this dataset in the literature.

4.3.1. Classification results on different modalities

In this subsection, we study the influence of different data modalities on gesture recognition performance. To evaluate how each data modality performs individually, we add an average pooling layer with a kernel size of $7 \times 7 \times 1$ and a softmax layer right after the last layer of each corresponding stream in 3D-CDCN and 2D-MRCN, and then we train them separately. Fig. 10(a) and (b)

[1] https://github.com/agethen/ConvLSTM-for-Caffe.

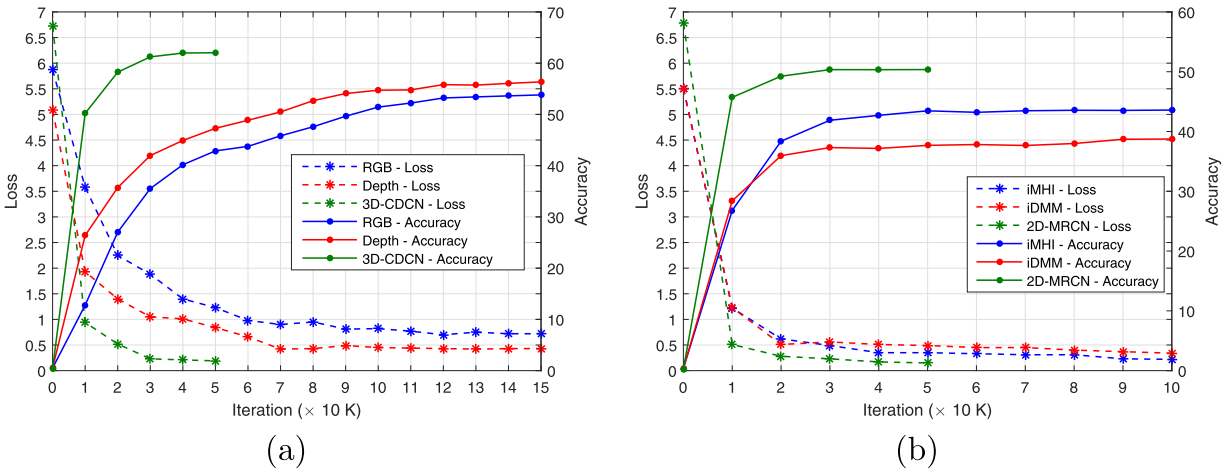


Fig. 10. Performances of (a) 3D-CDCN and (b) 2D-MRCN using different training settings along with training iterations on IsoGD. The left y-axis with dashed curves indicates the variation of loss, while the right y-axis with continuous curves stands for validation accuracy.

Table 1
Recognition results (in terms of classification accuracy in %) using different training configurations on the validation and testing sets of IsoGD dataset.

Training configuration	Validation set	Testing set
3D-CDCN (RGB only)	53.81	56.12
3D-CDCN (Depth only)	56.35	58.43
3D-CDCN (Fusion)	62.04	66.27
2D-MRCN (iMHI only)	43.59	47.82
2D-MRCN (iDMM only)	38.74	45.15
2D-MRCN (Fusion)	50.36	52.07
MultiD-CNN (Linear fusion)	69.84	72.53
MultiD-CNN (Max fusion)	66.37	68.24
MultiD-CNN (Product fusion)	67.42	71.16

show respectively the performance of these data modalities for 3D-CDCN and 2D-MRCN during the training. As can be seen, applying different regularization forms make the network training less likely to be trapped in overfitting even with a large number of iterations, where data augmentation, dropout regularization, and batch normalization are the key-success to obtain an effective network training with good generalization capability. We can also observe that for both modalities of 3D-CDCN (i.e. RGB and depth), the training loss and the accuracy change dramatically up to 120,000 iterations. After that, both of them tend to be stable and almost no variation when the training reaches 150,000 iterations. We then consider that the network has learned enough from the training data. Similar observation can be deduced for input modalities of 2D-MRCN (i.e. iMHI and iDMM), where after 50,000 iterations, the variation of both training loss and validation accuracy is not that much noticeable.

We also report the recognition results of the proposed model using various training settings on the validation and testing sets of IsoGD in Table 1. As can be observed from the performance measures of individual modalities, depth stream achieves better accuracy than RGB stream with a slight improvement of 2.5% on the validation set and 2.3% on the testing set. This can be attributed to the fact that depth data is more capable to deal with complex backgrounds, light changes, clothing, and skin color, hence, making the impact of these factors on the learning process less significant. As for the input modalities of 2D-MRCN, we can see that iMHI stream usually performs better than iDMM stream with an increasing of 4.8% on the validation set and 2.6% on the testing set in term of accuracy. One possible reason is that iMHI image effectively encodes the spatial distribution of movements in the temporal order, where more recently moving regions are brighter than the oldest ones, as clearly visible in Fig. 7, whereas, iDMM accumulates the motion energy across the video sequence without taking into consideration the temporal order of movements.

4.3.2. Effectiveness of multimodal fusion

We now evaluate the effectiveness of multimodal fusion on the system performance. In this experiment, two independent fusions are examined: the first one is to combine the spatiotemporal features from both RGB and depth streams for 3D-CDCN (illustrated in Fig. 2), while the other one is to fuse deep features from iMHI and iDMM streams for 2D-MRCN (illustrated in Fig. 5). From the learning curves in Fig. 10(a) and (b), we observe that the training process of both networks is consistent, where, as the number of iterations increases, the training loss decreases and even tends to be almost close to 0. The accuracy is significantly improved with the number of iterations. This improvement starts to be saturated at around 50,000 iterations for both 3D-CDCN and 2D-MRCN. There is no doubt that combining the merits of multiple modalities for feature description is more discriminative and robust than relying on a single modality features. This can be viewed from the numerical results in Table 1, where multimodal fusion scheme significantly outperforms individual modalities on both of validation (by more than 5.7% for 3D-CDCN and 6.7% for 2D-MRCN) and testing sets (by more than 7.8% for 3D-CDCN and 4.2% for 2D-MRCN). This proves the effectiveness of multimodal fusion strategy in adequately exploiting the comprehensive information hold by different input modalities. The improvement in the accuracy demonstrates also that the RGB and depth information (as well as iMHI and iDMM) supplements each other. Another interesting conclusion that can be drawn from Table 1 is that by treating the temporal dimension as feature channel during the learning process (case of 3D-CDCN), the classifier gives superior classification results (with an improvement of 11.6% and 14.2% on the validation and testing sets, respectively) than accumulating the motion across the video sequence into motion representation before learning process (such in the case of 2D-MRCN).

4.3.3. Effectiveness of multi-dimensional fusion

In this experiment, we analyze the impact of fusing the prediction scores from both 3D-CDCN and 2D-MRCN models on the final decision of MultiD-CNN (the deep architecture illustrated in Fig. 1). We compare among three different late score fusion strategies: linear fusion (described in Section 3.3), maximum fusion and product fusion. The results are also reported in Table 1. We first notice that,

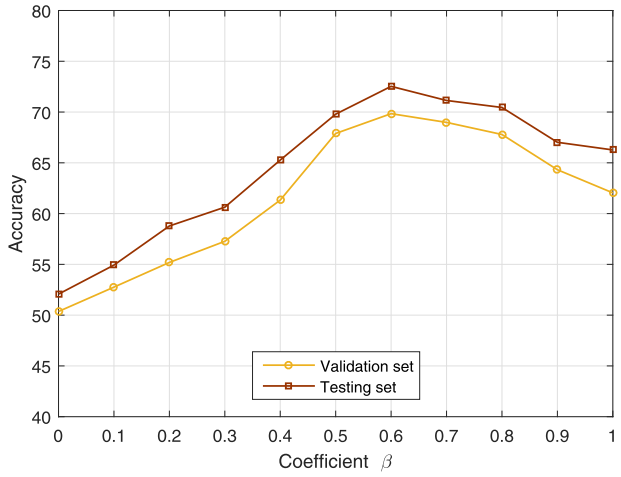


Fig. 11. Impact of varying the coefficient β of late linear fusion on the overall gesture recognition accuracy, the results are reported on the validation and testing sets of IsoGD dataset.

regardless of the strategy used, late score fusion improves greatly the recognition accuracy compared with individual model prediction. This indicates that both 3D-CDCN and 2D-MRCN models contribute positively to the overall system performance, although they treat the motion information differently. Besides, combining different ways of encoding the temporal and spatial information can help improving the generalization capability of the system. We also notice that, among all the compared late fusion strategies, linear fusion yields the best classification accuracy (69.84% and 72.53% on the validation and testing sets, respectively). Fig. 11 shows the effect of varying the coefficient β of linear fusion (Eq. (15)) on the overall system performance. As can be seen, the highest recognition accuracy is obtained when setting the coefficient β to 0.6, demonstrating that 3D-CDCN contributes slightly better than 2D-MRCN to the final decision of MultiD-CNN.

4.3.4. Qualitative results

Some qualitative results of our approach on IsoGD dataset are visualized in Fig. 12. We provide the top-three predicted labels of each recognition result along with their confidences. The examples show users performing gestures in uncontrolled environments (such as complex background, light conditions, noise, and so on). As can be seen, our approach is shown to consistently deliver visually plausible results, where most of the gestures are correctly classified. However, there are some instances of failures where our approach becomes trapped in incorrect prediction (misclassification), such in the case of examples seventh and eighth of Fig. 12. This misclassification might be due to the high interclass similarity between some gesture categories. For example, some parts of movements in gesture of label 221 (*Gesture: SurgeonSignals/StraightForceps*) are very similar to those in gesture of label 37 (*Gesture: ChineseNumbers/wu*). The motion information alone is not enough to discriminate between these gesture categories, which require extra higher-level information that need to be more explored to eliminate these confusions. Nevertheless, there is still sufficient correct confidence score (i.e. correct label is still among the top-three final predicted) that an accurate gesture recognition can still be achieved if the proposed system is further trained on more training data.

4.3.5. Comparison with state-of-the-art methods

In this subsection, we compare the proposed MultiD-CNN approach with other existing state-of-the-art approaches on the validation and testing sets of IsoGD dataset. The comparison is provided in Table 2. As can be seen from this table, handcrafted feature-based methods, such as MFSK (Wan et al., 2016b) and MFSK-Deep ID (Wan et al., 2016b), achieve comparatively less result than deep learning-based methods. This is due to the high capability of convolutional networks in extracting rich representative features from images as well as videos. Similar to our 2D-MRCN model, the methods eDMM-SPM (Zhang et al., 2017a), AMRL (Wang et al., 2016c) and Action Map (Wang et al., 2017a) also learn on handcrafted representations. This is done by firstly transforming the video stream into 2D spatiotemporal feature maps and then using CNN (ex. AlexNet or VGG-16) for gesture classification. Interestingly, our 2D-MRCN model alone significantly outperforms these baselines by a large margin on the validation set (by over 11% improvement in accuracy). This demonstrates that the proposed motion representation (iMHI and iDMM) can better capture the motion information throughout the whole video in a robust manner. On the testing set, AMRL (Wang et al., 2016c) yields 55.57% accuracy, which is slightly better than that of 2D-MRCN by 3% difference. Other approaches, such as Pyramidal C3D (Zhu et al., 2016a) and FLiXT (Li et al., 2016), employ 3D CNN-based architectures to learn the spatiotemporal deep features directly from RGB-D videos. The proposed 3D-CDCN model individually achieves comparatively better results than these approaches on both validation and testing sets with an increasing of 9% on the average.

The idea of combining multiple models for gesture recognition has been extensively used by many other approaches. This combination has demonstrated effective results compared to single model-based methods. For instance, the approach in (Zhu et al., 2017) uses 3D convolution and convLSTM to learn long short-term spatiotemporal features of gestures and achieves 51.02% accuracy on the validation set. (Zhang et al., 2017b) improved the recognition results by further encoding the extracted long short-term spatiotemporal features into 2D convolution. Another interesting combination is proposed in (Duan et al., 2016), where a Two Stream Consensus Voting Network (2SCVN) and 3D Depth-Saliency Network (3DDSN) are integrated into an ensemble learning network. This combination attains 67.19% accuracy on the testing set. We also investigate the same idea on this dataset by combining 2D-MRCN and 3D-CDCN (i.e. MultiD-CNN), while achieving a substantial improvement of 7.8% on the validation set (6.2% on the testing set) compared to 3D-CDCN alone and of 19.4% on the validation set (20.4% on the testing set) compared to 2D-MRCN alone. This obviously demonstrates that 3D-CDCN and 2D-MRCN are both mutually supportive to each other. Overall, the proposed MultiD-CNN combination yields satisfactory performance, where it outperforms all the first 14 baselines in Table 2 on the validation set by a significant margin. On the testing set, our approach performs significantly better than all the first 17 baselines of the same Table 2. Particularly, compared to the ChaLearn 2017 competition winners (ASU) (Miao et al., 2017), we outperform their approach by more than 5.4% on the validation set and 4.8% on the testing set. Finally, the best results that we are aware of on this dataset are those of FOANet (Narayana et al., 2018), which leverages the advantages of multiple residual networks to extract deep features from different focus attention areas within the scene.

4.4. Performance evaluation on SKIG dataset

4.4.1. Results and analysis

For evaluation on SKIG dataset, we adopt the three-fold cross-validation strategy, where we divide the dataset into three subsets with samples of two subjects for each, and at each time we train the model on two subsets while the remaining one subset is used for testing. The performance confusion matrices of 3D-CDCN alone, 2D-MRCN alone, as well as the proposed fusion model (MultiD-CNN) on this dataset are displayed in Fig. 13. As can be

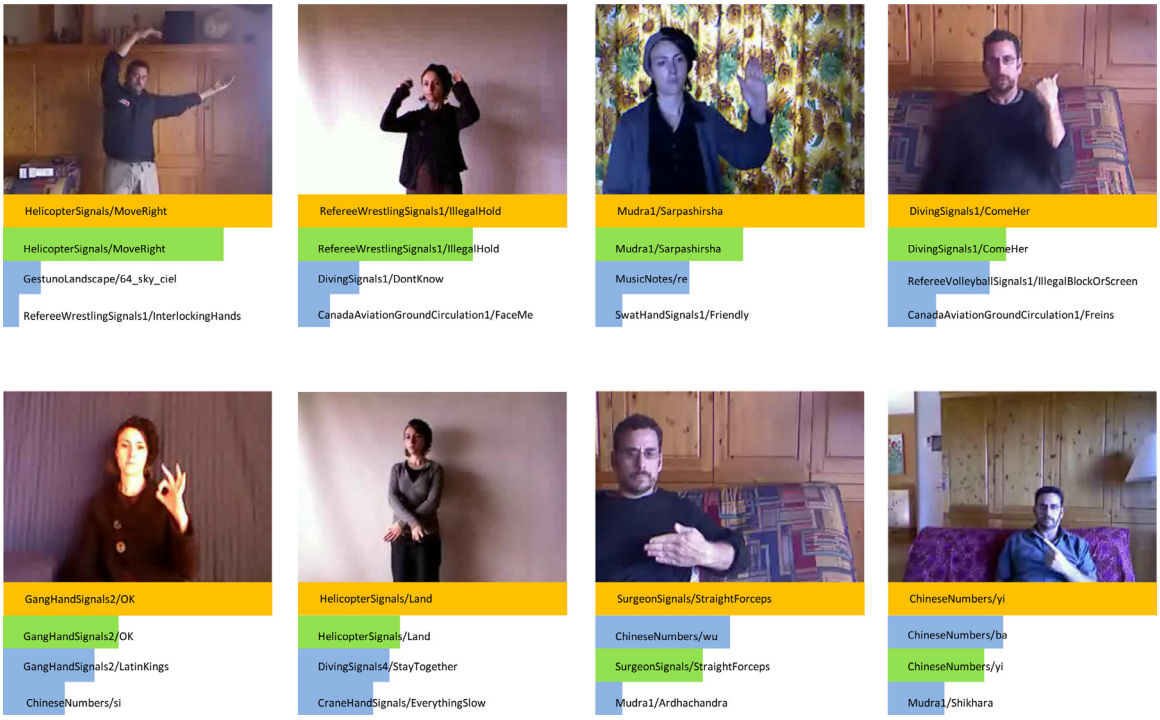


Fig. 12. Qualitative recognition results for some sample gestures from validation set of IsoGD dataset. We show the top-three final predicted labels using our MultiD-CNN approach. Ground-truth is shown in orange bars, correct predictions in green and wrong in blue. Bar length indicates prediction confidence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

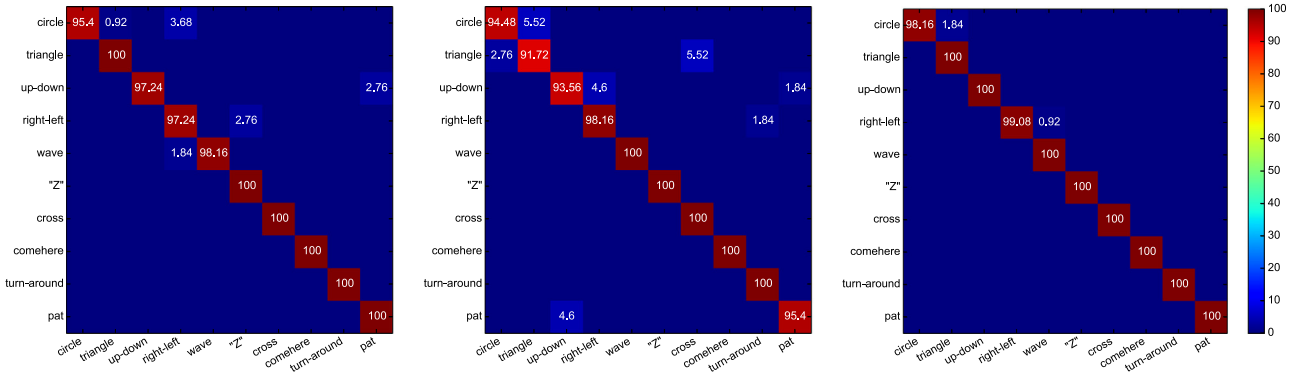
Table 2
Performance comparison with state-of-the-art approaches on the validation and testing sets of IsoGD dataset (in terms of classification accuracy in %).

Validation set		Testing set	
Team / Method	Accuracy	Team / Method	Accuracy
MFSK-Deep ID (Wan et al., 2016b)	18.23	NTUST	20.33
MFSK (Wan et al., 2016b)	18.65	MFSK-Deep ID (Wan et al., 2016b)	23.67
Action Map (Wang et al., 2017a)	36.27	MFSK (Wan et al., 2016b)	24.19
eDMM-SPM (Zhang et al., 2017a)	36.69	TARDIS	40.15
AMRL (Wang et al., 2016c)	39.23	eDMM-SPM (Zhang et al., 2017a)	43.91
Pyramidal C3D (Zhu et al., 2016a)	45.02	XJTUfx	43.92
FLiXT (Li et al., 2016)	49.20	ICT-NHCI (Chai et al., 2016)	46.80
C3D-ConvLSTM (Zhu et al., 2017)	51.02	Pyramidal C3D (Zhu et al., 2016a)	50.93
XDETVP	58.00	AMRL (Wang et al., 2016c)	55.57
3DCNN-CLSTM (Zhang et al., 2017b)	58.65	FLiXT (Li et al., 2016)	56.90
SYSU_ ISEE	59.70	XDETVP	60.47
DycImage + 3DCLSTM (Wang et al., 2017b)	60.81	3DCNN-CLSTM (Zhang et al., 2017b)	62.14
Lostoy	62.02	DycImage + 3DCLSTM (Wang et al., 2017b)	65.59
ASU (Miao et al., 2017)	64.40	Lostoy	65.97
FOANet (Narayana et al., 2018)	80.96	SYSU-ISEE	67.02
–	–	2SCVN-3DDSN (Duan et al., 2016)	67.19
–	–	ASU (Miao et al., 2017)	67.71
–	–	FOANet (Narayana et al., 2018)	82.07
3D-CDCN	**62.04**	**3D-CDCN**	**66.27**
2D-MRCN	**50.36**	**2D-MRCN**	**52.07**
MultiD-CNN	**69.84**	**MultiD-CNN**	**72.53**

seen from this figure, the confusion matrices are strongly diagonal with negligible confusion errors, this indicates that most of the gestures are accurately recognized (with above 95% classification accuracy per class) using either 3D-CDCN or 2D-MRCN models, while achieving relatively reasonable average accuracies of 98.80% and 97.33%, respectively. The results demonstrate also the complementarity of the behaviors of 3D-CDCN and 2D-MRCN, where after multi-dimensionality fusion, the average classification accuracy is improved to 99.72%, which is obviously better than separate models and even close to the optimum. In addition, we can also observe that our framework is robust against pose, illumination and background variations.

4.4.2. Comparison with state-of-the-art approaches

We further compare our framework with several existing methods working on the SKIG dataset and the competitive results are reported in Table 3. Note that the first half of the compared approaches in this table rely on handcrafted feature representations for gesture classification. We significantly outperform these



(a) 3D-CDCN (b) 2D-MRCN (c) MultiD-CNN

Fig. 13. Confusion matrices of 3D-CDCN, 2D-MRCN, and MultiD-CNN (with $\beta = 0.5$) models for three-fold cross-validation on SKIG dataset. Average accuracy rates for each gesture class are listed along the diagonal.

Table 3
Performance comparison with state-of-the-art approaches on SKIG dataset (in terms of classification accuracy in %).

Method	Accuracy
RGGP+RGB-D (Liu & Shao, 2013)	88.70
hierarchical structure (Choi & Park, 2014)	91.90
4DCOV (Cirujeda & Binefa, 2014)	93.80
Depth Context (Liu & Liu, 2016)	95.37
EDSM (Tung & Ngoc, 2014)	96.50
MRNN (Nishida & Nakayama, 2015)	97.80
DLEH2(DLE+HOG2) (Zheng et al., 2017)	98.43
3DCNN+RNN+CTC (Molchanov et al., 2016)	98.60
C3D-ConvLSTM (Zhu et al., 2017)	98.89
3DCNN-CLSTM (Zhang et al., 2017b)	99.53
3D-CDCN	**98.80**
2D-MRCN	**97.33**
MultiD-CNN	**99.72**

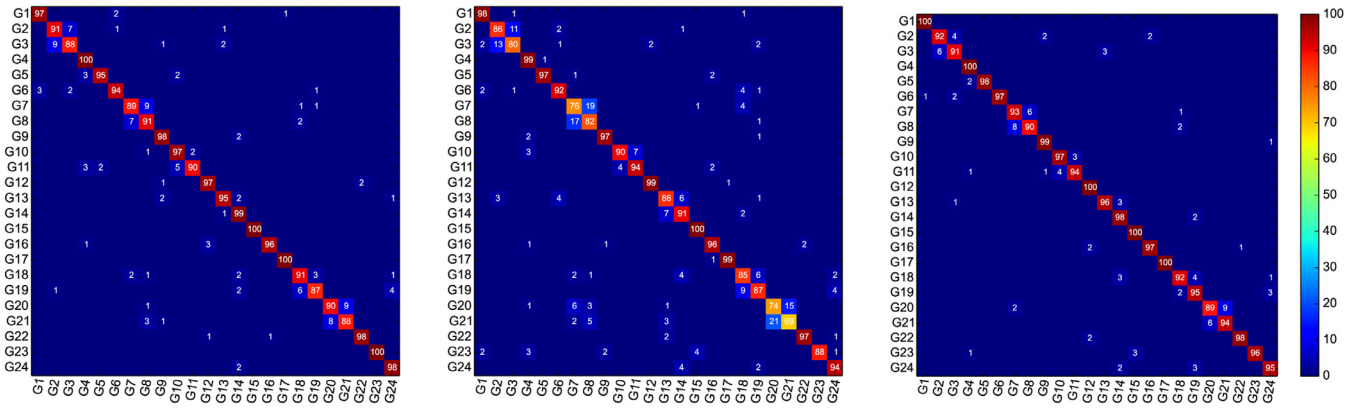
baselines by a large margin when using either 3D-CDCN or 2D-MRCN individually.

As can be seen that the only handcrafted feature based method that outperforms our 2D-MRCN model is DLEH2 (Zheng et al., 2017), which combines three types of feature descriptors (LBP, EOH and HOG) on DMM and feeds them into SVM classifier for gesture recognition. Nevertheless, we believe that with more training data, our 2D-MRCN model would be better than DLEH2 (Zheng et al., 2017) since the performance of handcrafted based methods is unlikely to improve much with the availability of training data. Compared to deep learning based methods, the 3D-CDCN model alone outperforms the MRNN (Nishida & Nakayama, 2015) baseline with a slight improvement of 1% on the average, while achieving competitive performance with 3DCNN+RNN+CTC (Molchanov et al., 2016) and C3D-ConvLSTM (Zhu et al., 2017) methods. The way of learning the spatial and temporal information has a great impact on gesture recognition performance. For instance, the MRNN (Nishida & Nakayama, 2015) and 3DCNN+RNN+CTC (Molchanov et al., 2016) approaches learn the spatial and temporal features consecutively using CNN and then feeds these features into RNN for gesture classification. However, both methods do not support the encoding of spatial correlation information in the recurrent process, which assumes a fundamental importance in gesture recognition considering that it depicts the extent of understanding of the entire sequence. In contrast, our approach with that of ConvLSTM (Duan et al., 2016) and 3DCNN-CLSTM (Zhang et al., 2017b), simultaneously embed the temporal and spatial correlation cues through the whole feature learning process. This demonstrates the superiority of these approaches compared to those of (Molchanov et al., 2016; Nishida & Nakayama, 2015). As can be seen from Table 3, by combining the advantages from both 3D-CDCN and 2D-MRCN models, it is possible to expressively overcome many advanced and relatively complex techniques while also achieving the state-of-the-art accuracy on this dataset.

4.5. Performance evaluation on NATOPS dataset

4.5.1. Results and analysis

On this dataset, we follow the standard evaluation protocols. Namely, the proposed approach is trained on the samples of the last 10 subjects and tested on the samples of the first 5 subjects, while the remaining samples of other subjects are used for validation. The confusion matrices of separately performing 3D-CDCN, 2D-MRCN and the proposed fusion model (MultiD-CNN) on the testing subset are shown in Fig. 14. We first notice that the individual models achieve pretty good classification results on this dataset (with average accuracies of 94.54% and 89.83% for 3D-CDCN and 2D-MRCN, respectively). In contrast to the SKIG dataset, the 3D-CDCN model achieves better results than 2D-MRCN model by a significant margin with over 4.7% accuracy. This again strengthens the explicit statement that learning spatiotemporal features simultaneously is more suitable for gesture recognition than encoding the temporal information into a spatial domain before the learning process. We can also observe from Fig. 14 that there is a high confusion between gestures (G2, G3), (G7, G8) as well as (G20, G21). This is mainly due to the similarity in performing these gestures with movements along with the same direction, and can only be distinguished by the hand pose. As the 3D-CDCN and 2D-MRCN models are insensitive to hand pose variation, they will have more difficulties in recognizing gestures which primarily differ in this aspect. By fusing the 3D-CDCN and 2D-MRCN models, it is possible to overcome this limitation. This can be viewed in Fig. 14(c), in which the confusion errors between these gestures are significantly reduced. Overall, when using MultiD-CNN all gestures obtain better performance than individual models, especially that 18 out of 24 gesture classes are classified with above 94% classification accuracy. It seems that 2D-MRCN contributes with little improvement on the MultiD-CNN classification accuracy. One possible explanation is that the 3D-CDCN has already achieved very high accuracy on this dataset and the remaining interval for improvement is relatively small.



(a) 3D-CDCN (b) 2D-MRCN (c) MultiD-CNN

Fig. 14. Confusion matrices of 3D-CDCN, 2D-MRCN and MultiD-CNN (with $\beta = 0.7$) models for the first 5 subjects on NATOPS dataset. Average accuracy rates for each gesture class are listed along the diagonal.

Table 4
Performance comparison with state-of-the-art approaches on NATOPS dataset (in terms of classification accuracy in %).

Method	Accuracy
CRF (Song et al., 2012)	53.30
EODPCA (Joshi et al., 2015)	67.74
EOD (Joshi et al., 2015)	76.63
HMM (Song et al., 2012)	77.67
HCRF (Song et al., 2012)	78.00
eDMM-SPM (Zhang et al., 2017a)	83.47
SK+HS (Joshi et al., 2015)	84.77
Coupled HCRF (Song et al., 2012)	86.00
Temporal Template (Ijjina & Chalavadi, 2017)	86.58
Linked HCRF (Song et al., 2012)	87.00
SK+HS+EODPCA (Joshi et al., 2015)	87.35
3D-CDCN	**94.54**
2D-MRCN	**89.83**
MultiD-CNN	**95.87**

4.5.2. Comparison with state-of-the-art approaches

A comparative study between our work and other approaches working on NATOPS dataset with the same experimental setup is reported in Table 4. As can be seen from the results of this table, the performances of individual models are really impressive as they all score high compared to competing methods. Furthermore, the proposed MultiD-CNN combination obtains the state-of-the-art accuracy of 95.87% which outperforms all previous methods by a large margin. The baselines in (Joshi, Monnier, Betke, & Sclaroff, 2015; Song, Morency, & Davis, 2012) make use of, in addition to the RGB-D modalities, both skeletal and hand pose information to better distinguish between confused gestures. Our 3D-CDCN model alone achieves better results than these baselines (with a significant improvement of 7.2% in accuracy over the best performing one) using RGB-D information only. This is probably due to the fact that 3D-CDCN model gains extremely high recognition accuracy on the non-confused gesture classes, while the effect of low accuracies on other classes that differ only on hand pose is be less pronounced. Compared to eDMM-SPM (Zhang et al., 2017a) and temporal template (Ijjina & Chalavadi, 2017) approaches, our 2D-MRCN model achieves comparatively better accuracy with an improvement of 6.3% and 3.2%, respectively. We believe this comparison would be equitable since both approaches also extract deep features from motion representation using 2D convolution. It is thus fair to say that our method of encoding the motion information in iMHI and iDMM is effective in handling various challenges in RGB-D gesture recognition.

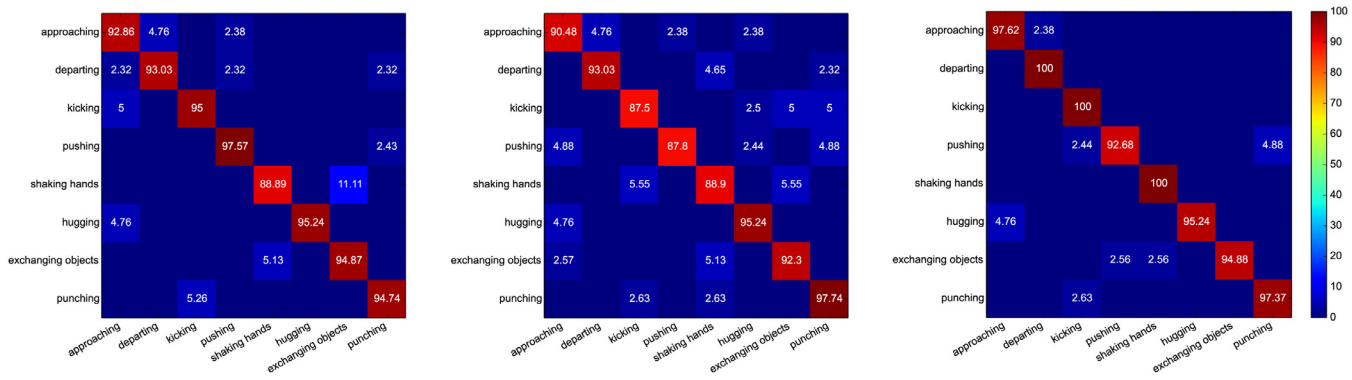
4.6. Performance evaluation on SBU dataset

4.6.1. Results and analysis

To further consolidate our experiments, we also investigate the behavior of the proposed approach for classification of two-person interaction activities on the well-known SBU dataset. Note that the actions in this dataset are basically different from those tested in the previous experiments, where instead of only one person performing the action, two persons are involved in the interaction. In most cases, one person is acting and the other one is reacting. As we mentioned before, five-fold cross-validation is used for evaluation on this dataset. Fig. 15 shows the confusion matrices when using 3D-CDCN only, 2D-MRCN only and MultiD-CNN as their multi-dimensional fusion on SBU dataset. 2D-MRCN separately achieves an average classification accuracy of 91.13%, while 3D-CDCN alone permits to obtain a slightly better performance with an average accuracy of 94.32%. As can be seen from Fig. 15(a) and (b), both individual models often confuse between the shaking hands and exchanging object interactions since both of them involve extending the arms by both subjects. The individual models also have difficulties while distinguishing between *kicking*, *pushing* and *punching* because of the acting action (initiation of the action) and the reacting action (response to the action) are almost the same in their interactive activity models. Likewise, the noticeable confusion between *approaching* and *departing* interactions is due to the same reason. One can see from Fig. 15(c) that the confusion matrix of MultiD-CNN is roughly diagonal with few confusion errors, and as expected, the recognition accuracy of most interaction classes is significantly improved when using the fusion of 3D-CDCN and 2D-MRCN models. Especially, three out of eight classes are recognized with 100% classification accuracy, and the other five classes are also classified with above 92% accuracy. This result indicates that the proposed framework is very efficient for person-to-person interaction modeling, while suggesting broad applicability to other video classification problems.

4.6.2. Comparison with state-of-the-art approaches

In this subsection, we compare the performance of our proposed method with several other competing methods working on the same SBU dataset. The comparative results are reported in Table 5. We first notice that most of the compared approaches rely on the skeletal information provided along with RGB-D data to recognize human interactions. For instance, the baselines in (Ji, Ye, & Cheng, 2014; Yun et al., 2012) incorporate several body-pose features (ex. joint distance and joint motion) from skeletal data and



(a) 3D-CDCN (b) 2D-MRCN (c) MultiD-CNN

Fig. 15. Confusion matrices of 3D-CDCN, 2D-MRCN, and MultiD-CNN (with $\beta = 0.5$) models for five-fold cross-validation on SBU dataset. Average accuracy rates for each two-person interaction class are listed along the diagonal.

Table 5
Performance comparison with state-of-the-art approaches on SBU dataset (in terms of classification accuracy in %).

Method	Accuracy
Raw skeleton (Yun et al., 2012)	49.70
Raw skeleton (Ji et al., 2014)	79.40
Joint features (Yun et al., 2012)	80.30
Hierarchical RNN (Du et al., 2015)	80.35
Cluster analysis of pose (Edwards & Xie, 2015)	83.90
Deep LSTM (Zhu et al., 2016b)	86.03
Joint features (Ji et al., 2014)	86.90
Generative topic model (Huynh-The et al., 2015)	90.30
Co-occurrence LSTM (Zhu et al., 2016b)	90.41
Temporal Template (Ijjina & Chalavadi, 2017)	90.98
STA-LSTM (Song et al., 2017)	91.51
ST-LSTM + Trust Gate (Liu et al., 2016b)	93.30
Radius-margin bound (Lin et al., 2016)	93.40
3D-CDCN	**94.32**
2D-MRCN	**91.13**
MultiD-CNN	**97.51**

input them into SVM classifier for interaction spotting. (Huynh-The et al., 2015) improves the recognition performance by encoding the extracted joint features in a codebook representation, in which they use the binary-tree of SVM to classify the action. Our MultiD-CNN significantly outperforms these three baselines by a large margin with over 17.2%, 10.6%, and 7.2% accuracy, respectively. Even though using skeletal features with LSTM that is effective for recognizing time series data (Du et al., 2015; Liu et al., 2016b; Song et al., 2017; Zhu et al., 2016b), the performance of the proposed approach still remains crucially superior. The impressive obtained results demonstrate that the learned deep features from RGB-D raw data maintain considerably higher discriminative power compared to skeletal information. While comparing against the best performing method on this dataset which is Radius-margin bound (Lin et al., 2016), our compositional model achieves the state-of-the-art accuracy, with a substantial improvement of over 4.1%.

4.7. Statistical model analysis

Through our analysis, it looks like some models have good performance than others. But does this represent a statistical difference between these models in terms of mean accuracy? To assess the statistical significance of our results, we use the Wilcoxon test which is a non-parametric test used to compare two sample groups. In this assessment, the level of significant used is $\alpha = 0.05$. Note that on each of the models, if the test reports a large p-value than the significant level, then this means that there is no significantly difference. Otherwise, one model is performing significantly better than the other. In what follows, we provide an analysis of the results that encompasses three types of comparison.

In the first comparison, we analyze the statistical significance of the 3D-CDCN network where we compare the depth and RGB streams on each instance. The null hypothesis H_0^{CD} is that no differences between recognition accuracy of the two depth and RGB streams. The results obtained through the four studied datasets of IsoGD, SKIG, NATOPS, and SBU are reported in Table 6. Throughout the analysis of 3D-CDCN, the obtained p-value from the Wilcoxon test is smaller than the significance level (0.05), which suggests to reject our null hypothesis H_0^{CD}. Thus, saying that the differences we found are unlikely to have occurred by chance.

In the second comparison, we compare the proposed iMHI and iDMM streams so that to analyze the statistical significance of the 2D-MRCN network. Our null hypothesis H_0^{MR} is that the two streams of iMHI and iDMM produced no different recognition accuracy. As can be seen from Table 6, the returned p-values over the four studied datasets are still smaller than α level of significant, suggesting that H_0^{MR} is unlikely to be true. This demonstrates that the differences observed are statistically significant.

In order to prove beyond a shadow of a doubt that the two proposals of 3D-CDCN and 2D-MRCN of the overall MultiD-CNN model is statistically significant and it does not work due to chance, we carry out a hypothesis contrasting. In this case study, our null hypothesis H_0^{MultiD} is that both 3D-CDCN and 2D-MRCN networks contribute equally to the recognition accuracy of the gesture. The general assumption is that different feature learning models contribute differently to the recognition accuracy. This was refuted by the obtained results over the four studied datasets, where the returned p-values by Wilcoxon test are strongly smaller than α level of significance, allowing to reject H_0^{MultiD}, and to claim that the differences observed are statistically significant. These findings suggest that further investigation should focus on analyzing ensemble-learning network effects.

4.8. Computation time analysis

The overall computation time using the above-mentioned hardware specifications is on two parts: training time and testing time. In particular, the training time per iteration of the 3D-CDCN model takes about 0.9 s. This allows us to complete the 150,000 training iterations in approximately 38 h. The training time of each iteration of the 2D-MRCN model is much lower, where it takes less

Table 6
The corresponding *p*-value to the *z*-value returned by Wilcoxon test over depth and RGB streams for 3D-CDCN network, iMHI and iDMM streams for 2D-MRCN network, and 3D-CDCN and 2D-MRCN networks for MultiD-CNN model.

Datasets	3D-CDCN		2D-MRCN		MultiD-CNN	
	z-value	*p*-value	*z*-value	*p*-value	*z*-value	*p*-value
IsoGD	−3.0986	0.00194	−2.8373	0.00452	−3.3786	0.00072
SKIG	−3.6586	0.00026	−3.6026	0.00032	−3.7146	0.00020
NATOPS	−3.8079	0.00014	−3.3039	0.00096	−3.7333	0.00020
SBU	−3.7706	0.00016	−3.0239	0.00252	−3.7519	0.00018

than 0.26 *s*. This results in a total time of 7 *h* to train the 2D-MRCN network for 100,000 iterations.

As for the testing time, the proposed MultiD-CNN can successfully perform real-time gesture labeling on GPU. Namely, to propagate a gesture forward through the 3D-CDCN network, it takes about 100 ms, while 5 ms is required for forward propagation of the same gesture through the 2D-MRCN network. By considering the number of frames in each gesture as 32, the 3D-CDCN model can run for an average speed of about 320 fps (i.e. 10 gestures per second). And by considering that each gesture is represented by one single frame, we get 200 fps (i.e. 200 gestures per second) for 2D-MRCN model. We finally obtain an average speed of 10 gestures per second for their combination.

Note that for generation of motion representation in both testing and training phases, the computation time is a little bit longer (it takes about 0.3 s for extraction of both iMHI and iDMM for a single RGB-D gesture) since only CPU is employed to perform this task.

It is evident that the accomplishment of 3D convolution requires more multiply-and-addition operations than 2D convolution. This explains the high computational complexity of 3D-CDCN in both training and testing phases compared to 2D-MRCN, even though this later contains more convolutional layers than the former. Nevertheless, we are confident we can achieve further acceleration by parallelizing our framework. This can be easily done since the convolutional networks at each sub-model of MultiD-CNN are independent. In the current work, only the convolutional networks are implemented on GPU. Therefore, more efficient and fast processing can be boosted if the overall system is fully implemented on GPU. For example, implementing the generation of motion representation stage as a new processing layer in 2D-MRCN should improve the performance of the pipeline significantly. Furthermore, both 3D-CDCN and 2D-MRCN in our approach are trained step-by-step, which increase the training and testing complexity. We expect further improvements in computational efficiency might be obtained if both of them are trained in an end-to-end manner. This will enable the proposed methods to be easily embedded into real-time expert systems. However, such task requires one or multiple GPUs with sufficient memory capacities to handle this huge amount of trainable parameters.

4.9. Effectiveness to complex background

One of the major difficulties that are usually encountered in any gesture recognition system is the uncontrolled environment. In fact, an ideal gesture recognition system should operate regardless of the background complexity and on the variety of lighting conditions. In order to evaluate how effective our model is when working with complex scene backgrounds, we conduct experiments on 300 gestures selected randomly from the validation and testing sets of IsoGD dataset. We chose this dataset among the studied ones since it contains varieties of cluttered backgrounds. Notice that in this experiment, we evaluate only the influence of static background since dynamic scenes are not considered in the current work. The assessment deals with a pre-processing stage of the data in which the background is firstly subtracted before exploiting it. Specifically, for each gesture sample, we first align the corresponding depth video to RGB video. Then, we apply Faster R-CNN, pre-trained on PASCAL VOC for object detection, to extract the gesture regions from RGB video frame by frame (i.e. the upper body part of the performer in each frame). The detected bounding boxes are subsequently projected on the frames of depth video to get the depth gesture regions. Note that the missed regions are manually corrected when the Faster R-CNN detection is incorrect. The obtained gesture regions through both modalities are then inputted to MultiD-CNN with the same parameter setting to fairly evaluate the effect of background on the system performance. On these 300 samples, we achieve an average recognition accuracy of 87.66% without background subtraction versus 88.33% after background subtraction. From these results, we notice that there is no significant improvement when discarding or keeping the background, which ensures the robustness of MultiD-CNN to the cluttered background. This is not surprising since the motion representation employed in our approach makes the learning process focused only on the moving part in the videos, while other irrelevant visual details are filtered out through the de-background scheme. Fig. 16 depicts a sample gesture from the validation set of IsoGD dataset where the top-three predicted labels of each recognition result along with their confidences are reported. As one can observe, even though the background is much cluttered and consisted of many objects, the qualitative results are almost comparable with and without the background subtraction, thus again proving that our model is effective to the complex background.

4.10. Limitations

We believe that our MultiD-CNN has dramatically pushed the boundaries of human gesture recognition by exploiting the full complementary advantages embedded in RGB and depth cues, particularly in terms of efficiency and generalization. Despite the satisfactory results, it still suffers from certain limitations. Besides the training and testing complexity, another interesting issue, as indicated by our experimental results, is the drop in classification accuracy in scenarios where there are different gestures with movements along with the same direction and only differ in hand pose. As the present work does not take the pose variation into consideration, MultiD-CNN under such scenarios may not be sufficiently convenient for practical applications involving this type of gestures. Furthermore, investigating the complete spatial and temporal information from RGB-D sensor is efficient and generic, as MultiD-CNN takes into consideration all moving parts of the body that represent the gesture. However, this aspect will also make the spatiotemporal feature learning confuse between motion of gesture and other moving objects in case of dynamic scene background. Therefore, MultiD-CNN in its current form is suitable only for recognizing gestures in indoor scenes, where the background is almost static or constant.

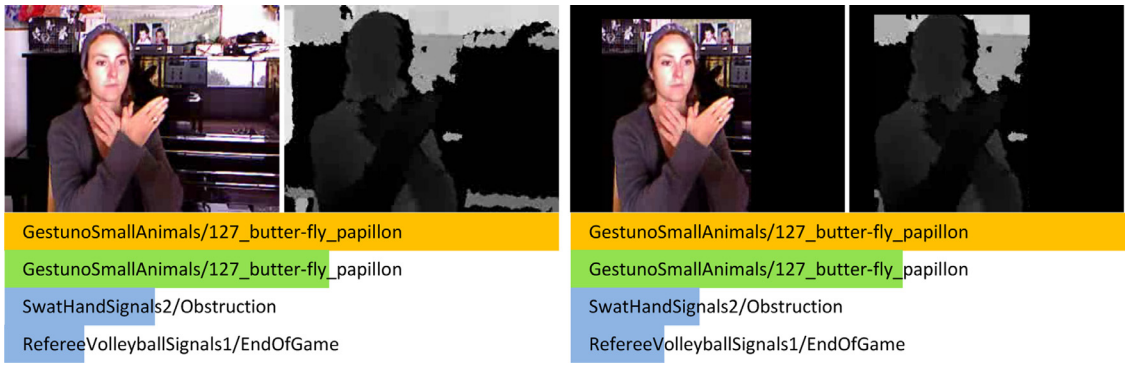


Fig. 16. Recognition results for a sample gesture from IsoGD dataset without background subtraction (*left side*) and with background subtraction (*right side*). We show the top-three final predicted labels using MultiD-CNN. Ground-truth is shown in orange bars, correct predictions in green and wrong in blue. Bar length indicates prediction confidence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Conclusion

In this paper, we presented a multi-dimensional feature learning approach for the task of human gesture recognition in RGB-D video, termed as MultiD-CNN. Existing methods in the field are either handcrafted or deep learning-based. Although deep learning-based methods achieve state-of-the-art performance, we managed to improve its efficiency by incorporating multiple concepts of encoding the spatial and temporal information whilst keeping its generalization power. To this end, we proposed two distinct feature learning methods capable of aggregating frame-level CNN outputs into gesture-level classification. The first method (3D-CDCN) incorporates 3D ResNets and ConvLSTM for processing of RGB and depth sequences, while the second method (2D-MRCN) uses 2D ResNets for treating their motion representation. Both methods are motivated by the fact that ensemble learning usually produces more accurate results than relying on a single model. Our motion representation is built upon iMHI and iDMM. In this context, we also proposed an effective method to filter the gesture-irrelevant factors and to capture discriminative motion patterns from RGB-D cues. Because the input data is multimodal, we investigated their intrinsic relationship by adopting an intermediate fusion which processed at the feature level. Additionally, to blend the recognition results of individual components, a simple linear fusion strategy on the scores is exploited according to the contribution of each component to the final classification.

We evaluated this compound model on four widely used datasets (i.e. IsoGD, SKIG, NATOPS, and SBU) and have shown the following. First, 2D/3D ResNets are good options to learn high-level features, and ConvLSTM is a better choice for modeling long short-term spatiotemporal dependencies. Second, the multimodal fusion of the different inputs results in a clear improvement over unimodal models due to the complementary nature of the different input modalities. Third, the multi-dimensional fusion of 3D-CDCN and 2D-MRCN models improves the recognition results considerably compared to individual models, thereby demonstrating the ability to recover from uncorrelated errors of each model. Fourth, our experimental validation on the four datasets has indicated that MultiD-CNN performs at the same level as other state-of-the-art methods. Fifth, the good potential of MultiD-CNN proven through its application in the three real-world problems of upper body gesture, hand gesture, and two-person interaction recognition suggest its investigation in other domains that involve video and motion classification. The codes would be released to facilitate future research.

In terms of theoretical contributions, the proposed system is built based on three principles as follows: (a) Effective spatiotemporal feature learning - simultaneous embedding the temporal and spatial correlation cues assumes a fundamental importance in gesture recognition, considering that it depicts the extent of understanding of the entire sequence. We investigated this principle in our system by building the 3D-CDCN model. In contrast to the works of (Molchanov et al., 2016; Nishida & Nakayama, 2015; Wang et al., 2016a; Wu et al., 2016a) which learn the spatial and temporal features consecutively, our 3D-CDCN allows capturing the fine-grained motion details encoded in multiple adjacent frames, thereby preserving the three-dimensional structural information of RGB-D videos throughout the whole learning process. Empirical results showed clear advantages of implementing this principle in our system over these relatively existing expert systems. (b) Motion representation theory - for recognizing gestures in complex environments, the system has to have enough knowledge to distinguish between relevant and irrelevant visual motion patterns. To achieve this, we adopted the concept of motion representation to eliminate irrelevant variations from data and make the learning only focus on the motion of performers. Compared to (Ijjina & Chalavadi, 2017; Wang et al., 2017a; 2016c; 2016b; Zhang et al., 2017a), our motion representation through iMHI and iDMM can better encodes the motion characteristics of the gesture in an explicit manner. The introduction of motion significance score strategy allows improving the quality of iMHI and iDMM so that to retain only the motion of moving body parts instead of being content with the standard MHI (Ijjina & Chalavadi, 2017) and DMM (Zhang et al., 2017a) that are sensitive to several environmental factors. The pseudo-coloring power rainbow transform is used to further enhance the visual aspect of both representations. This facilitates the feature learning process as motion patterns of the gesture can be easily distinguished. By leveraging these advantages, our 2D-MRCN alone has boosted the performance recognition and outperformed all these intelligent systems over the four studied datasets. (c) Multi-dimensional fusion - in order to make use of the full advantages embedded in different kinds of features learned by 3D-CDCN and 2D-MRCN networks, so that to improve the performance of recognition results, we employed the multi-dimensional fusion strategy through our MultiD-CNN model. 3D-CDCN is accurate for recognition but makes more mistakes when faced with gesture-irrelevant factors, whereas 2D-MRCN allows for robust recognition but is not as good in term of accuracy. Experimental results showed that multi-dimensional fusion between these two distinct models allows defeating the shortcoming of individual models, while achieving better accuracies compared to existing single and hybrid model-based intelligent systems (Duan et al., 2016; Miao et al., 2017; Wang et al., 2017b; Zhang et al., 2017b; Zhu et al., 2017).

The field of gesture recognition with the RGB-D camera is still in its infancy; however, it is obvious that it will become preva-

lent in the near future. There are several directions that are natural progressions of this work and which could underpin our future investigations. This would concern two aspects of the proposed approach: conceptual foundations and practical applications. The first aspect aims to overcome the major limitations of MultiD-CNN regarding its performance. In this backdrop, human gestures are highly related to different modalities. For instance, if a gesture is unrecognizable to the system in one modality it can be disambiguated using the other modality. Therefore, we may consider incorporating more complementary semantic cues such as skeletal information and other motion representations (e.g. optical flow as done in Duan et al., 2016) into a unified framework. Potentially, this additional knowledge will allow us to distinguish the subtle differences between some confusing gestures, and thus gaining further performance. Furthermore, although the proposed fusion scheme shows good potential over individual models, it is still not guaranteed to be the best suitable combination strategy for multi-dimensional feature learning. Hence, it would be also interesting to verify whether the performance can be improved further by exploiting more sophisticated fusion schemes such as hierarchical or weighted fusions, or even more advanced variants (Narayana et al., 2018). As previously mentioned, dynamic backgrounds bring negative influences to effective gesture recognition. To handle this issue, we also plan to further integrate a coarse segmentation step to differentiate background movements from gesture segments. Possibly this can be achieved by using Generative Adversarial Networks (GAN), as it has shown promising performance in modeling the scene dynamic for both video recognition and generation tasks (Vondrick, Pirsiavash, & Torralba, 2016). Concerning the second aspect, the focus is on enhancing the context of model application. Indeed, in this work, it is assumed that the beginning and end of a gesture are always known, and the final classification is performed on the whole duration of the isolated gesture. To fulfill this condition in continues stream, temporal gesture segmentation is required, which is another active research topic. Therefore, it would be interesting to bridge a temporal sliding window-based method (e.g. RNN (Molchanov et al., 2016)) with the proposed approach to further explore its application for other computer vision tasks as well as other intelligent systems such as semantic translation of sign language in a live video. Another promising research path is gesture prediction. In fact, we are faced with numerous situations in which we must guess what gestures the person is about to do in the near future. Predicting the gesture before it actually occurs has a wide range of applications in autonomous robots, healthcare, and surveillance. Such a critical component can enable intelligent interaction with computers on a daily basis (Sadegh Aliakbarian et al., 2017). Nevertheless, it is not simple given the challenge that we need to capture the subtle details inherent in human movements that may imply the upcoming gesture as quickly as possible. We look forward to exploring this in the future.

Conflict of interest

The authors have no conflict of interest to declare.

Credit authorship contribution statement

Abdessamad Elboushaki: Conceptualization, Investigation, Methodology, Writing - review & editing, Formal analysis, Validation, Resources, Visualization. **Rachida Hannane:** Conceptualization, Methodology, Writing - review & editing, Validation, Investigation, Visualization. **Karim Afdel:** Conceptualization, Validation, Investigation, Supervision, Visualization. **Lahcen Koutti:** Conceptualization, Supervision, Visualization.

Acknowledgments

The authors would like to thank the associate editors and the anonymous reviewers for their valuable and insightful comments and suggestions,which have contributed a lot towards improving the contents and presentation of this article. We gratefully acknowledge the support of PPR2-2015 project grant with the donation of the NVIDIA Geforce GTX 1080 Ti GPU used for this research. This work was supported by the "Centre National pour la Recherche Scientifique et Technique (CNRST)" funded by Moroccan government under the grant no: 14UIZ2015.

References

Abidi, B. R., Zheng, Y., Gribok, A. V., & Abidi, M. A. (2006). Improving weapon detection in single energy x-ray images through pseudocoloring. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews, 36*(6), 784.

Almeida, S. G. M., Guimares, F. G., & Ramrez, J. A. (2014). Feature extraction in Brazilian sign language recognition based on phonological structure and using RGB-d sensors. *Expert Systems with Applications, 41*(16), 7259–7271.

Althloothi, S., Mahoor, M. H., Zhang, X., & Voyles, R. M. (2014). Human activity recognition using multi-features and multiple kernel learning. *Pattern recognition, 47*(5), 1800–1812.

Asadi-Aghbolaghi, M., Clapes, A., Bellantonio, M., Escalante, H. J., Ponce-Lopez, V., Baro, X., … Escalera, S. (2017). Deep learning for action and gesture recognition in image sequences: A survey. In *Gesture recognition* (pp. 539–578). Cham: Springer.

Bobick, A. F., & Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(3), 257–267.

Camgoz, N. C., Hadfield, S., Koller, O., & Bowden, R. (2016, December). Using convolutional 3d neural networks for user-independent continuous gesture recognition. In *Pattern recognition (ICPR), 2016 23rd international conference on* (pp. 49–54). IEEE.

Chai, X., Liu, Z., Yin, F., Liu, Z., & Chen, X. (2016, December). Two streams recurrent neural networks for large-scale continuous gesture recognition. In *Pattern recognition (ICPR), 2016 23rd international conference on* (pp. 31–36). IEEE.

Chen, C., Jafari, R., & Kehtarnavaz, N. (2015). Action recognition from depth sequences using depth motion maps-based local binary patterns. In *Applications of computer vision (WACV), 2015 IEEE winter conference on* (pp. 1092–1099). IEEE. January

Chen, C., Liu, K., & Kehtarnavaz, N. (2016b). Real-time human action recognition based on depth motion maps. *Journal of Real-time Image Processing, 12*(1), 155–163.

Chen, C., Liu, M., Zhang, B., Han, J., Jiang, J., & Liu, H. (2016a). 3d action recognition using multi-temporal depth motion maps and fisher vector. In *IJCAI* (pp. 3331–3337).

Cheng, H., Dai, Z., Liu, Z., & Zhao, Y. (2016b). An image-to-class dynamic time warping approach for both 3d static and trajectory hand gesture recognition. *Pattern Recognition, 55*, 137–147.

Cheng, H., Yang, L., & Liu, Z. (2016a). Survey on 3d hand gesture recognition. *IEEE Transactions on Circuits System Video Technology, 26*(9), 1659–1673.

Choi, H., & Park, H. (2014). A hierarchical structure for gesture recognition using RGB-d sensor. In *Proceedings of the second international conference on human-agent interaction* (pp. 265–268). ACM. October

Chron, G., Laptev, I., & Schmid, C. (2015). P-CNN: Pose-based CNN features for action recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 3218–3226).

Cirujeda, P., & Binefa, X. (2014, December). 4DCov: A nested covariance descriptor of spatio-temporal features for gesture recognition in depth sequences. In *3d vision (3DV), 2014 2nd international conference on vol. 1* (pp. 657–664). IEEE.

De Smedt, Q., Wannous, H., & Vandeborre, J. P. (2016, December). 3d hand gesture recognition by analysing set-of-joints trajectories. In *International workshop on understanding human activities through 3d sensors* (pp. 86–97). Cham: Springer.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In *Computer vision and pattern recognition, 2009. CVPR 2009. IEEE conference on* (pp. 248–255). IEEE.

Dikmen, M., Ning, H., Lin, D. J., Cao, L., Le, V., Tsai, S. F., … Lv, F. (2008, November). Surveillance event detection. *TRECVID*.

Diraco, G., Leone, A., & Siciliano, P. (2013). Human posture recognition with a time-of-flight 3d sensor for in-home applications. *Expert Systems with Applications, 40*(2), 744–751.

Dollar, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *Visual surveillance and performance evaluation of tracking and surveillance, 2005. 2nd joint IEEE international workshop on* (pp. 65–72). IEEE. October

Du, Y., Wang, W., & Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1110–1118).

Duan, J., Zhou, S., Wan, J., Guo, X., & Li, S. Z. (2016). Multi-modality fusion based on consensus-voting and 3d convolution for isolated gesture recognition. arXiv:1611.06689.

Duin, R. P. (2002). The combining classifier: To train or not to train? In *Pattern recognition, 2002* In *Proceedings. 16th International Conference on: vol. 2* (pp. 765–770). IEEE.

Edwards, M., & Xie, X. (2015). Generating local temporal poses from gestures with aligned cluster analysis for human action recognition. *UK computer vision student workshop (BMVW)*. BMVA Press. 1–1

Escalera, S., Athitsos, V., & Guyon, I. (2017). Challenges in multi-modal gesture recognition. In *Gesture recognition* (pp. 1–60). Cham: Springer.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580–587).

Guyon, I., Athitsos, V., Jangyodsuk, P., & Escalante, H. J. (2014). The chalearn gesture dataset (CGD 2011). *Machine Vision and Applications, 25*(8), 1929–1951.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hernndez-Vela, A., Bautista, M., Perez-Sala, X., Ponce-Lpez, V., Escalera, S., Bar, X., … Angulo, C. (2014). Probability-based dynamic time warping and bag-of-visual-and-depth-words for human gesture recognition in RGB-d. *Pattern Recognition Letters, 50*, 112–121.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.

Huang, J., Zhou, W., Li, H., & Li, W. (2015, June). Sign language recognition using 3d convolutional neural networks. In *Multimedia and expo (ICME), 2015 IEEE international conference on* (pp. 1–6). IEEE.

Huynh-The, T., Banos, O., Le, B. V., Bui, D. M., Lee, S., Yoon, Y., & Le-Tien, T. (2015). PAM-based flexible generative topic model for 3d interactive activity recognition. In *2015 international conference on advanced technologies for communications (ATC)* (pp. 117–122). IEEE. October

Ijjina, E. P., & Chalavadi, K. M. (2017). Human action recognition in RGB-d videos using motion sequence information and deep learning. *Pattern Recognition, 72*, 504–516.

Jacob, M. G., & Wachs, J. P. (2014). Context-based hand gesture recognition for the operating room. *Pattern Recognition Letters, 36*, 196–203.

Jain, A., Tompson, J., LeCun, Y., & Bregler, C. (2014, November). Modeep: A deep learning framework using motion features for human pose estimation. In *Asian conference on computer vision* (pp. 302–315). Cham: Springer.

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(1), 221–231.

Ji, Y., Ye, G., & Cheng, H. (2014, July). Interactive body part contrast mining for human interaction recognition. In *Multimedia and expo workshops (ICMEW), 2014 IEEE international conference on* (pp. 1–6). IEEE.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014, November). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on multimedia* (pp. 675–678). ACM.

John, V., Boyali, A., Mita, S., Imanishi, M., & Sanma, N. (2016, November). Deep learning-based fast hand gesture recognition using representative frames. In *Digital image computing: Techniques and applications (DICTA), 2016 international conference on* (pp. 1–8). IEEE.

Johnson, J. (2012). Not seeing is not believing: Improving the visibility of your fluorescence images. *Molecular Biology of the Cell, 23*(5), 754–757.

Joshi, A., Monnier, C., Betke, M., & Sclaroff, S. (2015). A random forest approach to segmenting and classifying gestures. *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on Vol. 1*. IEEE. May, pp. 1–7

Kang, B. N., Kim, Y., & Kim, D. (2017, July). Deep convolutional neural network using triplets of faces, deep ensemble, and score-level fusion for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 109–116).

Kim, H., Lee, S., Kim, Y., Lee, S., Lee, D., Ju, J., & Myung, H. (2016). Weighted joint-based human behavior recognition algorithm using only depth information for low-cost intelligent video-surveillance system. *Expert Systems with Applications, 45*, 131–141.

Klaser, A., Marszaek, M., & Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th british machine vision conference* (pp. 275–281). September

Koller, O., Ney, H., & Bowden, R. (2016). Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3793–3802).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision, 64*(2–3), 107–123.

LaViola, Jr, & J. , J. (2015, July). Context aware 3d gesture recognition for games and virtual reality. In *ACM SIGGRAPH 2015 courses* (p. 10). ACM.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Li, Y., Miao, Q., Tian, K., Fan, Y., Xu, X., Li, R., & Song, J. (2016, December). Large-scale gesture recognition with a fusion of RGB-d data based on the c3d model. In *Pattern recognition (ICPR), 2016 23rd international conference on* (pp. 25–30). IEEE.

Lin, L., Wang, K., Zuo, W., Wang, M., Luo, J., & Zhang, L. (2016). A deep structured model with radius margin bound for 3d human activity recognition. *International Journal of Computer Vision, 118*(2), 256–273.

Liu, J., Shahroudy, A., Xu, D., & Wang, G. (2016b). Spatio-temporal LSTM with trust gates for 3d human action recognition. In *European conference on computer vision* (pp. 816–833). Cham: Springer. October

Liu, L., & Shao, L. (2013). Learning discriminative representations from RGB-d video data. In *IJCAI vol. 1* (p. 3). August

Liu, M., & Liu, H. (2016). Depth context: A new descriptor for human activity recognition by using sole depth sequences. *Neurocomputing, 175*, 747–758.

Liu, Z., Zhang, C., & Tian, Y. (2016a). 3d-based deep convolutional neural network for action recognition with depth sequences. *Image and Vision Computing, 55*, 93–100.

Maqueda, A. I., del Blanco, C. R., Jaureguizar, F., & Garcia, N. (2015). Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns. *Computer Vision and Image Understanding, 141*, 126–137.

Miao, Q., Li, Y., Ouyang, W., Ma, Z., Xu, X., Shi, W., & Cao, X. (2017). Multimodal gesture recognition based on the resc3d network. In *ICCV workshops* (pp. 3047–3055). October

Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2015). Hand gesture recognition with 3d convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 1–7).

Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4207–4215).

Narayana, P., Beveridge, J. R., & Draper, B. A. (2018). Gesture recognition: Focus on the hands. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5235–5244).

Nishida, N., & Nakayama, H. (2015, November). Multimodal gesture recognition using multi-stream recurrent neural network. In *Pacific-rim symposium on image and video technology* (pp. 682–694). Cham: Springer.

Oreifej, O., & Liu, Z. (2013). HON4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 716–723).

Pisharady, P. K., & Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review. *Computer Vision and Image Understanding, 141*, 152–165.

Rautaray, S. S., & Agrawal, A. (2011, December). Interaction with virtual game through hand gesture recognition. In *2011 international conference on multimedia, signal processing and communication technologies* (pp. 244–247). IEEE.

Reyes, M., Dominguez, G., & Escalera, S. (2011, November). Feature weighting in dynamic time warping for gesture recognition in depth data. In *Computer vision workshops (ICCV workshops), 2011 IEEE international conference on* (pp. 1182–1188). IEEE.

Sadegh Aliakbarian, M., Sadat Saleh, F., Salzmann, M., Fernando, B., Petersson, L., & Andersson, L. (2017). Encouraging lstms to anticipate actions very early. In *Proceedings of the IEEE international conference on computer vision* (pp. 280–289).

Seger, R. A., Wanderley, M. M., & Koerich, A. L. (2014). Automatic detection of musicians ancillary gestures based on video analysis. *Expert Systems with Applications, 41*(4), 2098–2106.

Sipiran, I., & Bustos, B. (2011). Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer, 27*(11), 963.

Song, S., Lan, C., Xing, J., Zeng, W., & Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. February, AAAI, vol. 1, 2, pp.4263–4270.

Song, Y., Demirdjian, D., & Davis, R. (2011). Tracking body and hands for gesture recognition: Natops aircraft handling signals database. In *Automatic face & gesture recognition and workshops (FG 2011), 2011 IEEE international conference on* (pp. 500–506). IEEE. March

Song, Y., Morency, L. P., & Davis, R. (2012, June). Multi-view latent variable discriminative models for action recognition. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on* (pp. 2120–2127). IEEE.

Suk, H. I., Sin, B. K., & Lee, S. W. (2010). Hand gesture recognition based on dynamic bayesian network framework. *Pattern Recognition, 43*(9), 3059–3072.

Tang, D., Yusuf, B., Botzheim, J., Kubota, N., & Chan, C. S. (2015). A novel multimodal communication framework using robot partner for aging population. *Expert Systems with Applications, 42*(9), 4540–4555.

Tompson, J., Stein, M., Lecun, Y., & Perlin, K. (2014). Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG), 33*(5), 169.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 4489–4497).

Tran, D., Ray, J., Shou, Z., Chang, S. F., & Paluri, M. (2017). Convnet architecture search for spatiotemporal feature learning. arXiv:1708.05038.

Tung, P. T., & Ngoc, L. Q. (2014, December). Elliptical density shape model for hand gesture recognition. In *Proceedings of the fifth symposium on information and communication technology* (pp. 186–191). ACM.

Vondrick, C., Pirsiavash, H., & Torralba, A. (2016). Generating videos with scene dynamics. In *Advances in neural information processing systems* (pp. 613–621).

Wan, J., Guo, G., & Li, S. Z. (2016a). Explore efficient local features from RGB-d data for one-shot learning gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(8), 1626–1639.

Wan, J., Ruan, Q., Li, W., & Deng, S. (2013). One–shot learning gesture recognition from RGB-d data using bag of features. *The Journal of Machine Learning Research, 14*(1), 2549–2582.

Wan, J., Zhao, Y., Zhou, S., Guyon, I., Escalera, S., & Li, S. Z. (2016b). Chalearn looking at people RGB-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 56–64).

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., & Schmid, C. (2009). Evaluation of local spatio–temporal features for action recognition. In *BMVC 2009-british machine vision conference* (pp. 124–131). September

Wang, H., Wang, P., Song, Z., & Li, W. (2017b). Large-scale multimodal gesture recognition using heterogeneous networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3129–3137). October

Wang, P., Li, W., Gao, Z., Zhang, Y., Tang, C., & Ogunbona, P. (2017a). Scene flow to action map: A new representation for RGB-d based action recognition with convolutional neural networks. In *IEEE conf. on computer vision and pattern recognition* (pp. 1–10). February

Wang, P., Li, W., Liu, S., Gao, Z., Tang, C., & Ogunbona, P. (2016c). Large-scale isolated gesture recognition using convolutional neural networks. In *2016 23rd international conference on pattern recognition (ICPR)* (pp. 7–12). IEEE.

Wang, P., Li, W., Liu, S., Zhang, Y., Gao, Z., & Ogunbona, P. (2016b). Large-scale continuous gesture recognition using convolutional neural networks. In *Pattern recognition (ICPR), 2016 23rd international conference on* (pp. 13–18). IEEE.

Wang, P., Song, Q., Han, H., & Cheng, J. (2016a). Sequentially supervised long short–term memory for gesture recognition. *Cognitive Computation, 8*(5), 982–991.

Willems, G., Tuytelaars, T., & Van Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *European conference on computer vision* (pp. 650–663). Berlin, Heidelberg: Springer. October

Wu, D., Pigou, L., Kindermans, P. J., Le, N. D. H., Shao, L., Dambre, J., & Odobez, J. M. (2016b). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(8), 1583–1597.

Wu, J., Ishwar, P., & Konrad, J. (2016a). Two-stream CNNs for gesture-based verification and identification: Learning user style. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 42–50).

Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems* (pp. 802–810).

Yang, X., Zhang, C., & Tian, Y. (2012). Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM international conference on multimedia* (pp. 1057–1060). ACM. October

Yun, K., Honorio, J., Chattopadhyay, D., Berg, T. L., & Samaras, D. (2012, June). Two-person interaction detection using body-pose features and multiple instance learning. In *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on* (pp. 28–35). IEEE.

Zhang, C., & Tian, Y. (2015). Histogram of 3d facets: A depth descriptor for human action and hand gesture recognition. *Computer Vision and Image Understanding, 139*, 29–39.

Zhang, L., Zhu, G., Shen, P., Song, J., Shah, S. A., & Bennamoun, M. (2017b). Learning spatiotemporal features using 3DCNN and convolutional LSTM for gesture recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3120–3128). October

Zhang, Z., Wei, S., Song, Y., & Zhang, Y. (2017a). Gesture recognition using enhanced depth motion map and static pose map. In *Automatic face & gesture recognition (FG 2017), 2017 12th IEEE international conference on* (pp. 238–244). IEEE. May

Zheng, J., Feng, Z., Xu, C., Hu, J., & Ge, W. (2017). Fusing shape and spatio-temporal features for depth-based dynamic hand gesture recognition. *Multimedia Tools and Applications, 76*(20), 20525–20544.

Zhu, G., Zhang, L., Mei, L., Shao, J., Song, J., & Shen, P. (2016a). Large-scale isolated gesture recognition using pyramidal 3d convolutional networks. In *Pattern recognition (ICPR), 2016 23rd international conference on* (pp. 19–24). IEEE.

Zhu, G., Zhang, L., Shen, P., & Song, J. (2017). Multimodal gesture recognition using 3-d convolution and convolutional LSTM. *IEEE Access, 5*, 4517–4524.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., & Xie, X. (2016b). Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. *AAAI, 2*(5), 3697–3703. February