Football Action Recognition using Hierarchical LSTM

Takamasa Tsunoda
Canon Inc.
tsunoda.takamasa@canon.co.jp

Yasuhiro Komori
Canon Inc.
komori.yasuhiro@canon.co.jp

Masakazu Matsugu
Canon Inc.
matsugu.masakazu@canon.co.jp

Tatsuya Harada
The University of Tokyo
harada@mi.t.u-tokyo.ac.jp

Abstract

We present a hierarchical recurrent network for understanding team sports activity in image and location sequences. In the hierarchical model, we integrate proposed multiple person-centered features over a temporal sequence based on LSTM's outputs. To achieve this scheme, we introduce the Keeping state in LSTM as one of externally controllable states, and extend the Hierarchical LSTMs to include mechanism for the integration. Experimental results demonstrate effectiveness of the proposed framework involving hierarchical LSTM and person-centered feature. In this study, we demonstrate improvement over the reference model [4] in two-stream LSTM based approach. Specifically, by incorporating the person-centered feature with meta-information (e.g., location data) in our proposed late fusion framework, we also demonstrate increased discriminability of action categories and enhanced robustness against fluctuation in the number of observed players.

1. Introduction

Activity recognition is a challenging task which has received a significant amount of attention in the computer vision field. In this work, we focus on the team sports activity involving multiple persons in team sports videos. Generally, team sports videos are composed of person images that have similar attributes, *e.g.*, age, gender, clothes, *etc.*; the recording scenes also don't change much. Therefore, it is necessary to purely understand activities rather than recognizing activities by background clutters among many sports, such like UCF101 [30].

In this work, we address the problem of play type recognition of "soccer" games (*e.g.*, futsal) in multi-view-videos obtained from multiple camera system. We also address the effective use of meta-information, the three dimensional location of players and ball in the activity recognition. We

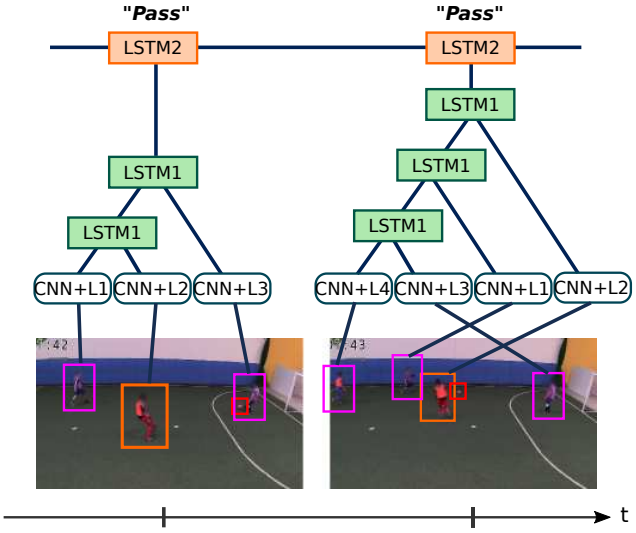

Figure 1. Our model is composed of CNN and two-layer LSTMs *i.e.* CNN+L*k*, LSTM1 and LSTM2, where CNN+L*k* denotes a person-centered feature. The person-centered feature is a concatenated feature of the CNN feature and meta-information (*e.g.,* person location, ball location, *etc*). LSTM1 recurrently integrates a variable number of person-centered features based on a distance between a ball and each person, and LSTM2 integrates a temporal sequence of integrated multiple person features. Finally, our model produces a futsal activity prediction.

prepared datasets of two games that consist of videos obtained from the multiple cameras and ground truth data of action categories (*e.g.*, *Pass*, *Shoot* and *Dribble*). The multi-view videos from a set of calibrated cameras can provide the 3D information of a ball and players, and we exploit the use of such meta-information to improve the recognition accuracy. The action or play-type categories are exclusively set in the given futsal games. It is sometimes difficult to distinguish *(long) Pass* from *Shoot* for a limited number of frames. In this study, we focus on such play type or actions

that involve at least 2 players (*e.g.*, kicker and receiver). As each camera has a finite field-of-view (FoV), the number of visible persons and the ball is variable at every moment (see Figure 1).

In this paper, we propose a new method for recognizing futsal plays that exploits integrating multiple person-centered features and temporal dynamics of features for multiple persons in person image sequences based on the hierarchical Long Short-Term Memories (LSTMs) [13, 8] model (see Figure 1). Here, the person-centered feature is composed of Convolutional neural network (CNN) [18, 23] features and meta-information, *e.g.*, person location, ball location, *etc*. The proposed network as shown in Figure 1 has a hierarchical structure composed of CNN, LSTM1 and LSTM2. The first LSTM (*i.e.* LSTM1) recurrently integrates a variable number of person-centered features. The second LSTM (*i.e.* LSTM2) integrates over a temporal sequence of integrated multiple person features. To achieve this mechanism, we introduce *Keeping state* in LSTM, which is an externally controllable state, and we actively switch *Keeping state* as well as *Reset state* which is usually used at only initial moment of sequences [37].

The main contribution of this paper is the proposal of a novel deep hierarchical structure incorporating multiple LSTMs and switching states. Moreover, we demonstrate the effectiveness of location information of players to recognize the plays in team sports matches. We evaluate our method using RGB image sequences of videos as well as optical flow image sequences. We compare our method with a previously proposed baseline method [4], and our proposed model improves group activity recognition performance compared to the baseline.

The rest of the paper is organized as follows. In Section 2, we review the related work on action recognition. In Section 3, we introduce the proposed method. Implementation details are given in Section 3.3 and the performance is evaluated and compared to the baseline method in Section 4. A conclusion and future work are discussed in Section 5.

2. Related Work

Recently, a number of pattern recognition fields have developed with a focus on deep learning techniques. Although deep learning also plays a central role in activity recognition, existing methods employing hand-crafted features have much of a presence. H. Wang and C. Schmid proposed improved Dense Trajectories [34] which is one of the most widely used [7, 6] trajectory-based hand-crafted representations. In this method, the representations of local descriptors, HOG [3], HOF [19], and MBH [33], are extracted from dense trajectories computed using optical flow. Furthermore, Fisher vector encoding [25] is applied to the representations.

Two-steam approaches are one of the dominant deep learning based action recognition frameworks using RGB and optical flow. L. Simonyan and A. Zisserman [27] achieve results comparable to [34]. J. Y-H. Ng et al. [24] apply LSTM to the two stream approach to capture the temporal structure of actions, B. Singh et al. [29] proposed a multi-stream framework full-frame and person-centric of RGB and flow, followed by bi-directional LSTM. C. Feichtenhofer et al. [7] proposed multiple fusion approaches for RGB and flow stream on a middle level feature map using several ways, *e.g,* 3 dimensional convolution, 1×1 convolution, *etc*. C. Feichtenhofer et al. [6] also proposed another fusion approach using inter-connection of 2 stream residual networks [11]. B. Zhang et al. [39] overcome a bottleneck of computational cost related to optical flow.

LSTM has various application fields, *e.g.,* speech recognition [9], machine translation [31], image description [32], natural language retrieval [14]. J. Donahue et al. [4] demonstrate availability of their monolithic network composed of CNN and LSTM for multiple applications, *i.e.,* image description, video description as well as activity recognition. In this paper, we introduce an externally controllable state in LSTM, called *Keeping state*. Our *Keeping state* operation is similar to "inactivation" for the multi-timescale updates of LSTM in Y. Wang et al., [35] and P. Liu et al., [21]. However, there has been no similar schemes like the proposed integration of person-centered features with the LSTM representation of temporal dynamics on top of the active control mechanism of LSTM's state.

In the domain of activity recognition of sports, there are a number of previous works to handle multiple objects (*i.e.,* players) interacting with each other. S. Chen et al. [2] proposed a play type recognition method for American football games. A. Maksai et al. [22] realize a unified framework to track a ball and to estimate the ball state, *i.e., flying, hit or possession* by a graphical model. They demonstrate the use of the framework on multiple public video datasets containing basketball, volley ball and soccer. The most closely related work to ours is M. S. Ibrahim et al. [15]. They proposed a hierarchical model composed of a CNN and a two layer LSTM, and recognize person-level activity and group activity simultaneously. Their method integrates multiple person-level activity with pooling, which is different from our approach. T. Bagautdinov et al. [1] proposed an end-to-end approach to perform person tracking and group activity detection on volley ball video.

Recent surveys by S. Herath et al. [12] and S. M. Kang et al. [17] cover vast literature in activity recognition. The former introduces a number of learning methods in the activity recognition, and the latter introduces the benchmark datasets.

3. Method

Our goal in this paper is to recognize futsal plays that involve multiple players in futsal matches, using frame sequences as well as meta-information (3 dimensional locations of persons and a ball, a camera location and team ID, automatically detected). Therefore, the input of our method comprises two parts, (1) a temporal sequence of cropped images extracted from a person's bounding boxes and (2) temporal meta-information (3 dimensional locations *etc.*) sequences for each player (See Section 4 for details). In this section, we will describe how to handle these two data parts and recognize futsal plays in the following two aspects:

- **Person-centered feature** aggregating lower and higher level CNN (Convolutional neural network) features as well as meta-information.
- **Hierarchical LSTMs** based on person-centered features and temporal dynamics integration.

Object locations are informative for understanding situations of team sports matches. The categories of group activities of team sports, *e.g., Pass, Shoot, etc.* in futsal, have a deep relationship with a player's location especially when in possession of the ball. Additionally, since in this case almost all appearances of a player's cropped images are similar in terms of foreground objects and background scene, the lower level features are more important than higher level features of CNN models pre-trained on the object detection task. Therefore, we try to aggregate the meta-information and multi-scale CNN features, including lower level features, as an individual person's feature.

Because each camera has a finite FoV and players continue to move at almost every moment, the total number of players captured by each camera varies every moment. As we mentioned above, the most important person is a player who is in possession of the ball. Hence, we try to integrate person-centered features centered upon the ball in a recursive fashion.

Inspired by the success of deep learning based solutions [18, 32, 31], in this paper, a novel hierarchical deep learning based model is proposed that is potentially capable of learning integration of multiple person-centered features and temporal dynamics in a unified end-to-end framework. Next, we describe the details of our network.

3.1. Person-centered feature

In the group activity recognition for team sports videos, recognition targets have similar appearance because targets are always human (and a ball), recording domains are nearly unchanged and target attributes are similar, *e.g.*, age, gender, clothes, *etc*. Hence, the higher level features extracted from CNN models pre-trained on the object detection task

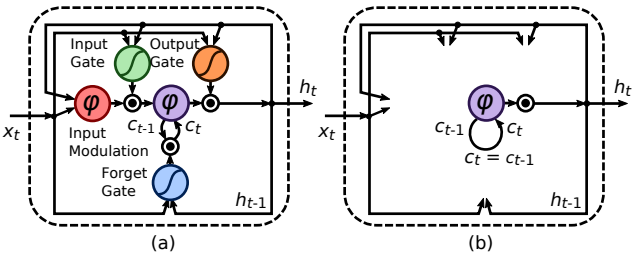

Figure 2. (a) LSTM, (b) LSTM of *Keeping State*

are not so informative for the current task. We think that the pose or parts locations are more effective than the output scores of pre-trained CNNs such as the 1000 category scores in the ImageNet [26].

We have two options for this notion: first we directly estimate poses by recently developed methods [36], second we extract information containing the parts' locations from a pre-trained model. It is a well-known fact that CNNs represent features from lower to higher level information in a layer-wise fashion, *e.g.,* blob, edge, pattern, shape, parts, object, *etc.* in the pre-trained model of ImageNet [38, 40]. Therefore, we consider that extracting the location of parts from the CNN lower layer is an effective approach. We extract the features of various layers (including the lower level layer), and we adjust their dimensionality using 1×1 convolution and pooling. We concatenate them, and utilize it as a CNN feature (See Section 3.3 for details). In addition, we consider that the meta-information (player location, ball location, team ID, *etc.*) is important to recognize the futsal plays. For instance, there is a high possibility of *Shoot* when a player close to the ball is located in the goal area. We utilize 3 dimensional locations of players and a ball, and a player's team ID. Moreover, we calculate the distances between the player location and the ball location, and between the player location and the camera location. We consider the meta-information mentioned above as a feature vector, and concatenate it with the CNN feature.

Finally, we employ this concatenated feature as the person-centered feature.

3.2. Hierarchical LSTMs

Given an input sequence $\boldsymbol{x} = (x_1, ..., x_T)$, a standard recurrent neural network (RNN) computes the hidden vector sequence $\boldsymbol{h} = (h_1, ..., h_T)$, and output vector sequence $\boldsymbol{y} = (y_1, ..., y_T)$ by iterating the following equations from $t = 1$ to T:

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

where the W terms denote weight matrices (e.g. W_{xh} is the input-hidden weight matrix), the b terms denote bias vec-

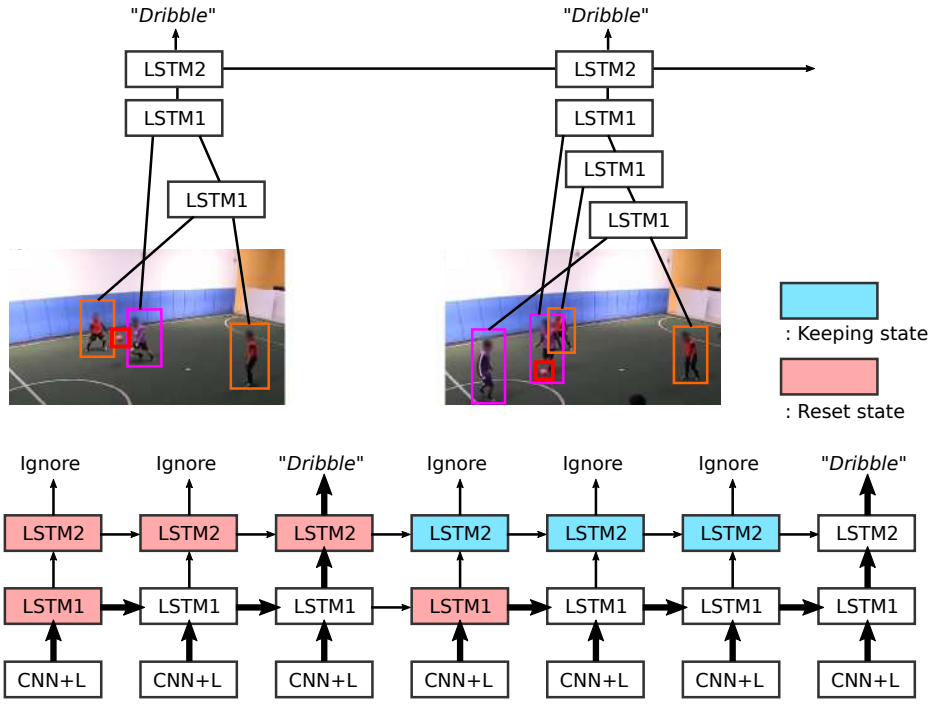

Figure 3. Illustrations for our method: *Upper*, An integration of appearance for two continuous frames, with three and four persons respectively. A first LSTM (LSTM1) recurrently integrates multiple person-centered features for persons near the ball. A second LSTM (LSTM2) integrates the integrated features from LSTM1. *Lower*, The switching aspect of the LSTM's states corresponding to the upper situation. Blue boxes denote the *Keeping state* and red boxes denote *Reset state*. We switch LSTM1 to *Reset state* on the initial moment of person-centered feature sequences each frame, and LSTM2 to *Reset state* until the end moment of the first person-centered feature sequence of the first frame. Subsequently, we switch LSTM2 to *Keeping state* during integration over the second person-centered feature sequence. A top line (consisting of Ignore and *Dribble*) is the sequence of GT or prediction labels. The third and seventh positions of the top line are the GT or the prediction label corresponding to each frame. Note, we ignore losses of an objective function and predictions except the tail positions corresponding to person-centered feature sequences each frame.

tors (e.g.b_h is hidden bias vector) and $\mathcal{H}$ is the hidden layer function.

$\mathcal{H}$ is usually an element-wise application of a sigmoid function. However, we found that the Long Short-Term Memory (LSTM) architecture [13, 8], which uses purpose-built memory cells to store information, is better at finding and exploiting long range context. Figure. 2 illustrates a single LSTM memory cell. For the version of LSTM used in this paper [4], $\mathcal{H}$ is implemented by the following composite function:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (4)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (5)$$
$$g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (6)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (7)$$
$$h_t = o_t \odot \tanh(c_t) \quad (8)$$

where σ is the logistic sigmoid function, $\odot$ represents the element-wise product with a gate value, and i, f, o, g and c are respectively the *input gate*, *forget gate*, *output gate*, *input modulation gate* and *cell* activation vectors.

To achieve embedding of the person-centered feature into the hierarchical LSTMs, we use the following two states that are implemented by external control for the activations of the gates.

Keeping state: plugging $f_t = 1$, $i_t = 0$ and $o_t = 1$ into Equations (7-8), they become:

$$c_t = c_{t-1} \quad (9)$$
$$h_t = \tanh(c_t) \quad (10)$$

This state realizes holding of the cell activity and the hidden vector of the LSTM (see Figure 2 (b)).

Reset state: plugging $f_t = 0$ into Equation (7), this state realizes flushing of a previous memory of the LSTM. This state is widely used when we initialize the hidden vector to zero at the initial moment of sequences only [37].

For a unified framework of both person-centered feature integration and activity recognition using a temporal feature sequence, we construct a hierarchical two-layer LSTM

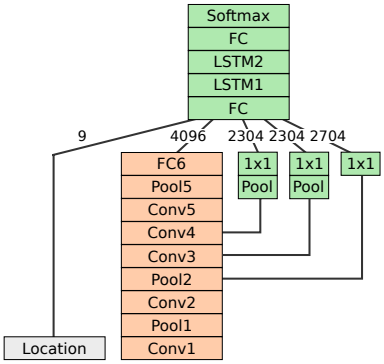

Figure 4. A detail of the proposed network structure.

Table 1. Number of samples in the dataset for each play label.

Label	Scene	View	Frame	Training	Test
Pass	334	1779	64959	1433	346
Dribble	177	1021	49509	865	156
Shoot	44	240	9573	205	35
Clearance	32	172	6159	-	-
LooseBall	12	69	2403	-	-
Total	599	3281	132603	2503	537

Note, *Scene* denotes number of occurrences about each label, *View* denotes number of samples when counting each camera view individually, *Frame* denotes number of frames about each label. We utilize *View* samples as training and test data.

structure. In this structure, the first layer LSTM (hereinafter called LSTM1) plays the role of the person-centered feature integration and the second LSTM (hereinafter called LSTM2) plays the role of activity recognition.

Suppose we got a sequence as a flattened nested array of a person-centered feature sequence. We set *Reset state* to LSTM1 on an initial moment of the person-centered feature sequence of each frame, and we set *Reset state* to LSTM2 from the initial moment of the first frame to the end moment of the first frame. We also set *Keeping state* to LSTM2 during integrating the person-centered features in LSTM1 except the first frame. In addition, we ignore predictions and losses of an objective function except when the feature integration is done. We illustrate this switching aspect in Figure 3.

3.3. Implementation Details

The CNN used in our model is the Caffe reference model [16] (a minor variant of AlexNet [18]). The input image to the CNN is cropped from each player bounding box with a 4 meters $\times$ 4 meters region centered around the player on the 3 dimensional space, and further resized it to 240×240. To obtain the fixed-size 227×227 CNN input images [18] (RGB as well as flow, as mentioned below), they are randomly cropped from the resized player bounding box images in the training phase (one crop per image per SGD iteration), and centrally cropped in the test phase. We extract lower features from a second pooling layer, a third convolution layer and a forth convolution layer of the CNN, respectively. These features are also processed by a pooling and/or a 1×1 convolution layer to reduce and homogenize the dimension (*i.e.*, height, width and channels) of each feature map, and the features from the second pooling layer, the third convolution layer and the forth convolution layer are 2704, 2304 and 2304 dimension, respectively. On the other hand, the meta-information is a 9 dimensional vector composed of 3 dimensional coordinates (X, Y and Z) normalized to $[0, 1]$ for (1-3) each player and (4-6) the ball, (7) a team ID corresponding to each color of team bibs (0/1 binary), (8) a scaled distance between each player and the ball and (9) a scaled distance between each player and the camera. Both distances, (8) and (9), are calculated from the player's location, the ball location and the fixed-camera location. We concatenate the features consisting of the location data, the dimensionality reduction lower level features of the CNN, as well as a first inner-product layer's feature of the CNN. We then get the 1024 dimensional feature by an inner-product layer. Finally, we get the score through the LSTM1, LSTM2, the final inner-product layer and the softmax layer. Figure 4 shows the entire network structure. We implement our model as a monolithic network using the Caffe framework [16]. Consequently, it is possible to train in an end-to-end fashion.

We consider both RGB image and optical flow as the CNN input. Optical flow is computed with [5] from the original (non-cropped) images and transformed into a "flow image" by centering x and y flow values around 128 and multiplying by a scaling factor such that flow values are normalized to $[0, 255]$. A third channel for the flow image is created by calculating the flow magnitude. We will explore the late fusion of both the softmax scores with the RGB image and the flow image by fusing the RGB and the flow stream scores as weighted average [4]. Sub-optimal weights are experimentally found as shown in Section 4.3.

4. Experimental Result

In this section, we first present the details of the dataset and the evaluation protocol. Then, we describe the details of the training procedure. Finally, we present the experimental results with discussions.

4.1. Dataset

The evaluation is conducted on a multi-camera futsal game dataset. This dataset consists of two 10 minutes matches recorded by 14 cameras approximately synchro-

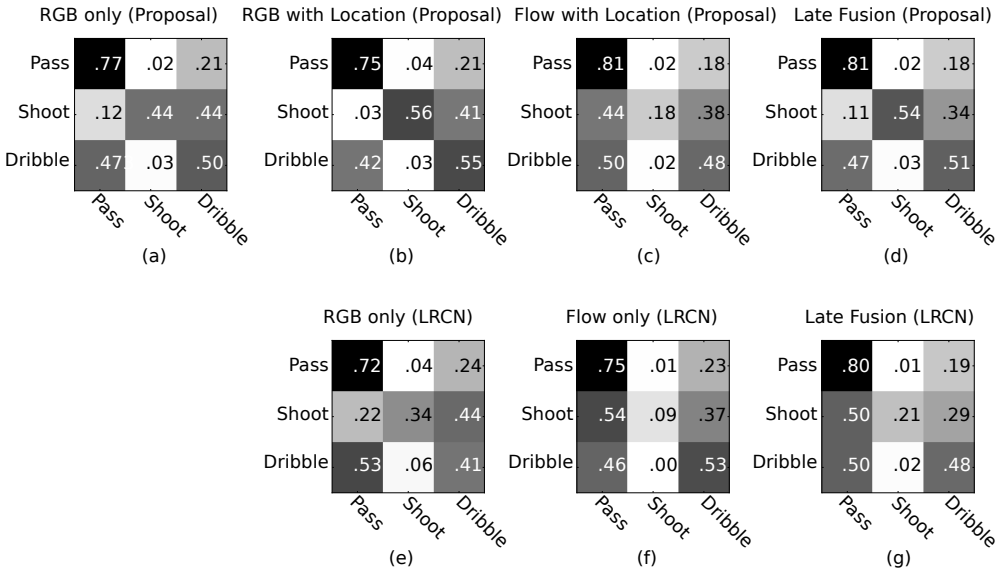

Figure 5. Confusion matrix. (a) Proposed method using RGB image without location data (Accuracy: 0.671, F1 score: 0.587), (b) with RGB and location data (Accuracy: 0.681, F1 score: 0.612), (c) with Optical flow and location data (Accuracy: 0.673, F1 score: 0.507), (d) Late fusion of (b) and (c) (Accuracy: 0.702, F1 score: 0.634, fusion method is weighted averaging (RGB : Flow = 0.5 : 0.5)), (e) LRCN [4] using RGB image only (Accuracy: 0.606, F1 score: 0.486), (f) LRCN [4] using Optical flow only (Accuracy: 0.646, F1 score: 0.462), (g) Late fusion of (f) and (g) (Accuracy: 0.671, F1 score: 0.518, fusion method is weighted averaging (RGB : Flow = 0.4 : 0.6)). Note, the weights for the late fusions, (d) and (g), are optimal values over various trials using the test dataset.

nized and calibrated. Therefore, each match has a set of 14 video streams, which are captured by 30-frame-per-second Full HD (1920×1080) cameras which are placed around the court. Additionally, a ball and a person tracker is applied to each video [20], and 3 dimensional locations of each object are obtained using triangulation under the epipolar constraint [10]. Note, the 3 dimensional locations of these objects contain mistakes; we didn't manually correct. We manually annotated ground truth, *Pass*, *Dribble*, *Shoot*, *Clearance* and *Loose ball*, for every frame of two matches. These labels are exclusive and occurred at least once during the futsal match. And these annotated samples have various durations.

Considering visibility of a ball, we remove following inappropriate samples using the 3 dimensional locations of the ball detection result, (1) 'far' samples from a camera position (*i.e.* over 18 meters, approximately corresponds to the distance from the center line to the goal line of the court), and (2) 'invisible' samples in which the ball is not captured from a camera FoV. We acquired samples as shown in Table 1, where *Scene* denotes number of occurrences about each label, *View* denotes number of samples when counting each camera view individually, *Frame* denotes number of frames about each label. In this work, we utilize the *View* as training and test samples.

We did a 5-fold split of the samples along the timeline of each match. We use a third for test data and the rest for training data. The last two columns of Table 1 show amounts of the training and the test samples. Here, we ignore the *Clearance* and the *Loose ball* samples due to the low number of samples for these labels. However, we did incorporate the *Shoot* samples despite the low number of samples, because generally *Shoot* is the most important action in football.

4.2. Experimental setting

We compared our model with a baseline proposed previously (hereinafter, referred to as baseline or LRCN) [4]. Similar to other approaches [4, 15], we initialize the CNN weights with the pre-trained models for an object classification task ILSVRC-2012 [26] in the case of RGB image input, and the ILSVRC-2012 as well as an action recognition task UCF-101 [30] in the case of flow image input. We fine-tune the whole network except the base CNN indicated in orange blocks in Figure 4 (green blocks are fine-tuned). The image pre-processing for our method have already mentioned in Section 3.3. In the case of the baseline, the original images (RGB as well as flow image) are resized from 1920×1080 to 320×240, and randomly cropped from the resized images for the CNN input in the training phase, and centrally cropped in the test phase. The network training procedure generally follows [4]. Namely, the training is carried out by optimizing the softmax regression objective using mini-batch gradient descent with momentum.

The batch sizes are set to 24 in the baseline and to 8 in our method. We use a base learning rate of 0.001 and a momentum of 0.9. The learning rate is dropped in 30,000 steps with $\gamma = 0.1$.

The time steps of the back-propagation through time (BPTT) are 16 frames in the baseline model. Although we also use 16 frames in a single sample in our model, we simultaneously set the maximum number of the person-centered features to five. Therefore, the time steps of BPTT are 80 steps in our model. The maximum number of person-centered features processed in one frame is a controllable parameter. Although there are 10 players in futsal games, as the time steps of one sample are excessively long, we set the maximum number of players to five. We examine less than five cases in the second experiment mentioned in the next section. We train the baseline and our network with continuous 16 frames randomly extracted from the play label samples. Because there is data imbalance in our dataset as shown in Table 1, we oversampled to equalize occurrence frequency of each label in the training phase. Both methods predict the play label at each frame and we average these predictions for final classification. At the test phase, we extract 16 frames with a stride of 8 frames from each play label sample and average across duration of each sample.

4.3. Results

Here we show several comparative experiments: (1) a comparison between several input conditions and the baseline, (2) a comparison of several conditions related to the maximum number of players features.

We first evaluate classification accuracy of the baseline and the proposed method using various inputs conditions: (i) RGB image only for the baseline and our method, (ii) RGB image with meta-information for our method, (iii) flow image with meta-information for our method, (iv) late fusion of network scores of the RGB image with meta-information and the flow image with meta-information for our method, (v) flow image without meta-information for the baseline and (vi) late fusion of network scores of the RGB image without meta-information and the flow image without meta-information for the baseline. The results are shown in Figure 5. From the figure, we can make the following observations: The proposed method with RGB image and meta-information inputs (Figure 5 (b)) outperforms the proposed method without meta-information input (Figure 5 (a)), and meta-information helps decreasing confusion of *Dribble* and *Pass* as well as *Dribble* and *Shoot*. The proposed method of RGB without meta-information input (Figure 5 (a)) outperforms the baseline of RGB without meta-information input (Figure 5 (e)). The late fusion of the proposed model is the best score including the baseline, *i.e.*, 1 percentage point improvement from the baseline in *Pass*, 33 percentage points improvement in *Shoot*, 3 percentage points improvement in total. However, it remains difficult to distinguish *Dribble* and *Pass*. Inspecting the actual test samples provides a reason for this problem. Figure 6 shows the test samples of the recognition results. Figure 6 (a) is one of *Pass* samples recognized correctly on the late fusion of the proposed model (iv): it has a long distance and duration from a passer to a receiver. There are many correctly recognized samples having these properties. On the other hand, Figure 6 (b) is one of the *Pass* samples recognized incorrectly by the proposed method (iv). It has a short distance and duration from a passer to a receiver. Because there are many kick-actions like this during the *Dribble* action, we think our late fusion method confuses *Dribble* and *Pass*. Figure 6 (c) is one of the *Pass* samples whose recognition is significantly improved by the RGB image with meta-information (ii) against the RGB without meta-information (i). This is the *Kick-in* action when restarting the game from off-play situations. Although a number of *Kick-in* samples are misclassified by the RGB without meta-information (i), they are correctly recognized by the RGB image with meta-information (ii). We observed that the recognition of these samples improved by utilizing the meta-information, especially the player coordinates on the court.

Table 2. Accuracy and F1 score for values of the maximum number of persons.

Max num.	Accuracy	F1 score
1 person	0.689	0.580
2 persons	0.686	0.610
3 persons	0.680	0.604
4 persons	**0.709**	0.604
5 persons	0.702	**0.634**

Secondly, we evaluate our approach under various conditions of the maximum number of person-centered features handled in one frame. Table 2 shows the experimental results that vary the maximum number from one to five persons. The results show that accuracy is higher when the maximum number is large. Note that, under the condition handling only one person at one frame, the LSTM1 is functionally equivalent to a feedforward network consisting of some inner-products and activation functions. Therefore, this result shows the evidence for the effectiveness of multiple person-centered features integration. In other words, the accuracy of integrating multiple person-centered features is better than that of the case only focusing on single person for play type recognition on team sports videos.

5. Conclusion and Future Work

In this paper, we propose a hierarchical model integrating multiple person-centered features and temporal dynam-

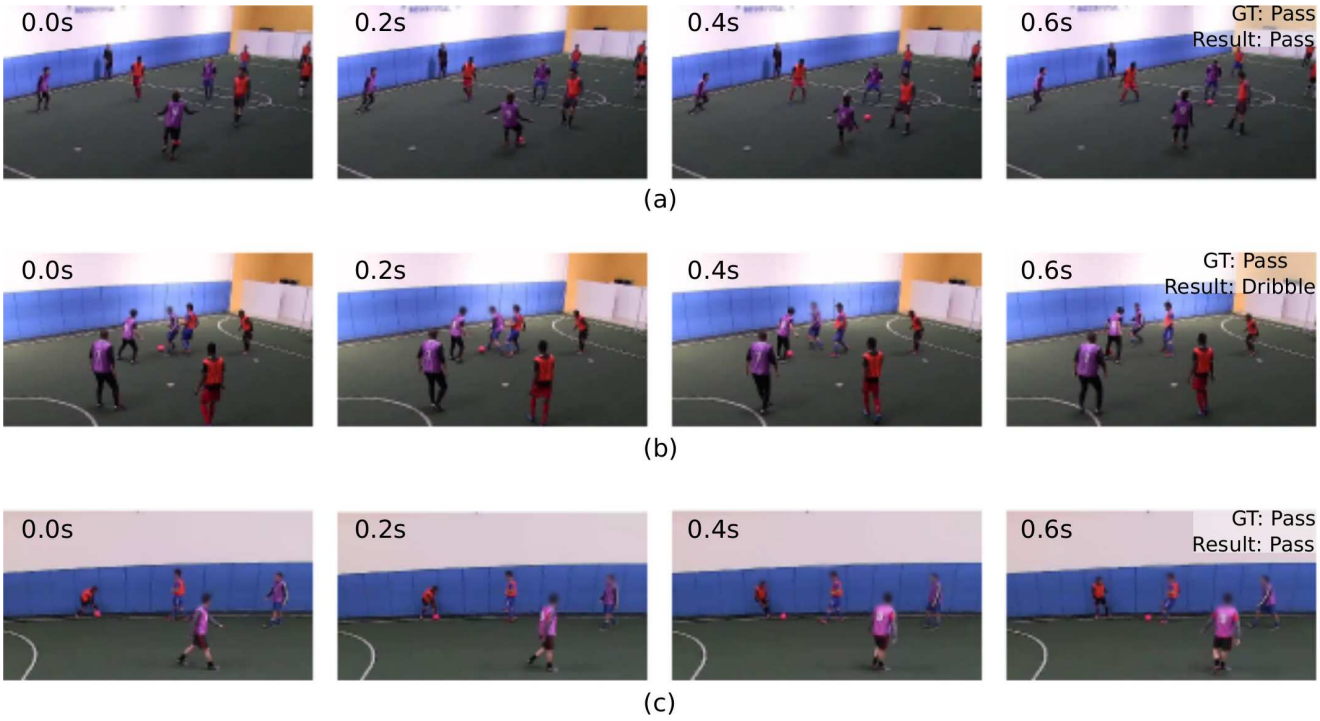

Figure 6. Results: (a) Successfully classified *Pass* by the proposed model (late fusion). (b) *Pass* misclassified as *Dribble* by the proposed model (late fusion). (c) One of the "Kick-in" (*i.e. Pass*) action samples. This is successfully classified *Pass* by the RGB image with meta-information condition. A number of the "Kick-in" samples are successfully classified by same condition, although they are misclassified by the RGB image without meta-information condition.

ics for group activity recognition. The experimental results suggest that the proposed framework outperforms the two-stream with LSTM approach using holistic images [4].

These results further support that (1) the object location information for futsal activity recognition is effective and (2) the approach of recurrently integrating multiple person-centered features is more accurate than the approach of using a single person-centered feature only.

We utilized a relatively small scale CNN [18] in this paper. Our approach is also available in the case of utilizing another deep scale CNN, *e.g.*, VGG [28] and residual network [11]. We believe our method will effectively improve the performance for team sports activity recognition.

Although our dataset was recorded by calibrated multi-cameras, we did not utilize this valuable information in this study. We will further investigate how to integrate features from multiple cameras in near future.

References

[1] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, and S. Savarese. Social scene understanding: end-to-end multi-person action localization and collective activity recognition, 2016.

[2] S. Chen, Z. Feng, Q. Lu, B. Mahasseni, and T. Fiez. Play type recognition in real-world football video, 2014.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014.

[5] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis*, LNCS 2749, pages 363–370, Gothenburg, Sweden, June-July 2003.

[6] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal residual networks for video action recognition, 2016.

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition, 2016.

[8] A. Graves. Generating sequences with recurrent neural networks. *arXiv:1308.0850*, 2013.

[9] A. Graves, A. r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks, 2013.

[10] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, second edition, 2003.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[12] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21, 2017.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[14] R. Hu, H. Xu, M. Rohrbach, J. Feng, K. Saenko, and T. Darrel. Natual language object retrieval. In *CVPR*, 2016.

[15] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori. A hierarchical deep temporal model for group activity recognition. In *CVPR*, 2016.

[16] Y. Jia. Caffe: An open source convolutional architecture or fast feature embedding, 2013. http://caffe.berkeleyvision.org/.

[17] S. M. Kang and R. P. Wilders. Review of action recognition and detection methods. *arXiv preprint arXive:1610:06906*, 2016.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[19] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[20] M. Li, Z. Zhang, K. Huang, and T. Tan. Rapid and robust human detection and tracking based on omega-shape features, 2009.

[21] P. Liu, X. Qiu, X. Chen, S. Wu, and X. Huang. Multi-timescale long short-term memory neural network for modeling sentences and documents. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2326–2335, 2015.

[22] A. Maksai, X. Wang, and P. Fua. What players do with the ball: a physically constrained interaction modeling, 2016.

[23] M. Matsugu, K. Mori, Y. Mitarai, and Y. Kaneda. Subject independent facial expression with robust face detection using a convolutional neural network. *Neural Networks*, 16(5).

[24] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: deep networks for video classification, 2015.

[25] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification, 2010.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. M. adn Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[27] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos, 2014.

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[29] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *CVPR*, 2016.

[30] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild, 2012. CRCV-TR-12-01.

[31] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[32] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1441.4555*, 2014.

[33] H. Wang, A. Kläser, C. Schmid, and C. L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International journal of computer vision*, 103(1):60–79, 2013.

[34] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[35] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li. Hierarchical attention network for action recognition in videos. *arXiv preprint arXiv:1607.06416*, 2016.

[36] S. E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[37] W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization, 2015.

[38] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR, abs/1311.2901v3*, 2013.

[39] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns, 2016.

[40] B. Zhou, A. Khosla, A. Lapedrize, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *ICLR*, 2015.