An End-to-End Model of ArVi-MoCoGAN and C3D with Attention Unit for Arbitrary-view Dynamic Gesture Recognition

Huong-Giang Doan[1], Hong-Quan Luong[2], Thi Thanh Thuy Pham[3]
Faculty of Control and Automation Electric Power University, Ha Noi, Viet Nam[1]
MQ Information and Communication Technology Solutions JSC, Ha Noi, Viet Nam[2]
Faculty of Information Security, Academy of People Security, Ha Noi, Viet Nam[3]

Abstract*—Human gesture recognition is an attractive research area in computer vision with many applications such as human-machine interaction, virtual reality, etc. Recent deep learning techniques have been efficiently applied for gesture recognition, but they require a large and diverse amount of training data. In fact, the available gesture datasets contain mostly static gestures and/or certain fixed viewpoints. Some contain dynamic gestures, but they are not diverse in poses and viewpoints. In this paper, we propose a novel end-to-end framework for dynamic gesture recognition from unknown viewpoints. It has two main components: (1) an efficient GAN-based architecture, named ArVi-MoCoGAN; (2) the gesture recognition component, which contains C3D backbones and an attention unit. ArVi-MoCoGAN aims at generating videos at multiple fixed viewpoints from a real dynamic gesture at an arbitrary viewpoint. It also returns the probability that a real arbitrary view gesture belongs to which of the fixed-viewpoint gestures. These outputs of ArVi-MoCoGAN will be processed in the next component to improve the arbitrary view recognition performance through multi-view synthetic gestures. The proposed system is extensively analyzed and evaluated on four standard dynamic gesture datasets. The experimental results of our proposed method are better than the current solutions, from 1% to 13.58% for arbitrary view gesture recognition and from 1.2% to 7.8% for single view gesture recognition.

Keywords*—*Dynamic gesture recognition; attention unit; generative adversarial network

I. Introduction

Human gesture recognition is an attractive field in computer vision with many applications such as human computer interaction, human behavior analysis, intelligent surveillance, and virtual reality [1], [2]. A recognition system could use (1) static gestures and (2) dynamic gestures. In comparison with static gesture recognition, dynamic recognition is much more challenging. Dynamic gesture recognition at multi-view points has received much research attention in recent years because of its closeness to real-world applications.

Several methods have been proposed for dynamic gesture recognition. They range from traditional machine learning algorithms, such as Dynamic Time Warping (DTW) [3], Hidden Markov Model (HMM) [4], etc., to deep learning architectures, such as 2D CNN (2-Dimensional Convolutional Neural Network) [5], 3D CNN or C3D (3-Dimensional Convolutional Neural Network) [6]. 2D CNNs utilize two-dimensional convolution and pooling solutions to process gesture data. However, 2D CNNs only model the spatial domain but not the time domain of gesture data. Thus, they are more suitable for static gesture recognition than dynamic gesture recognition. In order to overcome this weakness of 2D CNNs, 3D CNNs or C3D are proposed for modeling both spatial and temporal information from videos. C3D networks achieve promising results in dynamic gesture recognition with deep and complex enough network structures. However, increasing the network's depth and complexity indefinitely can cause degradation problems and increase the computing cost. In addition, one of the main obstacles to dynamic gesture recognition by deep learning models is the scarcity of available dynamic gesture datasets, especially those that contain a diversity of gestures at multiple view points and movements [7], [8]. In order to overcome this challenge, several data augmentation techniques have been proposed. They range from traditional techniques, such as rotate, slip, strength, and so on, to more complex techniques, such as the Generative Adversarial Network (GAN). For gesture data generation, GAN networks are mainly used to generate static gesture images from single viewpoint [9], [10] or multiple viewpoints [11], [12]. Some GAN-based networks are proposed for making synthetic videos of gestures or dynamic gestures. However, it is still extremely difficult to produce high-quality videos of dynamic gestures. The results of the existing generative models for dynamic gestures are blurry and inconsistent [13], [14]. This is caused by the fact that the input for the Generator networks in these works is mainly noise signals. The dynamic gesture generation at arbitrary viewpoints has not been much exploited [15]. In addition, the experiments with GAN-generated images or videos for an arbitrary-view recognition system are less considered or limited to skeleton images or simple skeleton frame sequences [16].

In this paper, a novel end-to-end system is proposed for (1) generating synthetic videos at multiple fixed viewpoints from a real dynamic gesture at an arbitrary viewpoint, and (2) classifying dynamic gestures from multi-view synthetic dynamic gestures. The proposed system contains two main components, and each is responsible for a certain task as follows:

- The first component is the improved version of the Vi-MoCoGAN architecture in [13], named ArVi-MoCoGAN. It is different from Vi-MoCoGAN in [13] and other available GAN-based approaches for gesture generation, in which the input is normally noise signal and the output is dynamic/static gesture. In ArVi-MoCoGAN, the input is a real dynamic gesture at an arbitrary view, and the output is synthetic gesture video at a certain view. Moreover, it also returns the probability that a real arbitrary view gesture belongs to which of the fixed-viewpoint gestures.

- The second component contains C3D backbones and an attention unit. C3D backbones take synthetic gestures generated by ArVi-MoCoGAN as inputs and output the feature vectors that correspond to the generated gestures at each viewpoint. These vectors are then multiplied by the probability returned by ArVi-MoCoGAN to form the new feature vectors. These new ones are put into an attention unit to give out the scores of viewpoints that each synthetic gesture belongs to. This approach is novel compared to other methods. In other methods, only one C3D network is used for single-view recognition, but in our work, we proposed several C3D backbones for multi-view gesture recognition. In addition, the integration of an attention unit in the block of gesture recognition is also a new and efficient approach for dynamic and multi-view gesture recognition.

Our proposed solution is evaluated on four datasets including: MICAHandGes [17], IXMAS [18], MuHAVi [19], and NUMA [20]. The experimental results of our proposed method are better than the current solutions, from 1% to 13.58% with arbitrary view gesture recognition and from 1.2% to 7.8% with single view gesture recognition.

The remainder of this paper is organized as follows. In Section II, we briefly survey recent works related to hand gesture recognition approaches. The proposed framework is explained in Section III. The experimental results are analyzed in Section IV. Finally, Section V concludes the paper and states research directions for future work.

II. Related Work

In this section, two brief reviews are presented for (1) dynamic gesture recognition and (2) GAN networks for gesture data augmentation.

A. Dynamic Gesture Recognition

In dynamic gesture recognition, three contexts are considered: gesture recognition at a single view, multiple views, and arbitrary views. In single view dynamic gesture recognition, dynamic gestures for training and testing the classification models are captured by one stationary camera. In [21], authors proposed a C3D architecture to recognize gesture video with input as an image sequence. Spatial features are achieved by 2D CNNs, and temporal features are then obtained by a 3D convolution on the input volume tensor. Resnet50-Temporal Attention network [22] was used for single video recognition. This method used Resnet50 to extract image-level features. Next, a temporal conv layer was applied on these frame-level features to generate temporal attention.

It is different from the single-view approach, the multi-view method considers the gesture images that are captured from multiple cameras at a certain time. In [23], the authors proposed a Mutual-Aid RNN to achieve multi-view action recognition. A view-specific attention pattern was deployed to control other viewpoints as well as discover potential information. This approach leveraged attention information and enhanced multi-view representation learning. [24] used common features to transfer from one view to another with an attention fusion module. A query from one view is matched with the other view by a set of key-value pairs. In the work of [25], the authors presented an extraneous frame scraping technique that employs 2D skeleton features with a Fine-KNN classifier-based HAR (Human action recognition) system.

In arbitrary-view gesture recognition, the model is trained from multiple viewpoints, but a new gesture is recognized from a novel viewpoint. This new gesture's viewpoint differs from a trained viewpoint. The arbitrary gesture recognition could be single-modal or multi-modal. [26] proposed a robust non-linear knowledge transfer model (R-NKTM) for human action recognition from a novel perspective. It transfers knowledge of dynamic gestures from any unknown view to a shared high-level virtual view through finding a non-linear virtual path. R-NKTM only focuses on the temporal features of synthetic models that are fitted to motion data. While the spatial features of a dynamic gesture are lightly taken. [27] proposed Geometric texture Transfer Network (GTNet). A synthetic video is obtained through geometric and appearance features that are extracted from the real viewpoint.

B. GAN-based Gesture Data Generation

Recently, GAN networks have been exploited for dynamic gesture generation. This comes from the growing demand for developing practical applications based on deep learning models. In [13], a conditional GAN-based model named Vi-MoCoGAN is proposed to generate hand gesture videos from multiple viewpoints. Two latent sub-spaces of content and motion are modeled in Vi-MoCoGAN for video synthesizing. In order to control the content and view of the generated gestures, two conditional vectors named content control vector and view control vector are utilized in the model. In addition, the objective function for training the network is also appropriately designed to measure the similarity in content, action, and view of the generated videos and the real ones. In [28] the authors introduced Dynamic Generative Adversarial Network (Dynamic GAN) model to generate photo-realistic videos from skeletal poses. The proposed model is evaluated on three benchmark datasets of RWTH-PHOENIX-Weather 2014T, Indian Sign Language (ISL-CSLTR), and the UCF-101. The quality of the output results are evaluated by the metrics of Similarity Index Measure (SSIM), Inception Score (IS), Peak Signal-to-Noise Ratio (PSNR), and Frechet Inception Distance (FID).

In terms of arbitrary view recognition, some methods utilized GAN models to learn common multi-view space from a training dataset in various viewpoints. Then, these trained GAN models are applied to project data from novel view into common space to detect, segment, or recognize a gesture [16], [27]. In general, GAN-based gesture generation is still challenging, especially in the case of multi and arbitrary viewpoints. The experimental results from the recent methods are promising, but further improvements should be made for high-quality synthetic videos from multi and arbitrary views. This is necessary for data augmentation in training the deep learning models and helps bring gesture recognition research closer to practical applications.

In order to solve these above-mentioned challenges for dynamic and multi-view point gesture recognition, we propose an efficient GAN-based architecture named ArVi-MoCoGAN for generating dynamic gestures at multiple fixed viewpoints from a real dynamic gesture at an arbitrary viewpoint. It is an improved model of Vi-MoCoGAN architecture in [13]. Vi-MoCoGAN generates fixed-viewpoint gestures from the input of noise signals, as do several GAN-based approaches. However, our ArVi-MoCoGAN utilizes real dynamic gestures at arbitrary viewpoints as the inputs. The ArVi-MoCoGAN is integrated with the gesture classifier block of C3D backbones and the attention unit to form a novel and efficient end-to-end system for dynamic and multi-view gesture recognition.

III. Proposed Method

In this section, we introduce an end-to-end framework and present in detail its components, including the ArVi-MoCoGAN architecture, the dynamic gesture recognition block of C3D backbones, and an attention unit.

A. The Overall Framework

The end-to-end framework for arbitrary-view gesture recognition is presented in Figure 1. It consists of two main blocks: (1) a view prediction and transformation block; and (2) a multi-view dynamic gesture recognition block. The first one is implemented by the ArVi-MoCoGAN network with the aim of (i) generating synthetic dynamic gestures ($Z^{syn}_{V_k}$ video) at multiple views by training ArVi-MoCoGAN on the videos that present the gestures at fixed viewpoints ($Z^{r}_{V_k}$ videos); and (ii) returning the view score or the probability that determines if a new generated video of $Z^{syn}_{V_k}$ (a generated dynamic gesture) belongs to which of the fixed-viewpoint gestures. The second block contains C3D backbones and an attention unit. The inputs of C3D networks are $Z^{syn}_{V_k}$ and the outputs are feature vectors $F^{C3D}_{V_k}$. $F^{C3D}_{V_k}$ is then multiplied by the probability P_{V_k} (returned by ArVi-MoCoGAN) to form new feature vectors. These new vectors are passed into the attention unit to give out the viewpoint scores V_k for each synthetic gesture generated by ArVi-MoCoGAN.

B. ArVi-MOCOGAN Architecture

The ArVi-MoCoGAN is proposed to generate fixed-viewpoint gestures from dynamic gestures at arbitrary views. Fixed-viewpoint gestures are captured by stationary cameras and subjects. Each camera captures a frame sequence of a stationary object, and this forms one video from a certain viewpoint. Multiple cameras will create multiple videos from multiple viewpoints. These videos will be used for training ArVi-MoCoGAN. The videos used for testing the ArVi-MoCoGAN model are captured by other fixed cameras and/or moving subjects. This produces multiple videos at arbitrary views. These arbitrary-view gesture videos will be put into the ArVi-MoCoGAN model to give out two outputs: (1) synthetic arbitrary-view gesture videos; and (2) the probability that an arbitrary-view gesture belongs to the fixed-viewpoint gesture.

The details of the proposed ArVi-MoCoGAN framework are illustrated in Figure 2. It consists of two main parts: the generator networks and the discriminator networks.

1) Generator networks: ArVi-MoCoGAN contains two generator networks of G_1 and G_2. G_1 tries to learn and creates a synthetic content image I^{syn}_{Vk}. The inputs of generator G_1 are four vectors:

- Z^*_M: is the hypothetical motion vector which is indicated in Eq. (1). Z^*_M is randomly chosen from 16 vectors of $\left[Z^{(*0)}_M, .., Z^{(*15)}_M\right]$ $\left(Z^*_M \in \left[Z^{(*0)}_M, .., Z^{(*15)}_M\right]\right)$. These 16 vectors are generated by putting the a frame into the encoder network E_1 and RNN network. This input frame is the first image/frame ($I^r_{V_j} = I^0_{V_j}$) in a video $Z^r_{V_j}$ ($Z^r_{V_j} = [I^{(0)}_{V_j}, ..., I^{(15)}_{V_j}]$). Z^*_M helps to control the information about the object's motion that needs to be presented in the expected outputs of the generator G1.

- Z_C: is the content vector which is the output of the encoder E_2 with the input is the first frame $I^{(0)}_{V_j}$. Z_C is intended to control the content of the videos generated by G_1 and G_2.

- Z_{V_k}: this vector is used to control the number of viewpoints of the generated images or videos from generators G_1 and G_2. In other words, how many viewpoints are generated depends on the number of viewpoints in the database used for training the model.

- $Z_{Subject}$: this vector plays the role of a conditional vector in conditional GAN models like Z_{V_k}. However, it controls the subject of the dynamic gesture.

$$Z^*_M = f^{RNN}(E_1(Z^{(r)})) = f^{RNN}(E_1(I^{(r)}(0), ..., (E_1(I^{(r)}(15)) \tag{1}$$

Generator G_2 tries to generate synthetic gestures in multiple fixed views from a real dynamic gesture in other view. In our consideration, the synthetic image sequence contains 16 frames $Z^{syn}_{V_k}$ ($Z^{syn}_{V_k} = [I^{(0)}_{V_k}, .., I^{(15)}_{V_k}]$). The inputs of generator G_2 are Z_C, Z_{V_k}, $Z_{Subject}$, and Z_M, in which, Z_M is the output result when we put a frame sequence (a real video $Z^r_{V_j} = [I^{(0)}_{V_j}, .., I^{(15)}_{V_j}]$) into encoder E_1 and RNN network. Z_M is calculated as in Eq. (2), with $\mathcal{N}(\boldsymbol{z}|0, I_z)$ is a noise vector and Z^{Random}_{Class} is a random category vector.

$$Z_M = f^{RNN}(\mathcal{N}(\boldsymbol{z}|0, I_z), Z^{Random}_{Class}) \tag{2}$$

A dynamic gesture is a frame sequence that contains both content and motion cues. Therefore, a gesture can be decomposed into two latent sub-spaces of content and motion. In the first sub-space, the content of gesture is mainly characterized by encoder E_2. The output of encoder E_1 and a RNN network are converted into Z^*_M as illustrated in top part of Figure 2. It is note that the inputs of generator G_1 consists of Z^*_M, Z_C, Z_{V_k}, and $Z_{Subject}$. Its output is a synthetic image $I^{syn}_{V_k}$ that presents the content of object. While the inputs of generator G_2 contains Z_M, Z_C and Z_{V_k}, the output is a synthetic video $Z^{syn}_{V_k}$ which presents the movement of a gesture. Both generators are sequentially trained. Their parameters are updated from generator G_1 to generator G_2 and vice versa.

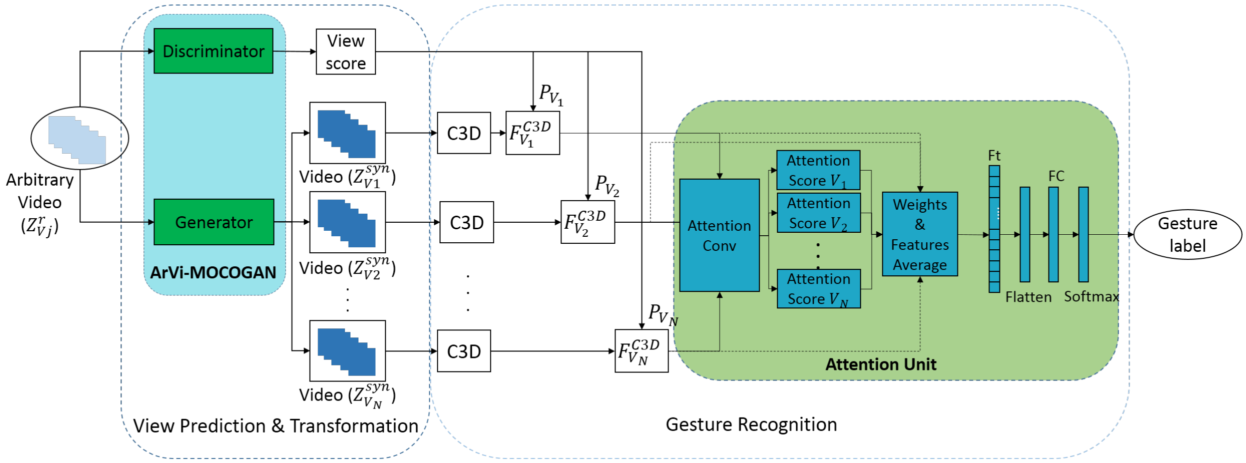

Fig. 1. The proposed end-to-end framework of ArVi-MOCOGAN and C3D backbones with attention unit for dynamic and arbitrary-view gesture recognition.

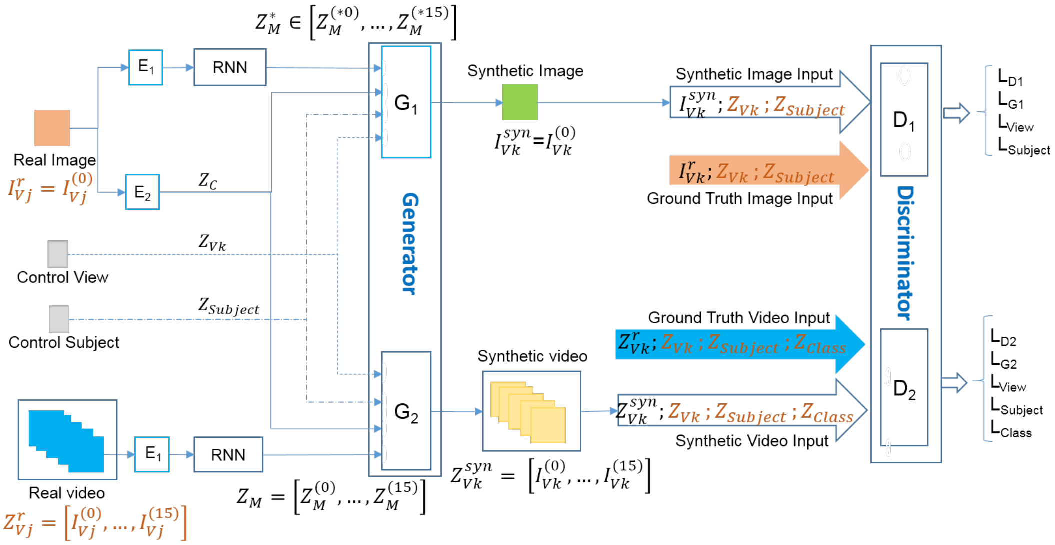
$Z^*_M \in \left[Z^{(*0)}_M, \dots, Z^{(*15)}_M\right]$
E1
RNN
G1
Synthetic Image
$I^{syn}_{Vk} = I^{(0)}_{Vk}$
Synthetic Image Input
$I^{syn}_{Vk}; Z_{Vk}; Z_{Subject}$
D1
L_{D1}
L_{G1}
L_{View}
$L_{Subject}$
Real Image
$I^r_{Vj} = I^{(0)}_{Vj}$
E2
Z_C
$I^r_{Vk}; Z_{Vk}; Z_{Subject}$
Ground Truth Image Input
Generator
Discriminator
Z_{Vk}
Control View
$Z_{Subject}$
Control Subject
Ground Truth Video Input
$Z^r_{Vk}; Z_{Vk}; Z_{Subject}; Z_{Class}$
Synthetic video
G2
D2
L_{D2}
L_{G2}
L_{View}
$L_{Subject}$
L_{Class}
$Z^{syn}_{Vk}; Z_{Vk}; Z_{Subject}; Z_{Class}$
Synthetic Video Input
Real video
$Z^r_{Vj} = \left[I^{(0)}_{Vj}, \dots, I^{(15)}_{Vj}\right]$
E1
RNN
$Z_M = \left[Z^{(0)}_M, \dots, Z^{(15)}_M\right]$
$Z^{syn}_{Vk} = \left[I^{(0)}_{Vk}, \dots, I^{(15)}_{Vk}\right]$

Fig. 2. The proposed ArVi-MoCoGAN architecture with two generators of G_1, G_2 for generating synthetic gesture images and videos and two discriminators of D_1, D_2 for distinguishing the real and synthetic samples.

2) Discriminator networks: Discriminator D_1 network tries to distinguish a real content image $I^r_{V_k}$ with a synthetic content image $I^{syn}_{V_k}$ ($I^{syn}_{V_k}$ is the output of discriminator G_1). Discriminator D_2 network distinguishes a real dynamic gesture $Z^r_{V_k}$ from a generated one $Z^{syn}_{V_k}$ ($Z^{syn}_{V_k}$ is the output of generator G_2).

The optimal function for the Generator G_1 and Discriminator D_1 is indicated in Eq. (3):

$$\begin{aligned}\max_{G_1,R_M}\min_{D_1}\mathcal{F}_1 = \max_{G_1,R_M}\min_{D_I}(\mathcal{F}_{mcg1}(D_1,G_1,R_M)+\\ \lambda L_{Image}(G_1,P_{Image})+\beta L_{View}(G_1,P_{View})+\\ \gamma L_{Subject}(G_1,P_{Subject}))\end{aligned} \tag{3}$$

For the Generator G_2 and Discriminator D_2, the optimal function is presented in Eq. (4):

$$\begin{aligned}\max_{G_2,R_M}\min_{D_2}\mathcal{F}_2 = \max_{G_2 R_M}\min_{D_2}(\mathcal{F}_{mcg2}(D_2,G_2,R_M)+\\ \lambda L_{Video}(G_2,P_{Video})+\beta L_{View}(G_2,P_{View})+\\ \gamma L_{Subject}(G_2,P_{Subject})+\alpha L_{Class}(D_2,P_{Class}))\end{aligned} \tag{4}$$

The optimal function for the ArVi-MoCoGAN model is indicated in Eq. (5):

$$\max_{G_1,G_2,R_M}\min_{D1,D2} F_{avmcg} = \max_{G_1,G_2 R_M}\min_{D1,D2}(\mathcal{F}_1+\mathcal{F}_2) \tag{5}$$

Where λ, β, α, γ are hyper-parameters. In this work, they are chosen by 1. P_{Image}, P_{Class}, $P_{Subject}$, and P_{View} are distribution approximations of the variables of gesture content, gesture category, subject and view that control video generation. P_{Class} element is added at the last feature layer of D_2 network, P_{Image}, $P_{Subject}$, P_{View} are components that adjoined in both Generators and Discriminators of ArVi-MoCoGAN network.

The trained ArVi-MoCoGAN model is then be used to generate synthetic dynamic gestures. The inputs of the trained ArVi-MoCoGAN model consist of a real dynamic gesture Z^r ($Z^r = Z_{(r,Video)}$), the control viewpoint $Z_{View} = Z_{V_k}^{Random}$ of G_2. The outputs are the synthetic dynamic gestures at arbitrary views ($Z^{syn} = Z_{(syn,Video)}$) gained from G_2 and the probability distribution P_{View} gotten from D_2. P_{View} shows the probability that an generated arbitrary-view dynamic gesture belongs to which of the fixed-viewpoint gestures. It is then utilized to classify a dynamic gesture from an unknown viewpoint, as presented in detail in the next section.

C. C3D Backbones and Attention Unit

In this work, two implementation scenarios for dynamic gesture recognition from arbitrary viewpoints are implemented, called ArViAU (Arbitrary view gesture recognition with Attention unit) and ArViAVR (Arbitrary View gesture recognition with Average method). ArViAU contains C3D backbones and an attention unit, but ArViAVR includes C3D backbones only.

1) Arbitrary view gesture recognition with Attention Unit (ArViAU): In this work, C3D models [29] are applied as backbones with transfer learning by dynamic gesture databases in N views. The parameters of the C3D models are independently retrained and updated by dynamic gesture databases on each view. The retrained C3D models are used as the 3D feature extractors for gesture-level features. The outputs of C3D extractors are taken from the FC6 layer with feature vectors $F_{vk}(M \times 1) \mid k = (1,..,N)$, M=4096 as presented in Eq. (6):

$$F_{V_k}^{C3D}(M \times 1) = \begin{bmatrix} F_{V_k}^{(1)} \\ F_{V_k}^{(2)} \\ ... \\ F_{V_k}^{(M)} \end{bmatrix} \tag{6}$$

Next, both feature vector F_{V_k} and probability distribution of view scores P_{V_k} are combined on each viewpoint as presented in Eq. (7):

$$F_{(V_k,P_{V_k})} = P_{V_k} F_{V_k}^{C3D} = \begin{bmatrix} F_{(V_k,P_{V_k})}^{(1)} \\ F_{(V_k,P_{V_k})}^{(2)} \\ ... \\ F_{(V_k,P_{V_k})}^{(M)} \end{bmatrix} = \begin{bmatrix} F_{V_k}^{(1)} P_{V_k} \\ F_{V_k}^{(2)} P_{V_k} \\ ... \\ F_{V_k}^{(M)} P_{V_k} \end{bmatrix} \tag{7}$$

All features of multiple viewpoints are normalized following the minimum and maximum values of all feature vectors on entire viewpoints. ($F_{min} = min(F_{(V_1,P_{V_1})}$,...,$F_{(V_N,P_{V_N})}$), and $F_{max} = max(F_{(V_1,P_{V_1})}$,...,$F_{(V_N,P_{V_N})}$)). The normalized vector is presented by $F_{(V_k,P_{V_k})}^{norm}$ as Eq. (8):

$$\begin{aligned} F^{norm} &= [F_{(V_1,P_{V_1})}^{norm}, ..., F_{(V_N,P_{V_N})}^{norm}] \\ &= [\frac{F_{(V_1,P_{V_1})} - F_{min}}{F_{max} - F_{min}}, ..., \frac{F_{(V_N,P_{V_N})} - F_{min}}{F_{max} - F_{min}}] \end{aligned} \tag{8}$$

All normalized vectors $F_{(V_k,P_{V_k})}^{norm} \mid k = (1,..,N)$ from C3D backbones are then put into an attention layer of (N × M × 1) to output attention scores $a_k \mid k = (1,...,N)$.

The attention scores a_k are calculated by $Sigmoid$ function and L_1 normalization function [30] as presented in Eq. (9):

$$a_k = \frac{\sigma^{x_k}}{\sum_{k=1}^{N} \sigma^{x_k}} = \frac{\frac{1}{1-e^{x_k}}}{\sum_{k=1}^{N} \frac{1}{1-e^{x_k}}} \tag{9}$$

The Attention Conv trains and generates attention factors according to the roles of synthetic features at N views. It presents the effects of feature vectors through attention scores. The attention weights are applied for all gesture features to obtain a feature vector of $F_t(1 \times 2048)$. The aggregated feature is built based on N single synthetic features (F_{V_k}) and efficient scores (P_{V_k}) that is presented in Eq. (10):

$$F_t = \frac{1}{N} \sum_{k=1}^{N} (a_k F_{(V_k,P_{V_k})}^{norm}) \tag{10}$$

In this work, the lost function of C3D models is exploited for entire viewpoints. In addition, the softmax cross-entropy loss function is also utilized to train the attention networks and classify dynamic gestures. Given a predicted result of dynamic gesture $\bar{p}_i$ with the ground truth is p_i, the loss function is calculated as in Eq. (11):

$$L_{softmax} = \frac{1}{K} \sum_{i=1}^{K} p_i log \bar{p}_i \tag{11}$$

2) Arbitrary View gesture recognition with average method (ArViAvr): This method combines the probability distributions of a view (P_v) and a gesture from FC6 layers of C3D models ($P_G^{V_k}(1 \times C)$, C is the number of gesture classes). The recognition accuracy of a real gesture is finally computed from all multi-view synthetic dynamic gestures as presented in Eq. (12):

$$\text{Acc = Argmax}(\frac{\sum_{k=1}^{N} P_{V_k} P_{G_1}^{V_k}}{N} \text{,...,} \frac{\sum_{k=1}^{N} P_{V_k} P_{G_C}^{V_k}}{N}) \tag{12}$$

IV. Experiment and Result

This section describes in detail the datasets used for the experiments, and two evaluation protocols are set for the experimental datasets: the single-view protocol and the arbitrary-view protocol. In addition, we also mention the metrics that are used for evaluating the quality of the synthetic samples generated by ArVi-MoCoGAN compared to the original ones. The enhanced experiments and the results of the proposed method for dynamic gesture recognition are also presented and discussed in this section.

A. Dataset and Evaluation Protocols

1) Dataset: In this study, four multi-view and dynamic gesture datasets are utilized for evaluating the proposed framework: the MICAGes dataset [31], three benchmark datasets of IXMAS [18], MuHAVi [19], and NUMA [20]. These datasets contain the gestures that are synchronously captured by multiple cameras (N cameras), a variety of subjects (S subjects), and categories of dynamic gestures (C classes) as presented in Table I:

TABLE I. THE FOUR MULTI-VIEW AND DYNAMIC GESTURE DATASETS OF MICAGES, IXMAS, MUHAVI, AND NUMA

	MICAGes	IXMAS	MuHAVi	NUMA
Camera (N)	05	04	07	03
Class (C)	09	12	07	10
Subject (S)	10	04	07	10
Video	1500	1584	3038	1475

We employ the *"Leave-one-subject-out-cross-validation"* strategy to split data in the training and testing phases. A multi-view database D^r has S subjects (l=(1,...,S)), N views (k=(1,...,N)), therefore S experiments are holdout. Considering a test subject l=s^{th}, a real dataset is divided into two parts as Eq. (13):

$$D^r = D_1^r \cup D_2^r = \begin{cases} D_1^r = \{D_{V_k}^l \mid k = (1, ..., N), l = s^{th}\} \\ D_2^r = \{D_{V_k}^l \mid k = (1, ..., N); l = (1, ..., S); l \neq s^{th}\} \end{cases} \quad (13)$$

Where D_1^r contains the dynamic gestures of the s^{th} subject at the entire N views, D_2^r are the remaining subjects at all N views.

2) Evaluation protocols: In this work, the evaluation protocols are set for experimental datasets used in (i) training the ArVi-MoCoGAN network and generating the dynamic gestures; (ii) training and testing the gesture classifiers. They are single-view protocol and arbitrary-view protocol.

- Single view protocol: we use *"Leave-one-subject-out-cross-validation"* strategy in all evaluations. Thus, the data for training ArVi-MoCoGAN and generating the synthetic gestures is separated as follows:

- Training of ArVi-MoCoGAN in single view evaluation: All dynamic gestures in D_2^r dataset are utilized as input for training ArVi-MoCoGAN model ($D_{ArVi-MOCOGAN}/D_{ArVi}$) as Eq. (14):

$$D_{ArVi}^{Tr} = D_2^r \quad (14)$$

- Data generating of ArVi-MoCoGAN in single view evaluation: Having N viewpoints means N experiments are conducted. For $k = j^{th}$ view evaluation, input gestures are taken from D_1^r, except for the data from j^{th} view. It means that the data on the other views is projected on the j^{th} view for data enrichment. The inputs of the retrained ArVi-MoCoGAN model are dynamic gestures of D_1^r on other viewpoints as Eq. (15):

$$D_{ArVi}^{Te} = \{D_1^r | k = (1, .., N); k \neq j^{th}\} \quad (15)$$

Synthetic data are output of ArVi-MoCoGAN model that is presented as Eq. (16):

$$D_{ArVi}^{Out} = \{D_{1,V_k j^{th}}^{Syn} | k = (1, ..., N); k \neq j^{th}\} \quad (16)$$

In this evaluation protocol, C3D networks are applied to recognize dynamic gestures, which are fine-tuned by the training data D_{C3D}^{Tr} (Eq. (17)). The testing data D_{C3D}^{Te} is then applied as Eq. (18):

- Training of C3D in single view evaluation:

$$D_{C3D}^{Tr} = \{D_2^r | k = j^{th}\} \cup D_{ArVi}^{Out} \quad (17)$$

- Testing of C3D in single view evaluation:

$$D_{C3D}^{Te} = \{D_1^r | k = j^{th}\} \quad (18)$$

- Arbitrary view protocol:

In this evaluation protocol, one view j^{th} is considered an unknown view, and the remaining views are observed as the fixed views. This work also composes two stages as follows:

In the first stage, because j^{th} view is consider an arbitrary viewpoint. Thus, only a part of the D_2^r dataset is used to train ArVi-MoCoGAN model is presented in Eq. (19). This work aims to create a common space from multiple fixed viewpoints. It means that data of j^{th} view (an arbitrary view) do not attend in creating common space with ArVi-MoCoGAN model:

$$D_{ArVi}^{Tr} = \{D_2^r | k = (1, ..., N); k \neq j^{th}\} \quad (19)$$

In the second stage, the ArVi-MoCoGAN model is used in two roles: (1) data augmentation for the gesture classifier; and (2) the ArVi-MoCoGAN model becomes an intermediate step for dynamic gesture recognition. In the role of data augmentation, gestures of D_1^r, except j^{th} subject are used as the inputs for the trained ArVi-MoCoGan model (Eq. (20)) to generate synthetic data D_{ArVi}^{Out1} (Eq. (21)). This synthetic data is then used as data augmentation for training the C3D model D_{C3D}^{Te} (Eq. (24)). The input and output data of the ArVi-MoCoGan model in the first role, as follows:

- The input of ArVi-MoCoGan in role (1):

$$D_{ArVi}^{Te1} = \{D_1^r | k = (1, .., N); k \neq j^{th}\} \quad (20)$$

- The output of ArVi-MoCoGan in role (1):

$$D_{ArVi}^{Out1} = \{D_{1,V_{k_1} V_{k_2}}^{Syn} | k_1 = (1, ..., N); k_2 = (1, ..., N), k_1, k_2 \neq j^{th}\} \quad (21)$$

In the second role of ArVi-MoCoGan, an arbitrary view gesture of j^{th} view in D_1^r is projected into a common space with the previously trained ArVi-MoCoGAN, whose input is D_{ArVi}^{Te2} (Eq. (22)), and output D_{ArVi}^{Out2} (Eq. (23)) contains synthetic gestures in a common space of fixed multiple views. D_{ArVi}^{Out2} is utilized to recognize gesture in Eq. (25). The input and output data of the ArVi-MoCoGan model in the second role are as below:

- The input of ArVi-MoCoGan in role (2):

$$D_{ArVi}^{Te2} = \{D_1^r | k = j^{th}\} \quad (22)$$

- The output of ArVi-MoCoGan in role (2):

$$D_{ArVi}^{Out2} = \{D_{1,j^{th} V_k}^{Syn} | k = (1, ..., N); k \neq j^{th}\} \quad (23)$$

In the arbitrary view evaluation protocol, C3D networks and an attention unit ($C3D - AU$) are applied to recognize synthetic dynamic gestures in the fixed multiple viewpoints as presented in Sec. III-C. This model is fine-tuned by the training data D_{C3D-AU}^{Tr} (Eq. (24)), and the testing data D_{C3D-AU}^{Te} is then applied as Eq. (25).

- Training data of $C3D - AU$ in role (2):

$$D^{Tr}_{C3D-AU} = \{D^r_2 | k = (1, ..., N); k \neq j^{th}\} \cup D^{Out1}_{ArVi} \quad (24)$$

- Testing data of $C3D - AU$ in role (2):

$$D^{Te}_{C3D-AU} = D^{Out2}_{ArVi} \quad (25)$$

Throughout the whole system, an arbitrary view dynamic gesture $D^r_{1,j^{th}}$ is firstly projected into a multi-view common space (ArVi-MoCoGAN network) to obtain the synthetic gestures $D^{Syn}_{1,j^{th}k}$. These synthetic gestures are classified on certain fixed multiple viewpoints. Finally, the dynamic gesture scores are computed by two strategies (ArViAU and ArViAVR) as presented in the previous sections (Sec. III-A and Sec.III-C). For each evaluation holdout, the computed accuracy metric is determined by all the accuracy scores of the synthetic gestures on the target views.

B. Model Configurations

Encoder 1 (E_1) and Encoder 2 (E_2) networks are applied by five Conv2d with layer sizes of [512, 256, 128, 64]. Generator 1 (G_1) and Generator 2 (G_2) networks utilize five ConvTrans2d layer which its sizes of [64, 128, 256, 512, 512], Kernel (4,4), Stride 2,2), Padding (1,1), BN2d and ReLU functions.

Discriminator 1 (D_1) uses six Conv2d with sizes of [512, 512, 256, 128, 128, 64]. Kernel sizes (4,4), S(2,2), Padding size (1,1), BN2d and LeakyReLU functions. Discriminator 2 (D_2) utilizes six Conv3d with sizes of [512, 512, 256, 128, 128, 64], Kernel (4,4,4), Stride (1,2,2), Padding (0,1,1), BN3d and LeakyReLU functions.

C. Evaluation Metrics

In this work, the quality of the synthetic videos generated by ArVi-MoCoGAN is evaluated based on two criteria: (1) the similarity between the videos generated by ArVi-MoCoGAN and the real ones; and (2) the performance of the dynamic gesture recognition when training the classifier on the augmented data compared to training only on the original data. The first criterion is evaluated based on the FVD score [32]. In addition, two other metrics, Structural Similarity (SSIM) [33] and Peak Signal-to-Noise Ratio (PSNR) [34] are also used to evaluate the quality of synthetic videos in comparison with the real ones. The higher values of SSIM or PSRN indicate better quality of synthetic gestures.

Before evaluating the quality of the synthetic videos generated by the ArVi-MoCoGAN model by the two above criteria, we evaluate (i) the performance of the ArVi-MoCoGAN training and (ii) the saturation in the amount of synthetic videos from the ArVi-MoCoGAN model on dynamic gesture recognition accuracy. The first one is shown by the loss values of discriminators D_1 and D_2 in the ArVi-MoCoGAN architecture. For the second one, C3DVS score (C3D Video Score) [32] is used for evaluation. Based on these, the optimal values are selected for later evaluations.

D. Experimental Evaluation

1) Evaluation of the saturation of synthetic videos by the ArVi-MoCoGAN model on dynamic gesture recognition accuracy: Figure 3 shows the C3DVS scores on various published datasets from no augmentation with zero generator (original dataset) to eleven synthetic videos (combination of original dataset and synthetic dataset). We use the ”Single-view protocol” in this experiment. It is apparent that data augmentation by the ArVi-MoCoGAN network dramatically improves dynamic gesture recognition compared to the evaluation on the original dataset. In addition, it also indicates the number of synthetic videos that should be used to improve the accuracy of gesture recognition. It can be seen from Figure 3 that for the IXMAS dataset, convergence occurs after 5 generator samples. MICAGes and NUMA datasets obtain convergence after 4 samples. MuHAVi dataset is stable at all. These sample numbers will be applied to the FVD score as well as the remaining evaluations.

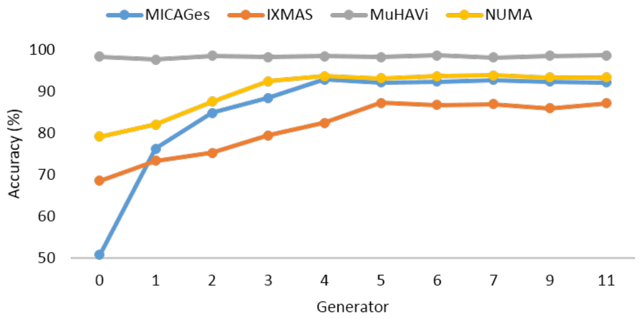

Fig. 3. C3DVS scores of ArVi-MoCoGAN network on different datasets.

2) Evaluation the similarity between synthetic videos and the real ones: The optimal synthetic samples calculated by the C3DVS score (Figure 3) are 03 synthetic samples for the MuHAVi dataset, 04 synthetic samples for each of the NUMA dataset and the MICAGes dataset, and 05 synthetic samples for the IXMAS dataset. In this work, we apply the FVD metric to generated data at various epochs of the retrained ArVi-MoCoGAN model.

The results in Figure 4 show that our arbitrary-view ArVi-MoCoGAN is dramatically reduced from 1400 FVD at epoch 10^{th} to around 600 FVD at epoch 200^{th} and converged after 200 epochs with MICAGes dataset (blue color line). FVD values of the IXMAS, MuHAVi, and NUMA datasets are presented in the green, violet, and orange color lines, respectively. It can be seen from Figure 4 that the worst results at all epochs belong to the MuHAVi dataset. Its FVD values are the lowest among the four datasets. It is clear that FVD values are stable after 350 epochs for all experimental datasets. As a result, we will use synthetic data at 350 epochs for the remaining evaluations.

The experimental results in Table II show the comparative results of the ArVi-MoCoGAN model with the image sequences generated by Vi-MoCoGAN at poch 350^{th} epoch. One dynamic action on a certain viewpoint is considered as an input of the 350^{th} model in order to generate six dynamic gestures on each remaining view. It can be seen from the

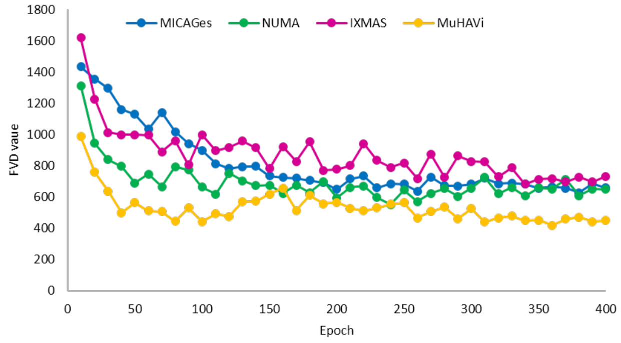

Fig. 4. The FVD values of data distribution between real videos and synthetic videos at various epochs of the ArVi-MoCoGAN network.

Table II that our framework outperforms Vi-MoCoGAN at all metrics as well as the datasets, with SSIM and PNRS values drastically higher and FVD values dramatically smaller than Vi-MoCoGAN.

TABLE II. SSIM, PSNR, and FVD Scores of ArVi-MoCoGAN and Vi-MoCoGAN on Various Datasets

Dataset	Model	$SSIM(\uparrow)$	$PNRS(\uparrow)$	$FVD(\downarrow)$
MICAGes	Vi-MoCoGAN	0.77	27.69	936
	ArVi-MoCoGAN	**0.86**	**30.65**	**629**
IXMAS	Vi-MoCoGAN	0.79	26.21	969
	ArVi-MoCoGAN	**0.87**	**33.21**	**719**
MuHAVi	Vi-MoCoGAN	0.68	25.59	791
	ArVi-MoCoGAN	**0.72**	**30.26**	**420**
NUMA	Vi-MoCoGAN	0.65	23.19	873
	ArVi-MoCoGAN	**0.76**	**32.01**	**606**

Figures 5-a, b, c illustrate the synthetic key frames of G_6 gesture of three different subjects in the MICAGes dataset at 350 epochs, respectively. The rows at the top of the figure are the generated videos by the Vi-MoCoGAN model, and at the bottom are the synthetic videos of the ArVi-MoCoGAN model. It can be seen that Vi-MoCoGAN generates videos with the wrong category and poor quality. The beginning frames of the Vi-MoCoGAN frame sequence show the hand gestures are far from the body. This is the opposite of ArVi-MoCoGAN, with the outputs having clearer frames and showing the truth category.

3) Evaluation the performance of dynamic gesture recognition using data augmentation by ArVi-MoCoGAN: The efficiency of (ArVi-MoCoGAN+C3D) is investigated on four other benchmark datasets of MICAGes, IXMAS [18], MuHAVi [19], and NUMA [20] as illustrated in the Figure 6. The results in this figure indicate that using augmentation data for dynamic gesture recognition outperforms the case of original data, with 68,87% (C3D) and 87.25% (ArVi-MoCoGAN+C3D) on IXMAS; 86.83% (C3D) and 94.51% (ArVi-MoCoGAN) on NUMA. However, the results on the MuHAVi dataset are nearly the same for (ArVi-MoCoGAN+C3D) and (C3D), with 98.36% and 98.27%, respectively.

The performance of (ArVi-MoCoGAN+C3D) is also compared to some SOTA methods, as presented in Table III. It is clear that our data augmentation solution significantly improves single-view gesture recognition accuracy in comparison with other solutions. The experiments on the IXMAS dataset show the highest recognition accuracy is 87,2% for ArVi-MoCoGAN+C3D), while the results for 3D Exemplars, SSM, WLE, $(D_{R18}+ELM)$, $(D_{R18}+ELM+aug)$, (D_A+ELM), $(D_A + ELM + aug)$ are 63,2%, 72,5%, 79,9%, 67.6%, 72,7%, 73,1%, and 79.4%, respectively. For MICAGes, (ArVi-MoCoGAN+C3D) solution is compared to other methods of Multi-Br TSN [35], Multi-Br TSN-GRU [35] and R34 (2+1)D With CVA [24]. The result of (ArVi-MoCoGAN+C3D) is 92.88% which is higher than R34 (2+1)D With CVA (91.71%), Multi-Br TSN - GRU (88.71%), and Multi-Br TSN (81.77%). The result on the MuHAVI dataset gained by (ArVi-MoCoGAN+C3D) is the highest, with 98.27% compared to the 93.6% of $(D_A+ELM+aug)$, 93.4% of $(D_{R18}+ELM+aug)$, 92.1% of $(D_R18+ELM)$, and 91.1% of (D_A+ELM). The experimental results on NUMA dataset show that the highest accuracy belongs to (ArVi-MoCoGAN+C3D) with 94.51%, the next ones are 93.81% of Multi-Br TSN - GRU, 92.78% of R34 (2+1)D With CVA, 92.1% of DA-Net[36], 90.3% of TSN [37], and 88.49% of Multi-Br TSN. The worst case happens to SAM[38] with 83.2%.

TABLE III. Comparison of Cross-subject Recognition Accuracy (%) of Single-view Dynamic Gesture Methods on Various Datasets ("aug" Symbol Means Data Augmentation)

	IXMAS	MICAGes	MuHAVI	NUMA
3D Exemplars[39]	63.2	-	-	-
SSM [40]	72.5	-	-	-
WLE [41]	79.9	-	-	-
SAM[38]	-	-	-	83.2
TSN [37]	-	-	-	90.3
DA-Net[36]	-	-	-	92.1
Multi-Br TSN [35]	-	81.77	-	88.49
Multi-Br TSN - GRU [35]	-	88.71	-	93.81
R34 (2+1)D With CVA [24]	-	91.71	-	92.78
$D_{R18} + ELM$ [42]	67.6	-	92.1	-
$D_{R18} + ELM + aug$ [42]	72.7	-	93.4	-
$D_A + ELM$ [42]	73.1	-	91.1	-
$D_A + ELM + aug$ [42]	79.4	-	93.6	-
ArVi-MoCoGAN + C3D	**87.25**	**92.88**	**98.27**	**94.51**

The experiments for two end-to-end methods, ArViAvr and ArViAU (presented in Sec. III-C) are implemented on four published multi-view datasets of MICAGes, IXMAS, MuHAVi and NUMA, as illustrated in the IV. In this work, we train the end-to-end CNN models with two strategies: (1) Training on the remaining (N-1) viewpoints and testing on one viewpoint; (2) Training on the main frontal viewpoints and testing on one non-frontal viewpoint.

It can be seen from the Table IV that the ArViAU method obtains the best accuracy on the benchmark datasets of MICAGes (94.03%), IXMAS (86.75%), MuHAVi (95.35%), and NUMA (93.19%). In addition, these results outperform the SOTA methods of $D_A + ELM + aug$ [42], Shah et al. [45]. For IXMAS dataset, our proposed method with the case of ArViAVR ((ArVi-MoCoGAN+C3D); AVR) has the accuracy of 82.03%. This is a little lower than $D_A + ELM + aug$ method in [42]. However, with data augmentation ((ArVi-MoCoGAN+C3D);AU), our method is about 3% higher than $(D_A+ELM+aug)$ method. For MuHAVI dataset, our ((ArVi-MoCoGAN+C3D);AU) is much better than $(D_A + ELM + aug)$ method, with 13.58% higher in recognition accuracy. In comparison with the solution of Shah et al. [45], our ((ArVi-MoCoGAN+C3D);AU) is increased by 1.5%.

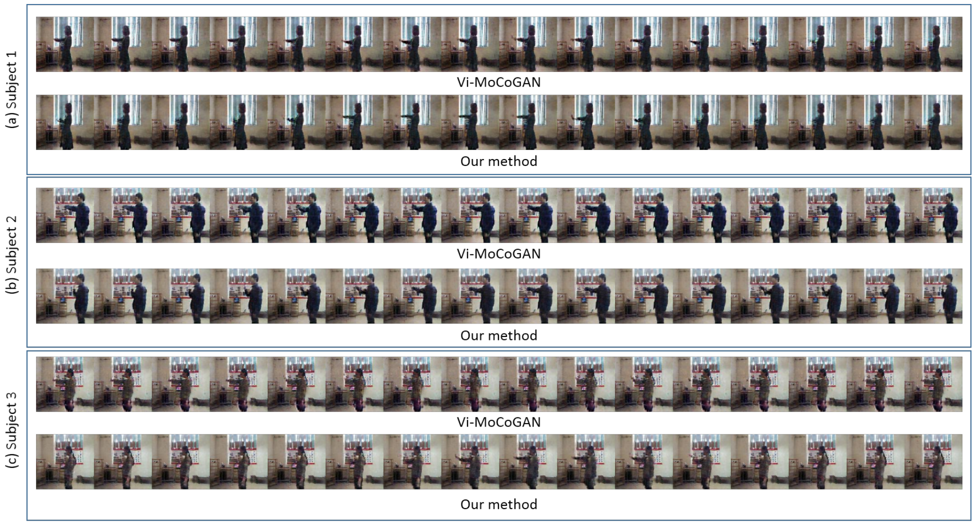

Fig. 5. Synthetic dynamic gestures of ArVi-MoCoGAN (our method) and Vi-MoCoGAN with the MICAGes dataset.

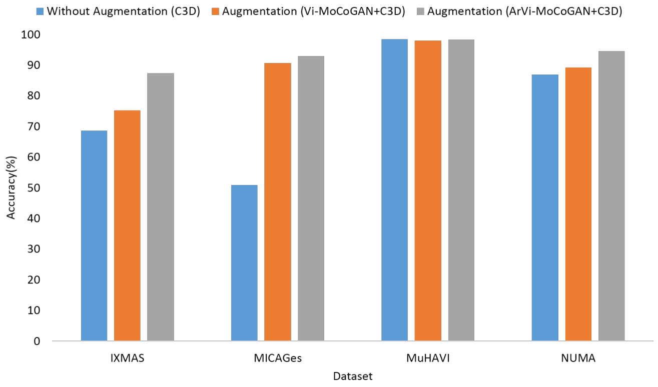

Fig. 6. Single view dynamic gesture recognition of different methods C3D, (Vi-MoCoGAN+C3D), and (ArVi-MoCoGAN+C3D) on various datasets.

V. Conclusion and Future Work

This work proposes a novel end-to-end framework based on GAN architecture and attention units for dynamic and arbitrary-view gesture recognition. Several enhanced experiments on the standard datasets of dynamic gestures are implemented to show the better results of our proposal compared to other solutions. The experimental results are remarkable and promising. However, they are only tested on experimental databases and have not been evaluated in real-world conditions. In order to be able to adapt to practical applications in future work, multi-modal elements such as RGB, depth, skeleton, etc. can be considered in the proposed system. In addition, to improve the quality of data augmentation, condition vectors can be added to the proposed GAN model to control the desired outputs of arbitrary-view dynamic gestures. The transformer-based architectures can also be deployed to improve both gesture data generation and recognition.

TABLE IV. The Comparative Results of our Proposed Method with other Solutions in Dynamic Gesture Recognition Accuracy (%) for Arbitrary View Testing

	IXMAS	MICAGes	MuHAVI	NUMA
3D Exemplars[39]	81.3	-	-	-
ST+Spin-Image features[43]	71.7	-	-	-
SSM [40]	72.7	-	-	-
SAM[38]	-	-	-	77.2
R-NKTM [26]	74.1	-	-	-
WLE [41]	82.8	-	-	-
TSN [37]	-	-	-	80.6
DA-Net[36]	-	-	-	84.2
Multi-Br TSN - GRU [35]	-	88.71	-	84.4
Glimpse Clouds [44]	-	-	-	87.6
R34 (2+1)D With CVA [24]	-	91.71	-	92.74
D_A+ELM[42]	79.3	-	77.78	-
D_A+ELM+aug[42]	83.8	-	81.76	-
Shah et al.[45]	-	-	-	91.7
ArViAVR (ArVi-MoCoGAN+C3D; AVR)	82.03	90.87	94.04	85.51
ArViAU (ArVi-MoCoGAN+C3D; AU)	**86.75**	**94.03**	**95.35**	**93.19**

Acknowledgment

This document is the results of the research project funded by Viet Nam government under grant number KC-4.0.28/19-25.

References

[1] R. Zhao, Y. Wang, P. Jia, C. Li, Y. Ma, and Z. Zhang, "Review of human gesture recognition based on computer vision technology," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, 2021, pp. 1599–1603.

[2] H. Zhao, M. Cheng, J. Huang, M. Li, H. Cheng, K. Tian, and H. Yu, "A virtual surgical prototype system based on gesture recognition for virtual surgical training in maxillofacial surgery," *International Journal of Computer Assisted Radiology and Surgery*, pp. 1–11, 2022.

[3] G. Plouffe and A.-M. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE transactions on instrumentation and measurement*, vol. 65, no. 2, pp. 305–316, 2015.

[4] M. Haid, B. Budaker, M. Geiger, D. Husfeldt, M. Hartmann, and N. Berezowski, "Inertial-based gesture recognition for artificial intelligent cockpit control using hidden markov models," in *2019 IEEE*

International Conference on Consumer Electronics (ICCE). IEEE, 2019, pp. 1–4.

[5] J. Yu, M. Qin, and S. Zhou, "Dynamic gesture recognition based on 2d convolutional neural network and feature fusion," *Scientific Reports*, vol. 12, no. 1, p. 4345, 2022.

[6] Y. Liu, D. Jiang, H. Duan, Y. Sun, G. Li, B. Tao, J. Yun, Y. Liu, and B. Chen, "Dynamic gesture recognition algorithm based on 3d convolutional neural network," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[7] G. Fronteddu, S. Porcu, A. Floris, and L. Atzori, "Dataset for dynamic hand gesture recognition systems," 2021. [Online]. Available: https://dx.doi.org/10.21227/43mn-bb52

[8] R. Jain, R. K. Karsh, and A. A. Barbhuiya, "Literature review of vision-based dynamic gesture recognition using deep learning techniques," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 22, p. e7159, 2022.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[10] N. Aldausari, A. Sowmya, N. Marcus, and G. Mohammadi, "Video generative adversarial networks: A review," vol. 55, no. 2, 2022.

[11] M. Garg, D. Ghosh, and P. M. Pradhan, "Generating multiview hand gestures with conditional adversarial network," in *2021 IEEE 18th India Council International Conference (INDICON)*. IEEE, 2021, pp. 1–6.

[12] H. G. Doan, "Multiple views and categories condition gan for high resolution image," in *Artificial Intelligence in Data and Big Data Processing*. Cham: Springer International Publishing, 2022, pp. 507–520.

[13] T.-H. Tran, V.-D. Bach, and H.-G. Doan, "vi-mocogan: A variant of mocogan for video generation of human hand gestures under different viewpoints," in *Proceedings of the Pattern Recognition: ACPR 2019 Workshops*. Springer Singapore, 2020, pp. 110–123.

[14] K. Yang, H. Zhang, D. Zhou, and L. Liu, "Tgan: A simple model update strategy for visual tracking via template-guidance attention network," *Neural Networks*, vol. 144, pp. 61–74, 2021.

[15] A. Schäfer, G. Reis, and D. Stricker, "Anygesture: Arbitrary one-handed gestures for augmented, virtual, and mixed reality applications," *Applied Sciences*, vol. 12, no. 4, p. 1888, 2022.

[16] K. Gedamu, Y. Ji, Y. Yang, L. Gao, and H. T. Shen, "Arbitrary-view human action recognition via novel-view action generation," *Pattern Recognition*, vol. 118, p. 108043, 2021.

[17] T.-H. Tran, H.-N. Tran, and H.-G. Doan, "Dynamic hand gesture recognition from multi-modal streams using deep neural network," in *Multi-disciplinary Trends in Artificial Intelligence*. Cham: Springer International Publishing, 2019, pp. 156–167.

[18] D.Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 249–257, 2006.

[19] F. Murtaza, M. H. Yousaf, and S. A. Velastin, "Multi-view human action recognition using 2d motion templates based on mhis and their hog description," *IET Comput. Vis.*, vol. 10, no. 7, pp. 758–767, 2016.

[20] L. Wang, Z. Ding, Z. Tao, Y. Liu, and Y. Fu, "Generative multi-view human action recognition," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6211–6220.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," 12 2015, pp. 4489–4497.

[22] Q. Liu, X. Che, and M. Bie, "R-stan: Residual spatial-temporal attention network for action recognition," *IEEE Access*, vol. 7, pp. 82 246–82 255, 2019.

[23] Y. Bai, Z. Tao, L. Wang, S. Li, Y. Yin, and Y. Fu, "Collaborative attention mechanism for multi-view action recognition," *CoRR*, vol. abs/2009.06599, 2020.

[24] H.-T. Nguyen and T.-O. Nguyen, "Attention-based network for effective action recognition from multi-view video," *Procedia Computer Science*, vol. 192, pp. 971–980, 2021.

[25] N. u. R. Malik, U. U. Sheikh, S. A. R. Abu-Bakar, and A. Channa, "Multi-view human action recognition using skeleton based-fineknn with extraneous frame scrapping technique," *Sensors*, vol. 23, no. 5, 2023.

[26] H. Rahmani, A. S. Mian, and M. Shah, "Learning a deep model for human action recognition from novel viewpoints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 667–681, 2018.

[27] M. I. Lakhal, D. Boscaini, F. Poiesi, O. Lanz, and A. Cavallaro, "Novel-view human action synthesis," *CoRR*, vol. abs/2007.02808, 2020.

[28] B. Natarajan and R. Elakkiya, "Dynamic gan for high-quality sign language video generation from skeletal poses using generative adversarial networks," *Soft Computing*, vol. 26, no. 23, pp. 13 153–13 175, 2022.

[29] D.-M. Truong, D. Giang, T.-H. Tran, V. Hai, and T. Le, "Robustness analysis of 3d convolutional neural network for human hand gesture recognition," *International Journal of Machine Learning and Computing*, vol. 9, pp. 135–142, 04 2019.

[30] Y. Liu, J. Yan, and W. Ouyang, "Quality aware network for set to set recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4694–4703.

[31] H. Doan, T. Tran, H. Vu, T. Le, V. Nguyen, S. V. Dinh, T. Nguyen, T. T. Nguyen, and D. Nguyen, "Multi-view discriminant analysis for dynamic hand gesture recognition," in *Pattern Recognition - ACPR 2019 Workshops, Auckland, New Zealand, November 26, 2019, Proceedings*, ser. Communications in Computer and Information Science, vol. 1180. Springer, 2019, pp. 196–210.

[32] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "Fvd: A new metric for video generation," 2019.

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[34] A. Rossholm and B. Lövström, "A new video quality predictor based on decoder parameter extraction," in *Signal Processing and Multimedia Applications*, 2018.

[35] A.-V. Bui and T.-O. Nguyen, "Multi-view human action recognition based on tsn architecture integrated with gru," *Procedia Computer Science*, vol. 176, pp. 948–955, 2020.

[36] D. Wang, W. Ouyang, W. Li, and D. Xu, "Dividing and aggregating network for multi-view action recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[37] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 20–36.

[38] S. Mambou, O. Krejcar, K. Kuca, and A. Selamat, "Novel Cross-View Human Action Model Recognition Based on the Powerful View-Invariant Features Technique," *Future Internet*, vol. 10, no. 9, pp. 1–17, 2018.

[39] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3d exemplars," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–7.

[40] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "Cross-view action recognition from temporal self-similarities," in *Computer Vision – ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 293–306.

[41] J. Liu, M. Shah, B. Kuipers, and S. Savarese, "Cross-view action recognition via view knowledge transfer," 06 2011, pp. 3209–3216.

[42] N. Nida, M. H. Yousaf, A. Irtaza, and S. Velastin, "Video augmentation technique for human action recognition using genetic algorithm," *ETRI Journal*, vol. 44, p. 327–338, 01 2022.

[43] J. Liu, S. Ali, and M. Shah, "Recognizing human actions using multiple features," 06 2008.

[44] F. Baradel, C. Wolf, J. Mille, and G. Taylor, "Glimpse clouds: Human activity recognition from unstructured feature points," 06 2018, pp. 469–478.

[45] K. Shah, A. Shah, C. P. Lau, C. M. de Melo, and R. Chellapp, "Multi-view action recognition using contrastive learning," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 3370–3380.