SPECIAL SECTION ON DEEP LEARNING TECHNOLOGIES FOR INTERNET OF VIDEO THINGS

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 8, 2021, accepted July 2, 2021, date of publication July 20, 2021, date of current version July 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098840

End-to-End Key-Player-Based Group Activity Recognition Network Applied to Basketball Offensive Tactic Identification in Limited Data Scenarios

TSUNG-YU TSAI[1], YEN-YU LIN[2], (Member, IEEE), SHYH-KANG JENG[1], (Senior Member, IEEE), AND HONG-YUAN MARK LIAO[3], (Fellow, IEEE)

[1]Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan
[2]Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan
[3]Institute of Information Science, Academia Sinica, Taipei 115, Taiwan

Corresponding author: Tsung-Yu Tsai (laputianzen@gmail.com)

ABSTRACT In this paper, we propose an end-to-end key-player-based group activity recognition network specially applied to the identification of basketball offensive tactics in limited data scenarios. Our previous studies show that basketball tactics can be better recognized via key player detection with multiple instance learning (MIL) using the support vector machine (SVM). However, the SVM in that work is required to extract features depending on basketball- and tactic-specific knowledge for good performance. Thus, in this study, we develop an end-to-end trainable neural network without prior knowledge and integrate MIL into it. As long as a tactic label is given, MIL can train the network to identify tactic's key players. For testing, our network can recognize the key players in a video clip and provide a tag of the tactic related to them. Like other neural network models, our network requires a large annotated dataset. At the same time, we could collect only a few labeled data, which is common in dealing with group activity recognition. To overcome such a limitation, we propose a novel data augmentation framework, the tactical-based conditional generative adversarial network (GAN), for generating new labeled trajectories. The experimental results show that our method significantly improves 9.13% in tactic recognition and 4.965% in key player detection.

INDEX TERMS Data augmentation, end-to-end deep neural networks, generative adversarial networks, group activity recognition, key player detection, multiple instance learning, sports video analysis.

I. INTRODUCTION

Group activity recognition is a widely used but challenging problem. Generalized from single-person activity recognition, group activity recognition needs to deal with complicated dynamics among people, including individual's role, the interaction among different individuals, and each behavior. Although the existing human activity recognition algorithm can accurately identify individual actions, there is still much room for improvement in identifying group interactions. For example, the graphical model can adequately describe the relationship of intermediate action through nodes and edges. However, the number of behaviors that can be included is greatly restricted. On the other hand, statistical learning has no limit on the number of actions, but its capability in analyzing the interaction among human behavior is deficient. Due to the rapid development of deep learning in recent years, one can use data-driven techniques to simultaneously identify a larger number of behavior patterns, capture the division of roles, and analyze their interaction.

In this work, we put our emphasis specifically on analyzing group behavior in sports. Sports include all forms of competitive physical activities or games through casual or organized participation. Such analysis improves physical ability and skills while providing enjoyment to participants and, in some cases, entertainment for spectators.

In cooperative group activities, multiple players typically act according to pre-defined tactics. Needless to explain, recognition of the tactics taken is essential for coaches and players. The audience can also enjoy more at the same time

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

if they can identify the tactics applied by both teams. Traditionally, such recognition is done by senior sports analysts at a very slow pace. It would be useful if the computer can recognize tactics and display them to the coaches and the audience. The players can also be beneficial in learning the tactics through such a computer program. This study aims to develop tools for recognizing basketball as an initial attempt for general group activity identification applied to sports.

For a given tactic, a subset of players is required to perform particular behaviors. Based on this observation, we can separate all players into two groups. One is the key player group, which covers core members of the tactic operation and has small intra-tactic variation. The other non-key player group contains the rest of the players and usually has a larger intra-class variation. Most tactics are characterized by activities of the key player group. Thus, we transform the tactic recognition to a detection problem of the key player group. Our method provides better tactic recognition results and better recognition interpretation since key players are detected for verification.

We adopt the MIL to detect the key players, with the identified tactic as the bag label and moving trajectories of each player subset as instance. The MIL combines with handcrafted spatial-temporal features named motion intensity maps (MIM) from recorded video clips and provides satisfactory recognition results in [1].

However, using the handcrafted features has many drawbacks. First, MIM features heavily rely on prior knowledge of basketball courts, which is pretty cumbersome to prepare. Second, to represent videos of different temporal lengths in vectors of the same dimension, a fixed number of time segments is taken. A simple average operation is then applied to get specific segment's features. All segment features are concatenated into a global dynamic feature. This re-sampling method ignores real pace information.

To overcome those drawbacks, we propose an end-to-end trainable network. This network's input is the position, i.e., (x, y) coordinate, of each player along the temporal axis. Since our model only requires raw trajectories without any prior basketball knowledge, it is also applicable to different group sports such as soccer and volleyball and group behaviors appearing in surveillance recordings. In addition to key player pattern detection, our method can carry out temporal pattern discovery, leading to an in-depth understanding of tactic interaction.

Deep networks require a large number of training data to train the parameters of the model. According to experience, the best number of training data is several times more than the number of network weights. For still image object recognition, a researcher can pre-train model on large-scale datasets such as ImageNet, CoCo, etc., then fine-tune the network using a customized dataset. However, for time series learning problem like ours, well-known datasets like ImageNet or MS COCO dataset that can provide a large amount of labeled data are not available. The dynamic characteristics: including the posture, trajectory, and position of the object that change over time, often yield large diversity due to the varying settings of the start and the end time, or the difference in the viewing position. To increase model robustness for temporal variation, we need more data for training the network. The disadvantage of a small dataset is even apparent. However, if we have only a small dataset, as in many application scenarios of group activity recognition, we may use data augmentation. Data augmentation can be achieved by either adding slightly modified copies of existing data or synthesizing new data from existing data. Adding slightly modified copies is easy to implement through geometric transformation (e.g., rotation, cropping, etc.), temporal transformation (e.g., resampling, repace, etc.), but its scale factor in such approaches is pre-defined and inflexible. Synthesizing new data does not have restrictions on the scale factor but needs a good generative model. We found that the GAN is an excellent one and is very suitable for our purpose.

II. RELATED WORK

Group behavior analysis on sports video has been explored in the literature to analyze individual and team performance. Methods of this category are used for various applications such as football play recognition [2], [3] and basketball behavior detector [4] as well as feature extraction such as team occupancy and team centroid features [5]. For basketball tactic recognition, Chen *et al.* [6] make a breakthrough by using dynamic time warping (DTW) with the Gaussian mixture model (GMM). Our previous work [1] first introduces the idea of tactic recognition via key player detection, where the authors extracted handcrafted basketball court MIM features with a MIL classifier.

Recently, deep neural networks have become a powerful learning algorithm owing to their strong feature extraction capability. Under the deep neural network's framework, group activity recognition shares many common layers, such as a dynamic feature extractor and a classification layer, with single-person activity recognition. The main difference between single-person and group activity recognition is that group activity recognition has an additional aggregation layer. An aggregation layer, which tries to merge multiple individual features, can be formulated in different ways, e.g., simple concatenation [7], [8], average or max-pooling [9], attention [10], [11], and the semantic graph [11], [12]. Our method introduces the so-called ''NchooseK'' layer, which is a new type of aggregation method based on an assumption of key players. It turns out that an end-to-end neural network for key player detection can be derived based on the proposed NchooseK layer.

Global average pooling (GAP) and class activation map (CAM) are introduced by Zhou *et al.* [13]. Via global average pooling and a fully-connected layer with linear activation before the softmax layer, their method not only reduces the number of learnable parameters but also applies the commutative law of multiplication to evaluate the contribution of each feature vector to the specific class score. Although CAM is effective, the requirement of global average pooling

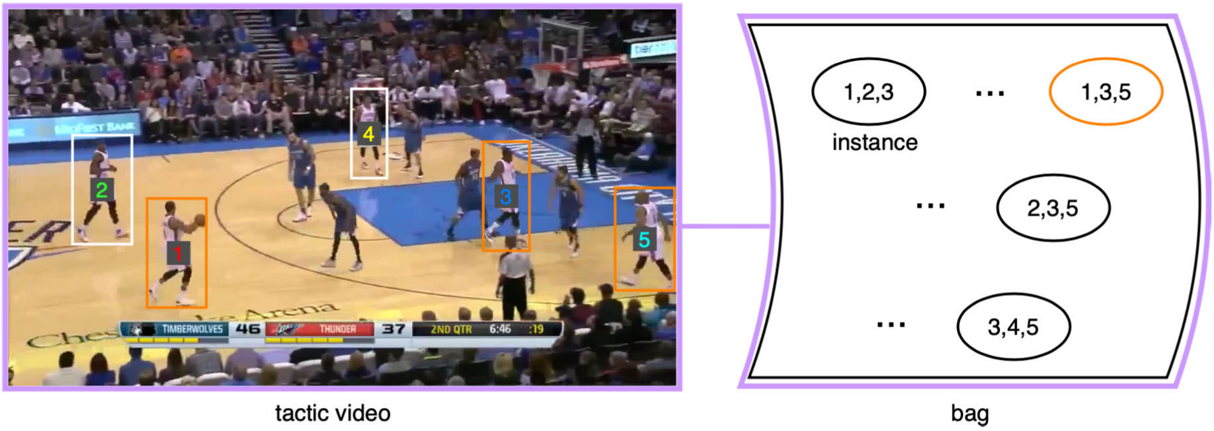

FIGURE 1. Concept of key-player-based tactic classification. This figure shows a video with a tactic that involves three key players. This video is viewed as a *bag* with a set of *instances* (ellipses), each of which covers a specific group of three of the five offensive players. The positive instance here is that includes player 1,3 and 5 as key players (orange ellipse).

placed on the last classification layer limits its usage on more complex and pre-trained neural networks. Selvaraju *et al.* [14] propose a modified version of CAM, called Grad-CAM. Grad-CAM uses backpropagation gradients of succeeding layers as class feature weight without requiring GAP layer. CAM and its variants can be used in a variety of applications. Due to their flexibility of feature maps' size, they can also be used on time series classification, sequence-to-one classification [15], [16], which merges different-length time sequences to a global feature of the same dimension. Our method uses a similar strategy to CAM, but we further leverage the mutually exclusive property of key player instances.

In this work, MIL and GAN play important roles in our method. Here we shall briefly review these two topics as follows. MIL is a type of weakly supervised learning. It is introduced by Dietterich *et al.* [17] for drug activity detection and has various applications to image classification [18], object detection [19], text or document categorization [20], and semantic segmentation [21], [22]. MIL algorithms originally work on pre-defined features. But with the development of deep learning, a variety of neural networks with MIL have been proposed for combining the powerful feature extraction capability of deep learning and the low-cost labeling of MIL. Zhou and Zhang [23] propose an instance-space MIL algorithm, casting instance features to instance scores followed by MIL pooling on the score layer. Wang *et al.* [24] propose an embedded-space MIL algorithm that performs MIL pooling directly on the feature domain. Instance-space methods allow the identification of positive instances but with a lower performance than embedded-space methods. Ilse *et al.* [25] use an attention layer to combine instance-space interpretation and embedded-space. Instances in this research are mutually exclusive. Thereby, instance-space max-pooling matches our requirement.

The GAN is a popular deep generative network in recent years. It is proposed by Goodfellow *et al.* [26]. Deep generative models before GAN are beautiful in theory, but it is not very effective in practical applications. Among them, the models belonging to the undirected graphical model include Restricted Boltzmann machine (RBMs) [27], Deep Belief Networks [28], and Deep Boltzmann Machines (DBMs) [29]. This kind of Boltzmann Machine-based generative model uses maximum likelihood to estimate the value when calculating the data distribution. The calculation is very complicated, and other solutions except trivial solutions are very difficult to obtain. Using the Markov Chain Monte Carlo (MCMC) method to find an approximate solution is an alternative way, but MCMC is also a complicated method. To avoid complicated calculations like the computation of log-likelihood, researchers have proposed other optimization methods like the score-matching [30] and the noise-contrastive estimation (NCE) [31]. But on most occasions, the density function is not normalized and estimating a normalization constant is also very time-consuming. As to the backpropagation method, although one can use labels to simplify the process, it is only feasible for the tasks like pattern matching. It cannot be used directly for a task that needs the recursive instruction to be executed during training. On the other hand, the GAN model uses two coupled networks and applies the min-max game algorithm to train both networks alternatively with the existing backpropagation methods. Because the generator in the GAN architecture has sound data generation capability, [32]–[34] prove that it can be used for a wide variety of data augmentation. There are two kinds of trajectory augmentation. One is known as trajectory prediction, in which an initial sequence is given, and with the initial sequence, the model generates successive data. The other is called trajectory simulation, where models simulate the entire trajectory. In general, trajectory simulation is more difficult to implement than trajectory prediction. Social GAN [35] and GD GAN [36] are examples of using GAN to perform trajectory prediction. Since we need to increase the total amount of training trajectories, trajectory simulation is more suitable for our demand. Crowd simulation [37] is a virtual simulation of the entire trajectory. The strategy it adopts is a method inherited from the trajectory prediction method.

The neural network corresponding to trajectory prediction will read the initial trajectory produced by another independent neural network and make the prediction. Those two neural networks work together to produce the final trajectory. The main difference between our method and the crowd simulation is that the latter adds movement constraint to generated trajectories to simulate various pedestrian interactions. Instead, we are inspired by conditional GAN [38]and add tactical information as conditional input to the original Crowd simulation GAN. We call this enlarged GAN as Group-Tactic-Role conditional GAN (GTRCGAN).

III. OUR APPROACH

A. NETWORK OVERVIEW

Given a set of half-court offensive videos, each of which belongs to one of the C tactics. The trajectories of the five offensive players in the video i have been retrieved and denoted by $\{\pi_{i,p}\}_{p=1}^{5}$. Each trajectory is a temporal sequence of that player's (x, y) positions in the court i.e. $\pi_{i,p} = \{\pi_{i,p,x}(t), \pi_{i,p,y}(t)\}_{t=1}^{F}$, where F is the frame number of video i. It is worth noticing that the frame number varies from video to video. Besides, the five trajectories in each video are orderless. Each player has a tactical label, which contains tactical type c_i and role ID r_p. Note that role ID r_p here represents a specific player movement in a specific tactic, not the basketball positions such as center, forward, and guard. As a result, players of different tactics that have the same role ID do not mean they have a similar trajectory. On the other hand, players of different tactics whose role IDs are different might have a similar trajectory.

Our neural network is designed for multi-class classification. It consists of multiple subnets for various key player groups. Figure 2 (a) shows the subnet where the number of key players is K. This network is composed of four high-level layers, including (1) RNN auto-encoder for individual player feature extraction, (2) the NchooseK layer for group instance aggregation, (3) temporal global average pooling for dimensionality reduction, and (4) instance-space miNet for MIL. To train the network, the loss function with two terms is described as (1):

$$\mathcal{L} = \mathcal{L}_{\text{cross-entropy}} + \lambda \mathcal{L}_{\text{auto-encoder}}, \tag{1}$$

where $L_{\text{cross-e}}$ represents the cross-entropy loss of tactic classification, and $L_{\text{auto-encoder}}$ represents the auto-encoder loss which calculates the Euclidean distance between an original trajectory and decoded trajectory from auto-encoder. The details of the network components and the loss functions are given in the following.

B. RNN AUTO-ENCODER FOR PLAYER FEATURE EXTRACTION

To capture the features of each player, a single layer of recurrent neural network (RNN) is adopted, which casts the player's (x, y) coordinates to a D-dimensional hidden state features $\overline{h}_p^t$ at each timestamp t. To maintain the correlation between the hidden state feature $\overline{h}_p^t$ and the original trajectory for video i, an auto-encoder is incorporated with RNN for regularization to avoid overfitting and its loss function is defined by (2)

$$\mathcal{L}_{\text{auto-encoder}} = \sum_{p=1}^{5} \|\pi_{i,p} - \tilde{\pi}_{i,p}\|^2, \tag{2}$$

where $\pi_{i,p}$ and $\tilde{\pi}_{i,p}$ represent the original and decoded trajectories, respectively.

C. NCHOOSEK LAYER FOR GROUP INSTANCE AGGREGATION

NchooseK layer aggregates individual player features to group features. The proposed aggregation layer is developed based on grouping the key player number n_c out of the five players together for a specific tactic c. Since trajectories are randomly ordered, we list all possible $C_{n_c}^5$ groups. The positive instance is one of the $C_{n_c}^5$ instances which covers all key players. This layer is called the NchooseK layer. The output of this layer is $\{\overline{h}_k^t\}_{k=1,t=1}^{C_{n_c}^5, F}$.

D. TEMPORAL GLOBAL AVERAGE POOLING

To overcome the problem that different videos may have different numbers of frames, temporal global average pooling (tGAP) is applied along the temporal dimension, as illustrated in Figure 3. After tGAP, the temporally pooled features for each instance k is denoted by $\overline{H}_k$.

E. INSTANCE-SPACE MINET FOR MULTIPLE INSTANCE LEARNING

A fully-connected layer is used to cast high-dimensional features $\overline{H}_k$ to class prediction probability s_k^c of whether instance k is positive for tactic c. Because each video has just one positive instance, which is supposed to be the one with the maximum score value s_k^{c*}. Regarded as MIL pooling, a max-pooling is used to obtain the final class prediction Y^c, i.e., $Y^c = \max_k s_k^c$. Finally, we compute Y^c for each tactic c and concatenate them into the tactic prediction vector $\overline{Y}$ of this video. Since we have the ground-truth tactic label for each training video, cross-entropy is used to define loss $L_{\text{cross-entropy}}$ in Eq. (1). Note that our method can predict the key players since the positive instance is found via max-pooling. Figure 2 (c) shows key instance prediction is obtained by adding an argmax pooling when network inferences.

F. GENERATING TRAJECTORIES USING GAN

As illustrated in Figure 4, for augmenting enough data, we take the architecture proposed in [37] and make two modifications: (1) we modify the characteristic of trajectory π from position-based coordinates to displacement-based coordinates. This modification is reasonable because a displacement-based feature can better characterize the causality relationship than a position-based feature; (2) we

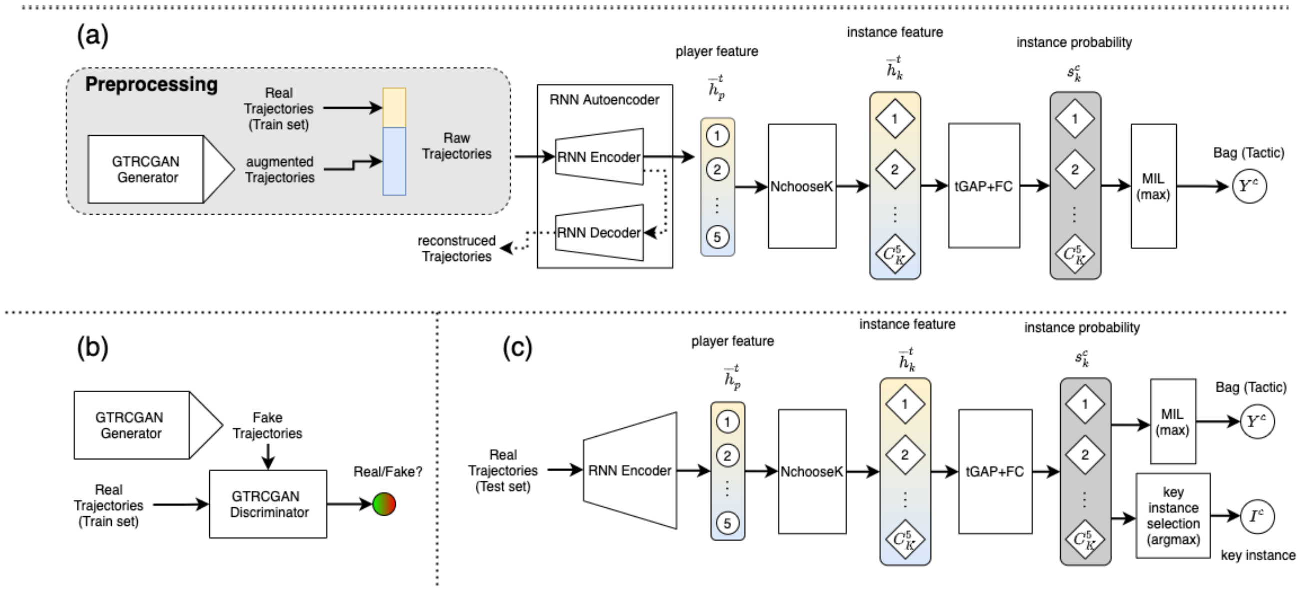

FIGURE 2. Overview of the proposed end-to-end trainable neural network, where a tactic with K key players is considered. (a) The training process of proposed MIL Network with pre-trained GTRCGAN generator as data augmentation preprocessing module. (b) The training process of our augmentation network (GTRCGAN), where the detail is described in Fig. 4. (c) Inference process of our MIL Network, a non-trainable argmax layer is added for key instance selection. Combined with the MIL layer, our network can predict a tactic and key players simultaneously.

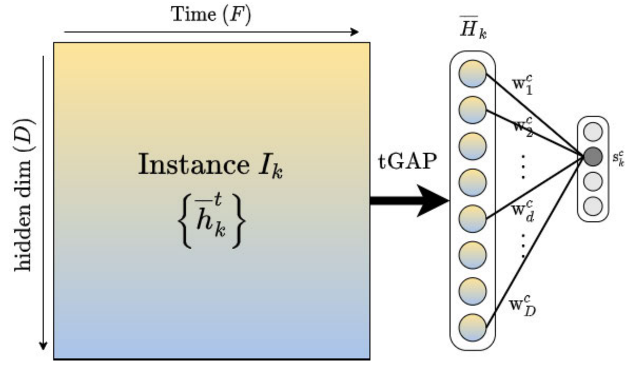

FIGURE 3. Overall of temporal global average pooling (left) and miNet (right). For each video, there are $K = C_{n_C}^5$ instances for tactic c. Each instance has temporal length F and hidden dimension D. We apply temporal global average pooling over the temporal axis. The temporally pooled features for instance k are denoted by $\bar{H}_k$. A fully connected layer is derived to estimate the probability s_k^c of whether instance k is positive for tactic c. A max-pooling is applied to all instance probabilities $\{s_k^c\}_{k=1}^K$ for MIL.

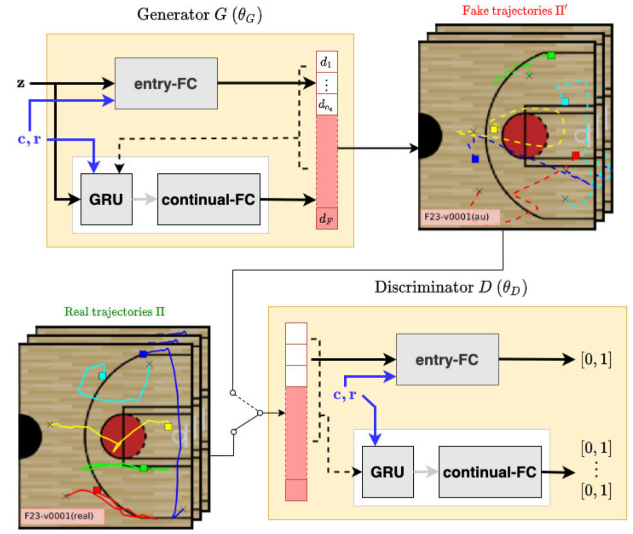

FIGURE 4. Overview of group-tactic-role conditional GAN. Generator (upper left) takes latent variable z and tactical label c, r as input to generate augmented trajectories. Discriminator (bottom right) takes trajectory and tactical label c, r to classify whether the each point of trajectory is real(1) or fake (0).

combine both the tactical label c_i and the tactical role r_p as the conditional input of GAN. Thus, in the implementation, we will combine c_i and r_p and make it a one-hot encoding vector $\mathbb{1}_{c_i,r_p}$ of size $C \times 5$. In the original architecture in [37], the entire trajectory was divided into two groups of generating tasks, namely the one for initial path (entry-points part) and the other for subsequent path (continual-points part). Entry-points part refers to the beginning part of the track, which contains n_e data points. As for its generator, an entry fully-connected layer is applied to integrate latent variables z and conditional input $\mathbb{1}_{c_i,r_p}$ to generate a displacement sequence $\{d_1, \cdots, d_{n_e}\}$. The discriminator of the original architecture will take the sequence $\{d_1, \cdots, d_{n_e}\}$ as the input of the entry fully-connected layer and then output a value sitting within the range [0,1]. We denote this sequence-to-one output as $v_e(d^{\pi}_{1:n_e}|\mathbb{1}_{c_i,r_p}; \Theta_D)$ for judging whether the initial path is correct or not. As for the continual-points part, which contains $F - n_e$ data points, we consider a sequence-to-sequence model. The generator uses the gated recurrent unit (GRU) to read the displacement sequence from time 1 to $t - 1$, the latent variable z, and the conditional input $\mathbb{1}_{c_i,r_p}$. Then we transform the hidden state at time t into displacement d_t through a continual fully-connected layer, and this transformation can be represented as $g(z|d^{\pi}_{1:t-1}, \mathbb{1}_{c_i,r_p}; \Theta_G)$. As to the discriminator, we use GRU and a continual fully-connected layer to calculate the probability that d_t is true under a given displacement sequence $\{d_1, \cdots, d_{t-1}\}$, which can be represented as $v_c(d^{\pi}_{1:t}|\mathbb{1}_{c_i,r_p}; \Theta_D)$.

During the training process, the generator will generate a set Π' of $N \times 5$ trajectories as quantities of batch size from different sequences of noise vectors in each iteration. The loss function in total includes four items: 1. The recognition success rate of the initial path is described as

$$\sum_{\pi}^{\Pi} \log v_e(d^{\pi}_{1:n_e}|\mathbb{1}_{c_i,r_p}; \theta_D) + \sum_{\pi}^{\Pi'} \log(1 - v_e(d^{\pi}_{1:n_e}|\mathbb{1}_{c_i,r_p}; \theta_D)), \tag{3}$$

where the first item above is the probability that trajectory π from the real database Π being judged as real, and the second item is the probability that trajectory π' of the augmented database Π' from the generator being determined as fake.

2. Recognition success rate of subsequent path is described as

$$\sum_{\pi}^{\Pi} \sum_{t=n_e+1}^{F_\pi} \log v_c(d^{\pi}_{1:t}|\mathbb{1}_{c_i,r_p};\theta_D) + \sum_{\pi'}^{\Pi'} \sum_{t=n_e+1}^{F_{\pi'}} \log(1 - v_c(d^{\pi'}_{1:t}|\mathbb{1}_{c_i,r_p};\theta_D)), \tag{4}$$

where the first term above is the probability of trajectory π being judged as real from the real database Π, while the second term is the probability of trajectory π' being judged as fake from the augmented datasbase Π' of the generator.

3. The Euclidean distance between generated trajectory $\pi'_{i,p}$ and real trajectory $\pi_{i,p}$ is described as

$$\sum_{\pi,\pi'}^{\Pi,\Pi'} \sum_{p=1}^{5} \|\pi_{i,p} - \pi'_{i,p}\|^2. \tag{5}$$

4. The boundary condition of the court is described as

$$L_{boundary} = \begin{cases} \|\pi'_{i,p,m} - \mathrm{LB}_m\|^2, & \pi'_{i,p,m} < \mathrm{LB}_m, \\ 0, & \mathrm{LB}_m \le \pi'_{i,p,m} \le \mathrm{UB}_m \\ \|\pi'_{i,p,m} - \mathrm{UB}_m\|^2, & \pi'_{i,p,m} > \mathrm{UB}_m, \end{cases} \tag{6}$$

where $m \in \{x, y\}$, LB_m and UB_m are the lower and upper boundary of the court. We have adopted soft boundary, i.e., the generated trajectory $\pi'_{i,p}$ is allowed a little bit over the boundary.

G. IMPLEMENTATION DETAILS

Our model is implemented using the TensorFlow deep learning framework. In the MIL Network, the RNN hidden state dimension is set to 512 and the NchooseK pooling is carried out by max pooling. We use the RMSprop optimizer for training with a learning rate 0.001 and batch size 2. Each trajectory is down-sampled by a factor of 10 to reduce training memory load and accelerate training speed.

In the GAN model, latent variable z uses the 2-dimensional uniform random distribution. In generator G and discriminator D, the entry-point part contains a layer of 512-dimensional fully-connected layer, and the number of the generated entry point is 1. In the continual-point part, it contains a 100-cell GRU block and 128-dimensional fully-connected layer. For obtaining the best accuracy, the GRU block will consider all trajectory points, including those generated from the entry-point part. When training GAN, after a large number of parameter tunings, we use RMSprop optimizer like the MIL Network, the learning rate is set to 0.001, but the batch size is changed to 50. Since the generator is more difficult to converge than the discriminator, every time we update the parameters of the latter once, we must update those of the former three times. For all fully-connected layers, we use leaky-ReLU activation to avoid the gradient-vanishing problem. To achieve the best performance of GAN, we set Euclidean loss weight 32.0 and boundary loss weight 1.0 in the loss term.

IV. EXPERIMENTAL RESULTS

A. DATASET USED FOR EVALUATION

To evaluate our method's performance, we perform experiment on a dataset from [6], which contains 134 videos of the NBA 2013-2014 season. These videos are distributed over 10 half-court offensive tactics and the offensive player trajectories are also available. The details of this tactic dataset are given in Table 1.

TABLE 1. **Abbreviation and the numbers of videos and key players for each tactic in the experiments.**

tactic	abbr.	# video	# key players
2-3 Flex	F23	15	3
Elevator	EV	11	3
Hawk	HK	20	3
Pin-Down	PD	9	3
Princeton	PT	13	5
Back-Side Pick and Roll	RB	15	3
Side-Pick Slip and Pop	SP	15	2
Warrior Single	WS	13	3
Weave	WV	16	5
Wing-Wheel	WW	7	2

B. PERFORMANCE MEASURE AND EVALUATION PROTOCOL

Our method requires two parts to verify its effects. One is the MIL Network, and the other is the GAN.

A simple yet effective accuracy measure is adopted to evaluate and compare the performance of different methods with our MIL Network. Average tactic accuracy, which is abbreviated as tactic accuracy, first calculates each tactic's accuracy and then averages over the accuracy of all tactics. Similar to the average tactic accuracy, the average key player accuracy, which is abbreviated as key player accuracy, is obtained by first calculating key player accuracy of each tactic and then averaging over all tactic key player accuracy.

The evaluation protocol in the experiments is 5-fold cross-validation. Due to the small size of the dataset, average tactic accuracy and key player accuracy are computed multiple times for each hyperparameter value to further reduce metric variations caused by random initialization.

In using GAN to augment the data, we use Euclidean distance of (x, y) coordinate (ED) to calculate the similarity between the augmented trajectory and the referenced trajectory. However, the augmented trajectory should be similar to the referenced trajectory but not precisely the same (i.e., ED is small yet not 0). We will use a visualization tool to check the quality of the augmented trajectory. As for selecting the best weight, we adopt the same protocol as the one in MIL Network, i.e., using 5-fold cross-validation for verification.

C. COMPARISON WITH THE STATE-OF-THE-ART METHODS

Four different methods are compared. The first one is learning the spatial-temporal template by unsupervised Gaussian mixture model [6]. The second one is the same as the first one except that the ground-truth tactic labels are provided to train the Gaussian mixture model. The third one adopts multiple-instance-learning mi-SVM with handcrafted spatial-temporal features named motion-intensity-map (MIM), where the feature dimension is set to 1040. The fourth one is our method, called RNN-tCAM-miNet+GTRCGAN, whose overall structure is described in Figure 2, where the feature dimension is 512.

TABLE 2. Performance comparison of four methods in both tactic recognition accuracy and key player detection accuracy.

method	tactic accuracy	key player accuracy
unsupervised GMM [6]	0.8550	-
supervised GMM [6]	0.8867	-
MIM + mi-SVM [1]	0.8933	0.7143
RNN-tCAM-miNet + GTRCGAN	**0.9846**	**0.76395**

Table 2 illustrates the results of tactic recognition accuracy and key player detection accuracy on different methods. Unsupervised and supervised GMM models are not able to provide key player results. The tactic accuracy of unsupervised GMM is 0.8550, and that of supervised GMM is 0.8867. The method using mi-SVM reaches tactic accuracy of 0.8933 with key player accuracy of 0.7143, serving as a baseline of supervised methods.[1] The proposed model achieves tactic accuracy of 0.9846 with key player accuracy of 0.76395. Our method substantially improves tactic accuracy by 9.13% and key player accuracy by 4.965% compared to the mi-SVM model. From the quantitative results, our deep neural network model outperforms conventional SVM models with fewer heuristic parameters and a lower feature dimension.

D. ABLATION STUDIES

Table 3 reports our method's ablation studies with other popular group-aggregation methods and handcrafted features with miNet. The first competing method MIM + miNet replaces the RNN auto-encoder in our model with handcrafted feature MIM and obtains tactic accuracy of 0.8868 with key player accuracy of 0.6100. Our method improves tactic accuracy by 9.78% and key player accuracy by 15.395%, which confirms that the end-to-end neural network provides better features than the handcrafted MIM. The second competing method is called RNN-LastStep-Net, where our temporal global average pooling layer is replaced with the last step of RNN auto-encoder achieves tactic accuracy of 0.9489 and key player accuracy of 0.7173. Our method improves tactic accuracy by 3.57% and key player accuracy by 4.665%, which results from replacing the last step pooling with temporal global average pooling. The third method RNN-CAM-Net uses the global average pooling as the aggregate layer on every single player, which removes a "NchooseK" aggregation layer, resulting in tactic accuracy of 0.9443 and key player accuracy of 0.6311. Our method improves tactic accuracy by 4.03% and key player accuracy by 13.285%. The large improvement in key player accuracy proves the effectiveness of the proposed NchooseK layer. The fourth competing method is RNN-GMP-Net, which replaces temporal global average pooling layer with global maximum pooling (GMP), obtaining tactic accuracy of 0.8263 and key player accuracy of 0.7137. The proposed method improves tactic accuracy by 15.83% and key player accuracy by 5.025%. The significant improvement in tactic accuracy results from tCAM which considers the whole trajectory time interval while GMP refers to a time step with maximum value. The fifth competing method is RNN-tCAM-miNet, which only uses real trajectory in model training, obtaining tactic accuracy of 0.98135 and key player accuracy of 0.70585. The proposed method improves tactic accuracy by 0.325% and key player accuracy by 5.81%. The improvement between the fifth and proposed methods proves the effectiveness of using GAN augmented data in training complex models. The reason that GAN only improves accuracy on key player detection is that in MIL tactic can classified true even if the key instance is not selected. GAN greatly increases the discriminativity of key players and reduces the chance of non-key players subgroup being selected as the key instance. Without GAN augmented data, RNN-tCAM-miNet can still recognize correct tactic but the rate of detecting wrong key instance increases. As a result, GAN has little effect on tactic accuracy. From the ablation study, we know that RNN is a powerful feature extraction tool. But to get RNN maximum capability, our design NchooseK and tCAM layer that allows tactic information to be properly backpropagated into RNN. GAN provides more role-player trajectory information that allows RNN-tCAM-miNet to predict trajectories of key players more precisely.

[1] The accuracy here is different from the result in [1]. Because in [1], the accuracy is executed only once and chooses the best result. Here, it is the average value obtained after repeated executions.

TABLE 3. Performance of different ablative methods.

Name	RNN Auto-encoder	NchooseK	GAP	mi-Net	GAN	tactic accuracy	key player accuracy
MIM + miNet	×	×	×	✓	×	0.8868	0.6100
RNN-LastStep-Net	✓	✓	×	✓	×	0.9489	0.7173
RNN-CAM-Net [13]	✓	×	✓	✓	×	0.9443	0.6311
RNN-GMP-Net	✓	✓	×	✓	×	0.8263	0.7137
RNN-tCAM-miNet	✓	✓	✓	✓	×	0.98135	0.70585
RNN-tCAM-miNet + GTRCGAN	✓	✓	✓	✓	✓	**0.9846**	**0.76395**

TABLE 4. Accuracy per tactic in RNN-tCAM-miNet + GTRCGAN model.

tactic	tactic accuracy	key player accuracy
F23	1.0000	0.9200
EV	0.9250	0.6100
HK	0.9375	0.9800
PD	1.0000	1.0000
PT	1.0000	1.0000
RB	1.0000	0.3600
SP	1.0000	0.7200
WS	0.9833	0.33995
WV	1.0000	1.0000
WW	1.0000	0.7100

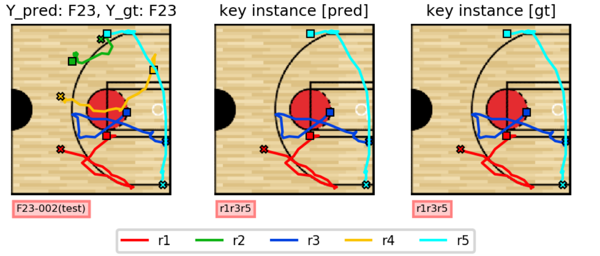

FIGURE 5. Correct tactic classification and key player detection. The left column shows a total of 5 offensive players with a tactic prediction*(Y_pred)* versus the ground truth*(Y_gt)* at the top of the court. The middle column shows key players from model prediction. The right column shows key players from expert-labeled ground truth. The legend on the bottom is the color annotation of role *(r1-r5)*.

E. VISUALIZATION

To analyze the impact of our model on each tactic thoroughly, we demonstrate the accuracy per tactic in Table 4. For tactical accuracy, the difference between each other is small, and all tactics have an accuracy rate of more than 90%. Our model has a relatively large difference between tactics in key player accuracy, which ranges from the lowest 33% to 100%. The reason for this difference is that although the MIL finds consistent instance for each tactic, but in some tactics the instance found is not a key players instance defined by experts. To explain this phenomenon intuitively, we design a visualization tool for illustration.

Figure 5 shows the result on video F23-002, which is classified as a correct tactic with correct key player detection. We notice that tactic accuracy is higher than key player accuracy in every model. After a thorough examination, errors can be separated into two categories. The first one is a wrong tactic classification. As shown in Figure 6, column (a) displays the trajectories of five offensive players. Column (b) plots the ground truth key instance labeled by professionals, and player 4 (yellow) on a HK-039 has a very different trajectory compared to the lower part of column (b). This long trajectory also makes another instance r2r3r4 (role 2, role 3, and role 4) at the upper part of column (c) looks like the positive instance of tactic F23 at the lower part of column (c).

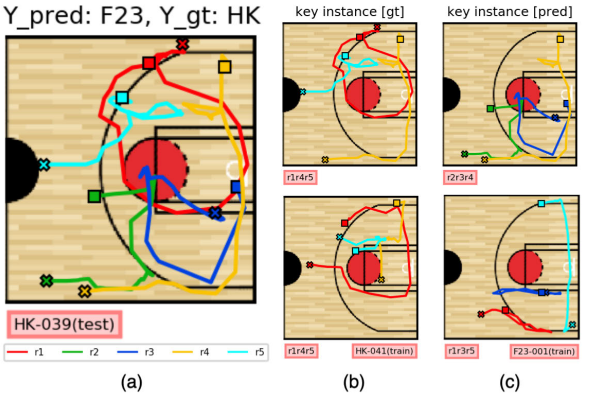

FIGURE 6. Failure case: Wrong tactic classification. Column (a) shows a total of 5 offensive players with a tactic prediction*(Y_pred)* versus the ground truth*(Y_gt)* at the top of the court. Column (b) shows ground truth key players of test video (top) versus key players of reference video from the same tactic (bottom). Column (c) shows predicted key players of test video (top) versus key players of reference video from misclassified tactic (bottom).

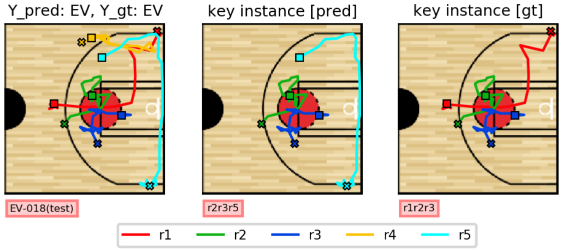

FIGURE 7. Failure case: Correct tactic classification but with wrong instance detection. The left column show a total of 5 offensive players with a tactic prediction*(Y_pred)* versus the ground truth*(Y_gt)* at the top of the court. The middle column shows key players from model prediction. The right column shows key players from expert-labeled ground truth. The legend on the bottom is the color annotation of role *(r1-r5)*.

The second category is a correct tactic classification with a wrong instance detection. As displayed in Fig. 7, video EV-018 is correctly predicted as tactic EV. However, instead of detecting the correct positive instance r1r2r3 at the middle column, the proposed system chooses another instance r2r3r5. This is due to that our features sometimes cannot separate non-key players from the key players, because non-key players may also have regular trajectories. Even humans cannot distinguish key players from non-key players

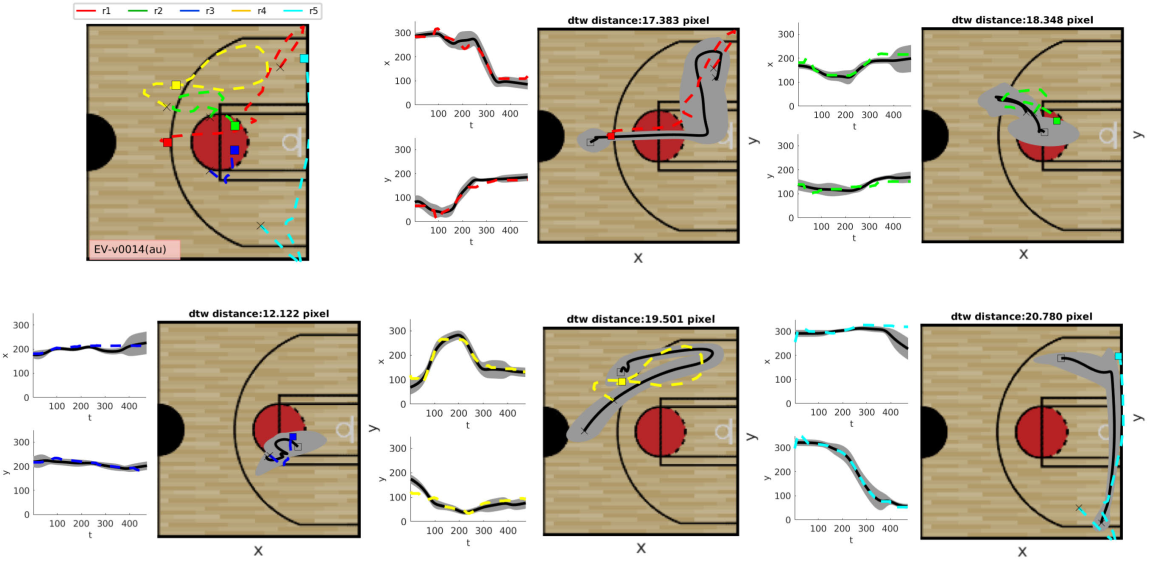

FIGURE 8. **Similarity visualization on template learning by spatial-temporal clustering [6].**

without extra information (e.g. basketball trajectory or player action). This semantic gap typically cannot be solved in weakly-supervised approaches.

More Classification results are shown on our website.[2]

F. QUALITY OF GAN-AUGMENTED TRAJECTORY

The original size of a basketball court in our test image is 348 × 326, and for a GAN started with the random initialization, root mean squared error (RMSE) is set about 126 pixels. After the process of hyperparameter tuning, RMSE will drop to 23 pixels. Figure 8 shows a comparison between GAN augmented trajectories and the template generated from spatial-temporal clustering [6]. We can see that the GAN augmented data does fall within the permitted range defined by the template.

V. CONCLUSION

Group activity recognition is a difficult but strongly demanded topic. In this paper, based on the concept of key-player-based tactic classification, we propose an end-to-end trainable neural network to automatically learn players' dynamic features. To overcome pace variations of players' trajectories, global average pooling (GAP) is applied. GAP also reveals an activation map on the time axis of each key players' trajectories, which allows experts to study the temporal pattern of each key players' subgroup. By Adopting deep neural networks, our approach significantly increases both tactic and key player accuracy without prerequisite knowledge on the basketball field, which generalizes the system to other group activation recognition applications. Furthermore, to solve the problem of insufficient raw data, we also design a GAN that can generate group tactical behavior as augmented data. Like Key-Player-Based group activity recognition Network, our GTRCGAN can also be used to simulate the behavior of other group activities. In the future, we will evaluate our model on other applications and datasets such as those of surveillance or different group sports.

[2] https://sites.google.com/view/ieee-access-2021/

REFERENCES

[1] T.-Y. Tsai, Y.-Y. Lin, H.-Y.-M. Liao, and S.-K. Jeng, "Recognizing offensive tactics in broadcast basketball videos via key player detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 880–884.

[2] S. S. Intille and A. F. Bobick, "A framework for recognizing multi-agent action from visual evidence," in *Proc. AAAI/IAAI*, 1999, vol. 99, nos. 518–525, p. 2.

[3] B. Siddiquie, Y. Yacoob, and L. Davis, "Recognizing plays in American football videos," Univ. Maryland, College Park, MD, USA, Tech. Rep. 111, 2009.

[4] M. Perše, M. Kristan, S. Kovačič, G. Vučkovič, and J. Perš, "A trajectory-based analysis of coordinated team activity in a basketball game," *Comput. Vis. Image Understand.*, vol. 113, no. 5, pp. 612–621, May 2009.

[5] A. Bialkowski, P. Lucey, P. Carr, S. Denman, I. Matthews, and S. Sridharan, "Recognising team activities from noisy data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 984–990.

[6] C.-H. Chen, T.-L. Liu, Y.-S. Wang, H.-K. Chu, N. C. Tang, and H.-Y.-M. Liao, "Spatio-temporal learning of basketball offensive strategies," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 1123–1126.

[7] Y. Zhong, "Learning person trajectory features for sports video analysis," Ph.D. dissertation, School Comput. Sci. Appl. Sci., 2017.

[8] N. Mehrasa, Y. Zhong, F. Tung, L. Bornn, and G. Mori, "Learning person trajectory representations for team activity analysis," 2017, *arXiv:1706.00893*. [Online]. Available: http://arxiv.org/abs/1706.00893

[9] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1971–1980.

[10] V. Ramanathan, J. Huang, S. Abu-El-Haija, A. Gorban, K. Murphy, and L. Fei-Fei, "Detecting events and key actors in multi-person videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3043–3053.

[11] M. Qi, J. Qin, A. Li, Y. Wang, J. Luo, and L. V. Gool, "StagNet: An attentive semantic RNN for group activity recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Dec. 2018, pp. 101–117.

[12] J. Wu, L. Wang, L. Wang, J. Guo, and G. Wu, "Learning actor relation graphs for group activity recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9964–9974.

[13] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[15] P. Nguyen, B. Han, T. Liu, and G. Prasad, "Weakly supervised action localization by sparse temporal pooling network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6752–6761.

[16] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.

[17] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artif. Intell.*, vol. 89, nos. 1–2, pp. 31–71, 1997.

[18] O. Maron and A. L. Ratan, "Multiple-instance learning for natural scene classification," in *Proc. ICML*, vol. 98, 1998, pp. 341–349.

[19] C. Yang and T. Lozano-Perez, "Image database retrieval with multiple-instance learning techniques," in *Proc. 16th Int. Conf. Data Eng.*, 2000, pp. 233–243.

[20] D. Kotzias, M. Denil, N. de Freitas, and P. Smyth, "From group to individual labels using deep features," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 597–606.

[21] K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang, "Augmented multiple instance regression for inferring object contours in bounding boxes," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1722–1736, Apr. 2014.

[22] F.-J. Chang, Y.-Y. Lin, and K.-J. Hsu, "Multiple structured-instance learning for semantic segmentation with uncertain training data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 360–367.

[23] Z.-H. Zhou and M.-L. Zhang, "Neural networks for multi-instance learning," in *Proc. Int. Conf. Intell. Inf. Technol.*, Beijing, China, 2002, pp. 455–459.

[24] X. Wang, Y. Yan, P. Tang, X. Bai, and W. Liu, "Revisiting multiple instance neural networks," *Pattern Recognit.*, vol. 74, pp. 15–24, Feb. 2018.

[25] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," 2018, *arXiv:1802.04712*. [Online]. Available: http://arxiv.org/abs/1802.04712

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[27] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[28] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[29] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2009, pp. 448–455.

[30] A. Hyvärinen, J. Hurri, and P. O. Hoyer, "Estimation of non-normalized statistical models," in *Natural Image Statistics*. Springer, 2009, pp. 419–426.

[31] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 297–304.

[32] F. Henrique Kiyoiti dos Santos Tanaka and C. Aranha, "Data augmentation using GANs," 2019, *arXiv:1904.09135*. [Online]. Available: http://arxiv.org/abs/1904.09135

[33] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," 2017, *arXiv:1711.04340*. [Online]. Available: http://arxiv.org/abs/1711.04340

[34] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, "BAGAN: Data augmentation with balancing GAN," 2018, *arXiv:1803.09655*. [Online]. Available: http://arxiv.org/abs/1803.09655

[35] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2255–2264.

[36] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "GD-GAN: Generative adversarial networks for trajectory prediction and group detection in crowds," in *Proc. Asian Conf. Comput. Vis.* Springer, 2018, pp. 314–330.

[37] J. Amirian, W. van Toll, J.-B. Hayet, and J. Pettré, "Data-driven crowd simulation with generative adversarial networks," in *Proc. 32nd Int. Conf. Comput. Animation Social Agents*, Jul. 2019, pp. 7–10.

[38] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

TSUNG-YU TSAI received the B.S. degree in electrical engineering and the M.S. degree in communication engineering from National Taiwan University, Taipei, Taiwan, in 2008 and 2011, respectively, where he is currently pursuing the Ph.D. degree in communication engineering.

From 2014 to 2018, he was a Research Assistant with the Institute of Information Science, Academia Sinica, Taipei. His research interests include computer vision, machine learning, and artificial intelligence for group activity analysis.

YEN-YU LIN (Member, IEEE) received the B.B.A. degree in information management and the M.S. and Ph.D. degrees in computer science and information engineering from National Taiwan University, in 2001, 2003, and 2010, respectively. He is currently a Professor with the Department of Computer Science, National Chiao Tung University. Prior to that, he worked for the Research Center for Information Technology Innovation, Academia Sinica, from January 2011 to July 2019. His current research interests include computer vision, machine learning, and artificial intelligence.

SHYH-KANG JENG (Senior Member, IEEE) received the B.S.E.E. and Ph.D. degrees from National Taiwan University, Taipei, Taiwan, in 1979 and 1983, respectively. He is currently a Professor with National Taiwan University. In 1981, he joined the Department of Electrical Engineering, National Taiwan University, as a Faculty Member. From 1985 to 1993, he was a Visiting Research Associate Professor and a Visiting Research Professor with the University of Illinois at Urbana–Champaign. In 1999, he visited the Center for Computer Research in Music and Acoustics, Stanford University, Stanford, CA, USA, for six months. His research interests include theory and applications of electromagnetic and acoustics field, multimedia signal processing, machine learning, and computational cognitive neuroscience. He received the 1998 Outstanding Research Award from the National Science Council, Taiwan, and the 2004 Outstanding Teaching Award from National Taiwan University.

HONG-YUAN MARK LIAO (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Northwestern University, in 1990. In July 1991, he joined the Institute of Information Science, Academia Sinica, Taiwan, where he is currently a Distinguished Research Fellow and the Director. He has worked in the fields of multimedia signal processing, computer vision, pattern recognition, multimedia protection, and artificial intelligence for more than 30 years. He is also jointly appointed as an Honorary Chair Professor of National Chiao Tung University. He received the Young Investigators' Award from Academia Sinica, in 1998; the Distinguished Research Award from the National Science Council, in 2003, 2010, and 2013; the Academia Sinica Investigator Award, in 2010; and the TECO Award from the TECO Foundation, in 2016. His professional activities include the President of Image Processing and Pattern Recognition Society, Taiwan, from 2006 to 2008; an Editorial Board Member of *IEEE Signal Processing Magazine*, from 2010 to 2013, and *ACM Computing Surveys*, since 2018; and an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, from 2009 to 2013, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, from 2009 to 2012, and IEEE TRANSACTIONS ON MULTIMEDIA, from 1998 to 2001.

• • •