Robustness Analysis of 3D Convolutional Neural Network for Human Hand Gesture Recognition

Dang-Manh Truong, Huong-Giang Doan, Thanh-Hai Tran, Hai Vu, and Thi-Lan Le

***Abstract*—Recently, a number of methods for dynamic hand gesture recognition has been proposed. However, deployment of such methods in a practical application still has to face with many challenges due to the variation of view point, complex background or subject style. In this work, we deeply investigate performance of advanced convolutional neural networks for a specific case of hand gestures and evaluate how robust it is to above variations. To this end, we adopt an existing 3D convolutional neural network which was originally proposed for general human action recognition and obtained very competitive accuracy. We extend it to two-streams architecture (RGB and optical flow) and apply transfer learning on our dataset of hand gestures. To evaluate the robustness of the method, we design carefully a multi-view dataset that composes of five dynamic hand gestures in indoor environment with complex background. Experiments with single or cross view on this dataset show that background and viewpoint has strong impact on recognition robustness. In addition, the network's performances are mostly increased by multi-modality combinations and fine-tuning strategy. This analysis helps to make recommendation for deploying the method in real situation.**

***Index Terms*—Deep learning, convolutional neural network, dynamic hand gestures, optical flow, multi-view.**

I. Introduction

In recent years, hand gesture recognition has gained a great attention of researchers thanks to its potential applications such as sign language translation, human computer interactions [1]-[3], robotics, virtual reality [4], [5], autonomous vehicles [3]. Particularly, Convolutional Neuronal Networks (CNNs) [6] have been emerged as a promising technique to resolve many issues of the gesture recognition. Although utilizing CNNs has obtained impressive results [7], [8], there exists still many challenges that should be carefully carried out before applying it in reality. Firstly, hand is of low spatial resolution in image. However, it has high degree of freedom that leads to large variation in hand pose. Secondly, different subjects usually exhibit different styles with different duration when performing the same gesture (this problem is identified as phase variation). Thirdly, hand gesture recognition methods need to be robust to changes in viewpoint. Finally, a good hand gesture recognizer needs to effectively handle complex background and varying illumination conditions.

Motived by these challenges, in this paper, we comprehensively analyze critical factors which affect to performance of a common CNN through conducting a series of experiments and evaluations. The network's performances are examined under different conditions such as view-point's variations, muti-modality combinations and fine-tuning strategy. Through these quantitative measurements, the important limitations of deploying CNNs could be revealed. Results of these evaluations also suggest that only by overcoming these limitations, one could make the methods being able to be applied in real situation.

To this end, a common 3D convolutional neural network [9] (shortly named C3D) is adopted for the case of hand gesture recognition. C3D has been successfully utilized in many relevant works of the human action recognition. A specific character of C3D is that instead of 2D convolutional operation, C3D used 3D convolutional operation to take temporal information into account. This network composes of eight convolutional layers, six max pooling layers and three fully connected layers. In this paper, we firstly investigate performances of the C3D with a fine-tune strategy. This evaluation confirms that tuning parameters is an important procedure to achieve the high performances. The parameters of a C3D network for hand gestures recognition could be appropriately tuned by an underlying model learnt from a public human activity dataset.

The original C3D has only one stream which is RGB video. We then have extended the original C3D to two streams architecture by considering additional motion information which is optical flow. This extension opens a new opportunity to improve performance of C3D thanks to combinations of multi-modality that utilize not only single RGB image sequence, but also additional motion data. We then evaluate the robustness of the proposed approach by both two common fusion schemes: late and early fusion.

In addition, we are highly motivated by the fact that variation of view-points and complex background are real situations, particularly when we would like to deploy hand gesture recognition techniques automatic controlling home appliances using hand gestures. These factors ensure that strict constraints in common systems such as controlling's directions of end-users or context's background are eliminated. They play important roles for a practical system which should be maximizing natural feeling of end-user. To do this, we design carefully a multi-view dataset of dynamic

Manuscript received October 9, 2018; revised March 5, 2019. This material is based upon work supported by the Air Force Office of Scientific Research under award number FA2386-17-1-4056.

Dang-Manh Truong was with Hanoi University of Science Technology, Vietnam (e-mail: dangmanhtruong@gmail.com).

Huong-Giang Doan is with Electrical Power University Hanoi, Vietnam (e-mail: giangdth@epu.edu.vn).

Thanh-Hai Tran, Hai Vu, and Thi-Lan Le are with Hanoi University of Science Technology (Corresponding author: Thanh-Hai Tran; e-mail: thanh-hai.tran@mica.edu.vn, hai.vu@mica.edu.vn, Thi-Lan.Le@mica.edu.vn).

doi: 10.18178/ijmlc.2019.9.2.777

hand gestures in home environment with complex background. The experimental results show that the change of viewpoint could make a significant reduction of C3D's performances. It also indicates that removing background could increase recognition accuracy to 99.05%. This suggests constructing a success end-to-end network for both hand segmentation as well as recognition tasks.

Finally, other factors such as phase variations, length of a hand gesture sequence that could impact the C3D's performances are analyzed. As a consequent, we show that 3D convolutional neural network although has been proved to be very efficient for human action recognition recently, but this method is only evaluated on general actions in the wild. Its performance for hand gestures has never been confirmed in case of hand gestures with many challenges in real situations.

The remaining of this paper is organized as follows: Section II presents some approaches that use the deep learning technique for the hand gesture recognition problem. Section III describes our proposed approach. The experiments and results are analyzed in Section IV. Section V concludes this paper and proposes some future works.

II. Related Works

A significant number of researches in hand gesture recognition inherit from successful techniques for general human action recognition. These techniques are broadly divided into two categories: methods using hand-crafted features, and deep learning based methods. In this section, we will focus on the state of the art works that are closely related to our works: dynamic hand gestures recognition.

A. Hand-Crafted Feature Based Approaches

The main purpose of hand-crafted based approaches for dynamic hand gestures recognition is to design and compute the feature space for dynamic hand gesture representation in which hand gestures can be distinguished and recognized. As dynamic hand gestures possess motion and temporal characteristic, in order to leverage this characteristic, the recent works follow two main approaches. The works belonging to the first approach focus on designing accumulative features that are accumulatively computed from hand gesture's image sequence while the second approach represents dynamic hand gestures by its evolution along with the time. The work in [10] falls to the first approach. In order to capture the hand motion, the authors proposed to integrate Edge Enhanced Depth Motion Map (EEDMM) and Histogram of Gradient descriptor. In this, after edge enhanced depth motion map extraction and dynamic temporal pyramid (DTP) organization, the authors applied Histogram of Gradients (HoG) to generate vectored representation of the two levels of EEDMM. Finally, a SVM (Support Vector Machine) classifier is trained and utilized for classification.

In the second approach [1], the authors represented hand gesture sequences in a spatial-temporal feature space. In this space, the hand shapes are exploited through an isometric feature mapping algorithm ISOMAP (ISOmetric MAPing). The dominant trajectories of the hand are extracted by connecting key-points tracked using KLT (Kanade-Lucas-Tomasi) technique. To deal with phase variation in dynamic hand gesture recognition, an interpolation scheme is deployed on each dimension to reconstruct the phase-normalized image sequence.

B. Deep Learning Based Approaches

Recently, with the impressive results obtained by deep learning techniques in numerous computer vision tasks [7], there is a growing trend toward feature representations learned by deep neural networks for dynamic hand gesture recognition [6], [8], [11]. In [11], the authors combined both deep learning (CNN) and hand-crafted features (HoG) for hand shape representation. Hidden Markov models were trained on these complex features to model and recognize dynamic hand gestures for the interface of in-vehicle infotainment systems. In contrast to the aforementioned works that adopted HMM for capturing temporal aspect of dynamic hand gestures, the method in [6], [12] try to capture simultaneously spatial and temporal information by applying a 3D-CNN on the whole video sequence. In this, they interleaved depth and intensity channels to build normalized spatial-temporal volumes, and trained two separate sub-networks, a High-Resolution Network (HRN) and a Low-Resolution Network (LRN), with these volumes. The final class-membership probabilities for the gesture are computed by multiplying the class-membership probabilities from the two networks element-wise. To avoid over-fitting problem, the authors introduced space-time video augmentation techniques.

Taking into account that most previous methods either employ pre-segmented video sequences or treat detection and classification as separate problems, in [8] the authors proposed an algorithm for joint segmentation and classification of dynamic hand gestures from continuous depth, color and stereo-IR data streams. In this method, C3D [9] is employed to modeling the local spatial-temporal information while RNN (Recurrent Neural Network) is used for capturing the global one. Features extracted from C3D-RNN which is gone through a connectionist temporal classification (CTC) for gesture classification that is designed to enable gesture classification to be based on the nucleus phase of the gesture without requiring explicit pre-segmentation.

Previous studies have shown the effectiveness of C3D - 3D convolutional neural networks for hand action recognition in general [13] and gesture recognition in particular [8]. However, these require multi-modal hand gesture image sequences that are not always available. Moreover, the performance of C3D is not deeply analyzed for dynamic hand gestures. This paper aims at answering the two following question: (1) Would C3D, a deep architecture tested on general action recognition datasets, still be suitable for hand gesture recognition where hand has a relatively low spatial resolution? (2) How does viewpoint variation affect the performance of C3D? In order to further increase C3D recognition performance, we also study some techniques such as frame sampling in video, as well as combining OF features to RGB.

C. Review of 3C Convolutional Neural Network (C3D)

In this sub-section, we will review briefly the 3D

convolutional neural network (C3D) [9] that will be utilized in our work. C3D composes of 8 convolutional layers, 5 max pooling and 2 fully connected layer followed by a soft-max output layer. The architecture of C3D is illustrated in Fig. 1. In this network, the convolutional operation is 3D convolution which aims to capture both spatial and temporal information of video.

Fig. 1. Original architecture of C3D network proposed by [9].

The 3D convolution is obtained by convolving a 3D kernel to the cube formed by stacking multiple contiguous frames together. By this way, the feature maps in convolution layer are connected to multiple contiguous frames in the previous layer then motion cue is extracted. In C3D implementation and experimentation, the authors have indicated that $3\times3\times3$ is the best size for the kernel. All of 3D convolution filters are $3\times3\times3$ with stride $1\times1\times1$. All pooling layers are $2\times2\times2$ with stride $2\times2\times2$ except pool1 has kernel size $2\times2\times1$ and stride $2\times2\times1$. The number of filters for 5 convolution layers from 1 to 5 are 64, 128, 256, 256, 256, respectively.

C3D has been trained on Sport-1M train split. From every video, five 2-second long clips have been extracted randomly and all clips are resized to have frame size of 128×171. Then each 16 frames clip is randomly cropped to produce $16\times128\times171$ volumes input to the network for training.

Given an input video $W\times H\times L$ where W, H are weight and height of frame, L is the video length, to extract features, each video is split into 16 frame long clips with 8 frame overlap between two consecutive clips. These clips will be passed to C3D to extract features. These features of all 16 frame clips are averaged following a L2-normalization. A C3D network could be used as a classifier but also as a feature extractor. In our work, we will use C3D as feature extractor.

III. Our Approach for Hand Gesture Recognition

A. The Proposed Framework

Because hand gestures are characterized both by change of hand posture and hand movement. Therefore, additional information about hand motion could help to improve hand recognition accuracy. In this work, our proposed framework for hand gesture inherits our previous work on action recognition [13]. This work extended the original C3D architecture to two-streams architecture where one stream is RGB (as original C3D) and one stream is optical flow. The framework for hand gesture recognition is illustrated as in Fig. 2. It composes of three main components:

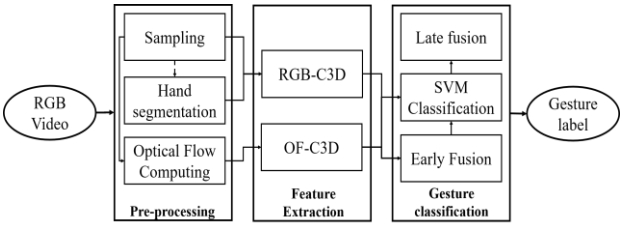

Fig. 2. General framework of proposed method for hand gesture recognition.

- **Pre-processing of hand gesture data:** This component consists of different sub-components such as sampling by splitting the whole video of different length into a fixed length clips that would be able to input into the neural network; hand segmentation frame by frame for investigating the impact of background; computing and stacking optical flow for the second stream of the network.
- **Feature extraction for hand gesture representation:** This component composes of a two-streams 3D convolutional neural network. One stream is RGB and one is optical flow. Both streams have similar architecture to C3D. We will use this network to extract features from both RGB and optical flow channels for hand gesture representation.
- **Hand gesture classification:** This component classifies features extracted in the previous components to produce the label of gestures.

In the following, we will explain in more detail each component of the proposed framework.

B. Pre-processing

1) Sampling

An important and common problem of hand gesture recognition and video classification is phase variation. That means the length of a gesture class could be different from subject to subject. However, the C3D network takes only a fixed length video as input. The original solution proposed in [9] was to divide the whole video into non overlapped 16-frame clips. The C3D network then receives each clip as input to generate a feature vector (or probability score vector, if the last layer is used). Finally, the feature vector/scores of all the clips are averaged for further classification or decision making.

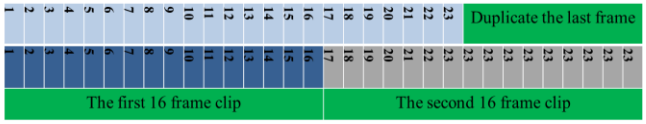

Fig. 3. Sampling method used by [9].

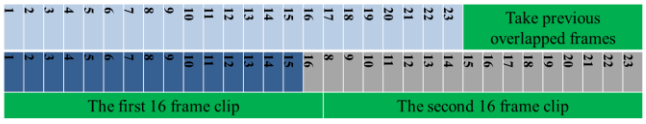

Fig. 4. The proposed sampling method.

According to this way, one problem that appears frequently is if the length of video is not dividable to 16, the remaining clip will have less than 16 frames. To have a 16 frame clip, the last frames of this clip will be duplicated. An example is shown in Fig. 3, all of the last eight frames duplicate the 23th frame. In case where the length of the whole video is short (that is always the case of hand gestures), the network could potentially learn "static frames". To overcome this situation, we propose to take previous overlapped frames. For example, in Fig. 4, the last frames are borrowed from the 8^{th} frame in the 1st clip to the 23^{th} frame. This would undoubtedly mitigate the "static effect". We also implement two methods for sampling 16 frames clip from the video. The first is to randomly take 16 frames according to [14]. The second method is to computing the most distinctive key-frames though calculating the

similarity between consecutive frames [15]. All of these methods will be compared in experimental section.

2) Hand segmentation

To analyze the effect of background on the performance of recognition algorithm, hand should be segmented from background before inputting to the neural network. Any algorithm of hand segmentation can be applied, from the simplest one basing on skin to more advanced techniques such as instance segmentation of Mask R-CNN [16]. In this work, we just apply an interactive segmentation tool[1] to manually detect hand from image. This precise segmentation helps to avoid any additional effect of automatic segmentation algorithm that could lead to wrong conclusion. Fig. 5 illustrates an original video clip and the corresponding segmented one annotated manually.

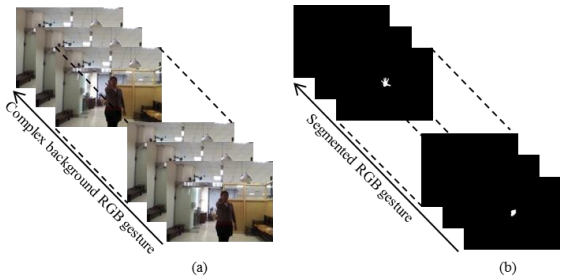

Fig. 5. (a) Original video clips; (b) The corresponding segmented video clip.

3) Computing and stacking optical flow

Although the 3D convolutional operator takes temporal information into account, as the size of the filter is only 3×3×3 meaning it takes only the previous frame, the current frame and the next frame. This cannot represent long term movement of hand. In our previous work, we have extended C3D to two-streams architecture which composes of RGB and optical flow streams [13]. We now present how to compute and stacking optical flow. To compute optical flow, we use an existing algorithm[2]. It takes two consecutive frames I_t and I_{t+1} and produces two optical flow maps according to x and y direction: O_{tx}, O_{ty}. The input of C3D is a volume of size 3×16×121×178 (three channels R, G, B, 16 is the number of frames of the clip, 121, 178 are height and width of the frame respectively. We then stack our optical flow with the same format. Specifically, we consider O_{tx}, O_{ty} as two first channels. We adjust a black frame (the value of all pixels equals to 0) as the third channels. Then we obtain a volume of optical flow of a 16 frame clip with dimension similar as RGB stream. Fig. 6 illustrates an example of two consecutive RGB images and its optical flow in two directions.

Fig. 6. From left to right: two consecutive frames and obtained optical flows in x and y directions respectively.

[1] http://www.cs.cmu.edu/ mohitg/segmentation.htm
[2] https://github.com/feichtenhofer/gpu flow

C. Two-Streams 3D Convolutional Neural Network

1) Architecture of two-streams 3D CNN

According to our previous work presented in [13], even though C3D was designed with 3D kernels where one dimension is temporal, the filter size is only 3×3×3 which seems to be unable to represent long term movement. This might be resolved if additional information, such as optical flow (OF), is incorporated into the algorithm. From this observation, and inspired by the two-streams architecture in [17] and [18], we propose two-streams C3D architecture for hand gesture recognition as illustrated in Fig. 7.

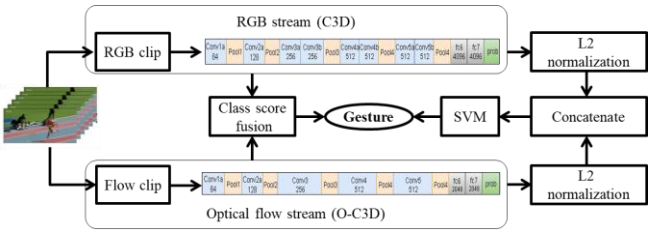

Fig. 7. Two stream C3D neural network architecture.

On the top is the RGB stream with the original C3D architecture that we name it as RGB-C3D. On the bottom is the OF stream with a simplified version of C3D named O-C3D to avoid over fitting. The reason for this is because the optical flow stream is trained on a UCF101, while the original C3D architecture (used in the RGB stream) is trained on Sport1m, which is a much larger dataset. O-C3D has only five space-time convolutional layers with 64, 128, 256, 256, 256 kernels followed by three fully connected (FC) layers of size 2048, and a final soft-max output layer. Similar to RGB-C3D, all O-3D convolution kernels are of size 3×3×3 with stride 1×1×1. Max pooling kernels are of size 2×2×2 except for the first one, which is of size 2×2×1 with stride 2×2×1.

2) Transfer learning on hand gesture dataset

An important problem of deep learning in general is to train a network in order to use it for classification. However, deep learning architectures generally require large datasets and this technique is not efficient for small datasets. Normally, it is better to use the network parameters that have been trained from a big dataset. In this case, there are several solutions to tackle this problem:

- Using ConvNet as a fixed feature extractor, then use a classifier to classify the features. However, using ConvNet as a fixed feature extractor usually does not give satisfying results when the target dataset is radically different from the original dataset. This is because the extracted features might be too specific to the original dataset and are therefore not of much use when being applied to the target dataset.
- Fine tuning several or all layers of the network, with parameters initialized from training on a larger dataset. This is called transfer learning and is common practice in deep learning. Therefore, in this paper, we perform fine tuning on gesture dataset. We have to fine-tune both RGB-C3D and O-C3D networks.

Fine tuning RGB-C3D: With the RGB stream, we use transfer learning of RGB-C3D with pre-trained model of C3D on RGB frames of Sport1m dataset. One question is how to

choose layers to perform fine-tuning in order to obtain the best gesture recognition results. We try to evaluate and compare between fine-tuning only the fully connected layers and fine-tuning all layers of the network. Experiments show that fine tuning the whole network obtains higher accuracy (96.67%) than fine tuning only fully connected layers (64.10% of accuracy) on the same dataset. So for all remaining evaluation, we will use the overall fine-tuned RGB-C3D.

Fine tuning O-C3D: The original C3D model introduced in [9] was trained using Sport1m dataset. However due to its large size there has not been any attempt to calculate the optical flows of the video frames in this dataset. Researchers usually use OF data from UCF101 dataset, which is much smaller but can potentially leads to over-fitting if not trained properly. In this paper we will also investigate and compare the cases of using a pre-trained model from RGB data of Sport1m, as well as pre-trained from OF data of UCF101. The results in Table I show that fine tuning OF-C3D with pre-trained model on RGB data of Sport1m is better than fine tuning C3D with pre-trained model on optical flow of UCF101. However, for two-streams architecture it is better to use pre-trained model from optical flow data of UCF101 for fine tuning the O-C3D.

TABLE I: ACCURACY (%) WITH DIFFERENT FINE TUNING MODES AT CAMERA VIEW K

Method	RGB-Sport1M	OF-UCF101
OF	93.28	89.88
OF-RGB (early fusion)	99.33	100
OF-RGB (late fusion)	93.11	100

D. Hand Gesture Classification

We fine tune separately each C3D network for RGB and Optical flow stream. Then we use these networks as features extractors. For a given video, we apply sampling techniques to split the video into 16 frames clips. We then input each clip to RGB-C3D and OF-C3D and we take features vectors at FC6 layers. Each C3D network outputs a 4096 D feature vector which is L_2 normalized. We do this for every clip and take average feature vector of all clips. Finally, we perform two kinds of fusion: early fusion and late fusion.

- Early fusion: We concatenate the two feature vectors to generate a 8192-D feature vector. We use linear SVM classifier for classify the features.
- Late fusion: We apply linear SVM classifier for each 4096-D feature vector extracted from each stream. This outputs a confidence score. We then use maximal score from both streams to make the final decision.

IV. EXPERIMENT

A. Materials and Data Acquisition

As mentioned previously, the evaluation on robustness of hand gesture recognition was not considered in the literature. Therefore, there does not exist a dataset dedicated to this problem. In our work, we carefully design a dataset which is collected from multiple camera viewpoints in indoor environment with complex background.

Our dataset consists of five dynamic hand gestures which correspond to controlling commands of electronic home appliances: G_1-ON/OFF, G_2-UP, G_3-DOWN, G_4-LEFT and G_5-RIGHT. Each gesture is combination of hand movement following a pre-defined direction and changing of hand shape. For each gesture, hand starts from one position with close posture, it opens gradually at half cycle of movement then closes gradually to end at the same position and posture as described in [1]. Fig. 8 illustrates the movement of hand and changes of postures during gesture implementation.

Five cameras $\{K_1, K_2, K_3, K_4, K_5\}$ are setup at five various positions in a simulation room of 4m×4m with a complex background (Fig. 9). This setup aims to capture hand gestures under multiple different viewpoints at the same time. Subjects are invited to stand at a nearly fixed position in front of five cameras at an approximate distance of 2 meters. Five participants ($\{P_i;\ i=1...5\}$) (3 males and 2 females) are voluntary to perform gestures. Each subject implements one gesture from three to six times.

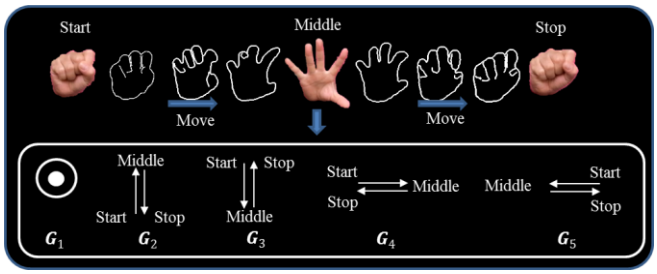

Fig. 8. Five defined dynamic hand gestures.

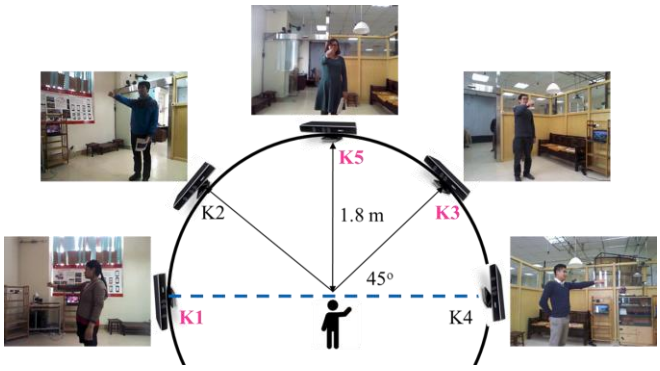

Fig. 9. Setup environment of different viewpoints.

Totally, the dataset contains 375 (5 views× 5 gestures × 5 subjects × (3 to 6 times)) dynamic hand gestures with frame resolution is set to 640×480. All views have the same number of gestures with others. (2) In each view, the number of gestures of G_3 is highest (33 gestures), G_1 and G_4 have the same number of gestures (26 gestures) while the number of G_2 and G_5 are 22, 23 gestures, respectively. Each gesture's length varies from 50 to 126 frames (depending on the speed of gesture implementation of different users) as present in Table II. We could see that G_1 has the smallest frame numbers (from 33 to 66 frames per gesture) while other gestures fluctuated approximately from 60 to 120 frames per a gesture. This leads to a different number of frames to be processed and create big challenges for phase synchronization between different classes and gestures.

Because the acquisition of data is continuous. Therefore, each video can contain several gestures. To evaluate the

gesture classification and impact of background, we have to perform two annotations: gesture spotting and hand segmentation. Gesture spotting means that we determine the starting and the ending frames of each gesture from video. For hand segmentation, we use an iterative segmentation tool[1] to segment hand region frame by frame. Hand segmentation requires huge time and also sometime it is very difficult because of motion blur when hand move very fast and low resolution of the hand. Due to this reason, in this work, we do experiment with three views K_1, K_3 and K_5 among five views. However, these three views could be sufficient to analyze influence of viewpoints: K_5 is in front of the subject; K_3 view make an angle of 45^0 with respect to K_5. K_1 view make an angle of 90^0 with respect to K_5. K_3 view make an angle of 135^0 with respect to K_1.

TABLE II: AVERAGE NUMBER OF FRAMES PER GESTURE

	P_1	P_2	P_3	P_4	P_5
G_1	49.25	51	33	54	66.3
G_2	61.75	115	49.7	104.75	126.25
G_3	55.8	98.7	118.5	106.5	103.3
G_4	70.25	101.7	69	108.8	107.25
G_5	59.5	83	72.7	92.7	102.5

B. Evaluation Procedure

In this paper, we use leave-one-subject-out-cross-validation as described in [1] in order to prepare data for training and testing in our evaluations. Which each subject is used as the testing set and the others as the training set. The results are averaged from all iterations. With respect to cross-view, the testing set is evaluated on different viewpoints with the training set.

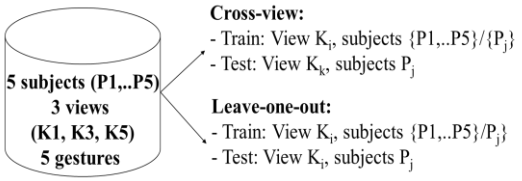

Fig. 10. Evaluation procedure.

The evaluation metric used in this paper is presented in (1) following:

$$\text{Accuracy} = \frac{\sum \text{corrects}(\%)}{\sum \text{total}} \quad (1)$$

C. Experimental Results

In this section, we will present our experimental results. Our objective is to evaluate impact of viewpoint, background, modality and sampling solution on the performance of our proposed method.

1) Evaluation of sampling solutions

We evaluate three sampling solutions: i) considering all 16 frames clips from the whole video and average the feature vectors of these 16 frames clips; ii) randomly selecting 16 ordered frames from the whole video; iii) selecting 16 discriminative frames from the whole video.

Table III shows the results of different sampling. We can see that all solutions give comparable average accuracy. One remarkable finding is among three sampling solutions, using sixteen randomly-sampled frames gives lightly higher results on both single views (with highest value at 97.33%) and the average result (with 65.05%). While this scheme requires the smallest computational time with the average calculation time could be reduced approximately L/16 where L is the number of frames of the considered video. We also observe that the convolutional neural network is quite independent of phase variation that subjects perform gestures.

TABLE III: ACCURACY (%) COMPARISON OF SAMPLING SOLUTIONS

	16 frames clips [9]			16 Randomly selected [14]			16 key frames [18]		
	K_1	K_3	K_5	K_1	K_3	K_5	K_1	K_3	K_5
G_2	**76.27**	45.60	50.07	**75.59**	56.60	41.97	**71.15**	51.79	33.61
G_3	47.76	**93.60**	76.04	48.78	**95.34**	77.45	47.98	**95.34**	78.79
G_4	30.41	65.77	**96.67**	36.15	56.25	**97.33**	38.31	61.51	**96.67**
Avr	64.68			**65.05**			63.90		

2) Analysis of the impact of background

Table IV shows the comparison of accuracy obtained on RGB stream with and without background removing. We could see that average accuracy increased significantly by 11.33% (from 64.68% to 76.01%). The highest accuracy (99.05%) has been obtained at frontal view (K_5). This confirmed that hand segmentation is an essential step to improve performance of recognition algorithm.

TABLE IV: ACCURACY (%) COMPARISON OF REMOVING AND WITHOUT REMOVING BACKGROUND OF RGB STREAM

	RGB			Segmented RGB		
	K_1	K_2	K_3	K_1	K_2	K_3
K_2	**76.27**	45.60	50.07	**89.97**	65.37	54.09
K_3	47.76	**93.60**	76.04	50.67	**99.38**	89.84
K_4	30.41	65.77	**96.67**	42.49	93.29	**99.05**
Avr	64.68			**76.01**		

3) Analysis of the impact of view point

From Table V, we observe that there is a big difference in performance among camera view-points. The highest performance was always obtained at frontal view. That means that when the subject stands in front of the camera, hand postures and hand movements are the most distinctive that helps to recognize best. The performance reduces gradually when the viewpoint deviates an angle of 45^0(K_3) and drastically reduces when the viewpoint deviates an angle of 90^0 (K_1) w.r.t frontal view (K_5). Another finding is that single view evaluation (training and testing on the same view) achieved better performance than cross view evaluation (training and testing views are different). The robustness of the method could be still acceptable when the deviation angle is smaller than 45^0.

4) Analysis of modalities

Compared to RGB stream, optical flow gives competitive performance (see Table V). A significant improvement has been obtained with cross view evaluation where the training view is K_3 and testing view is K_5. The accuracy obtained by RGB stream is only 65.77% while the accuracy obtained by

optical flow stream is 89.47%. This result shows that in some situations, information of movement is very important and could be a supplemental modality for gesture recognition.

TABLE V: ACCURACY (%) COMPARISON OF RGB STREAM AND OPTICAL FLOW STREAM

	RGB			Optical Flow		
	K_1	K_2	K_3	K_1	K_2	K_3
K_2	**76.27**	45.60	50.07	**70.98**	35.45	39.31
K_3	47.76	**93.60**	76.04	47.63	**95.68**	71.51
K_4	30.41	65.77	**96.67**	38.49	89.47	**93.28**
Avr		64.68			64.64	

5) *Analysis of two-streams fusion*

Table VI shows results obtained with early fusion and late fusion of both RGB and optical flow stream. Compared to using single stream (RGB or optical flow), fusing both streams helps to improves the average accuracy by about 3.11%. Late fusion is lightly better than early fusion. In both fusions, highest accuracy (100%) has been obtained at single view validation of K_5 (frontal view).

TABLE VI: ACCURACY (%) COMPARISON OF EARLY FUSION AND LATE FUSION

	RGB-OF early fusion			RGB-OF late fusion		
	K_1	K_2	K_3	K_1	K_2	K_3
K_2	**75.67**	55.30	47.53	**74.10**	52.07	45.61
K_3	47.72	**94.43**	81.12	51.59	**94.36**	80.07
K_4	36.70	68.17	**100.0**	41.55	70.83	**100.0**
Avr		67.40			67.79	

V. DISCUSSION AND CONCLUSION

In this paper, we have proposed an approach for human hand gesture recognition using two streams 3D convolutional neural network. Then we have deeply investigated the robustness of the method for hand gesture recognition. Experiments were conducted on a multi-view dataset that was carefully designed and constructed by ourselves. Different evaluations lead to some following conclusions: i) Concerning viewpoint issue, the proposed method has obtained highest performance with frontal view, it is still good when view point deviates in the range of 45^0 and reduced drastically when the viewpoint deviates from 90^0 to 135^0. So one of recommendation is to learn dense viewpoints so that testing view point could avoid huge difference compared to learnt views; ii) Background has impact on performance of recognition method. It is recommended to remove background before doing hand gesture recognition, so a step of hand segmentation would be preferable; iii) the method is quite independent of hand phase variation, a random sample could obtain a comparable result while reducing computational time; iv) Two stream architecture works better than single stream.

These conclusions open some directions in future works. Firstly, we will complete our annotation and evaluation of all of five views and compare our methods with other existing ones. We also perform automatic hand segmentation and integrate into unified framework. Some adaption of the network to face more with change of viewpoint also will be considered. One possibility is to learn more viewpoints and try to match the unknown gestures with the gestures having the most similar viewpoint in the training set. Another possibility is to extract invariant human pose features.

REFERENCES

[1] H. Doan, H. Vu, and T. Tran, "Dynamic hand gesture recognition from cyclical hand pattern," in *Proc. IAPR International Conference on Machine Vision Applications (MVA)*, 2017, pp. 97–100.

[2] M. M. Hasan and P. K. Mishra, "Robust gesture recognition using gaussian distribution for features fitting," *International Journal of Machine Learning and Computing*, vol. 2, no. 3, pp. 266-273, 2012.

[3] H. Takimoto, J. Lee, and A. Kanagawa, "A robust gesture recognition using depth data," *International Journal of Machine Learning and Computing*, vol. 3, no., pp. 245-2492, 2013.

[4] Q. Chen, A. El-Sawah, C. Joslin, and N. D. Georganas, "A dynamic gesture interface for virtual environments based on hidden markov models," in *Proc. IEEE International Workshop on Haptic Audio Visual Environments and Their Applications*, 2005, pp. 109-114.

[5] V. Dissanayake, S. Herath, S. Rasnayaka *et al*., "Real-time gesture prediction using mobile sensor data for VR applications," I*nternational Journal of Machine Learning and Computing*, vol. 6, no. 3, pp. 215-219, June 2016.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. International Conference on Neural Information Processing Systems*, 2012, vol. 1, pp. 1097–1105.

[7] P. Molchanov, S. Gupta, K. Kim, and J. Kautz, "Hand gesture recognition with 3d convolutional neural networks," in *Proc. CVPRW*, 2015, pp. 1–7.

[8] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks," in *Proc. CVPR*, 2016, pp. 4207–4215.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proc. ICCV*, 2015, pp. 4489–4497.

[10] C. Zhang and Y. Tian, "Edge enhanced depth motion map for dynamic hand gesture recognition," in *Proc. CVPRW*, 2013, pp. 500–505.

[11] N. Deo, A. Rangesh, and M. M. Trivedi, "In-vehicle hand gesture recognition using hidden markov models," *ITSC*, pp. 2179–2184, 2016.

[12] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," *FG*, vol. 1, pp. 1–8, 2015.

[13] V. Khong and T. Tran, "Improving human action recognition with two-stream 3d convolutional neural network," *MAPR*, pp. 1–6, 2018.

[14] L. Jing, X. Yang, and Y. Tian, "Video you only look once: Overall temporal convolutions for action recognition," *Journal of Visual Communication and Image Representation*, vol. 52, pp. 58–65, 2018.

[15] T. Le, V. Nguyen, T. Tran, V. Nguyen, and T. Nguyen, "Temporal gesture segmentation for recognition," *ComManTel*, pp. 369–373, 2013.

[16] K. He, G. Gkioxari, P. Doll ár, R. Girshick, "Mask r-cnn," in *Proc. ICCV*, 2017, pp. 2980–2988.

[17] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *CoRR*.

[18] T. L. Le, V. N. Nguyen, V. T. Nguyen, T. T. Nguyen *et al*., "Temporal gesture segmentation for recognition," in *Proc. 2013 International Conference on Com. Man. Tel*., IEEE, 2013, pp. 369–373.

Manh-Truong Dang was born in Vietnam. He received B.E. degree in information technology in 2018 from Hanoi University of Science and Technology, Vietnam. His research area are in image processing, computer vision.

Huong-Giang Doan received B.E. degree in instrumentation and industrial informatics in 2003, M.E. in Instrumentation and automatic control system in 2006 and Ph.D. in Control engineering and automation in 2017, all from Hanoi University of Science and Technology, Vietnam. She is working at Electric Power University, Ha Noi, Vietnam.

Thanh-Hai Tran received B.E. degree in information technology in 2001 from Hanoi University of Science and Technology (HUST) in 2001. She holds a M. S. degree and a Ph.D degree in Imagery vision robotic from Grenoble INP in 2002 and 2006 respectively. She is interested in image processing, computer vision, robot control through image, enhanced reality, machine learning.

Hai-Vu received B.E. degree in electronic and telecommunications in 1999, M.E. in information processing and communication in 2002 both from Hanoi University of Technology. He received his Ph.D. in computer science from Graduate School of Information Science and Technology, Osaka University, 2009. He is interested in medical imaging techniques, mainly video capsule endoscopy analysis for diagnostic assistance, computer vision supporting visually impaired people, human-computer/robotic interaction, computer vision in agricultural engineering.

Thi-Lan Le received her B.E. degree in Information technology in 2003 and M.E. in computer science in 2005, both from Hanoi University of Technology, Vietnam. She received her Ph.D. in Computer Science FROM the University of Nice – Sophia Antipolis at I.N.R.I.A in the ORION project team, 2009. She is interested in video surveillance indexing and retrieval, human-computer/robotic interaction, content-based image indexing and retrieval.