Watch and Act: Dual Interacting Agents for Automatic Generation of Possession Statistics in Soccer

Saikat Sarkar
University of Calcutta
Kolkata, India
to.saikatsarkar17@gmail.com

Dipti Prasad Mukherjee
Indian Statistical Institute
Kolkata, India
dipti@isical.ac.in

Amlan Chakrabarti
University of Calcutta
Kolkata, India
acakcs@caluniv.ac.in

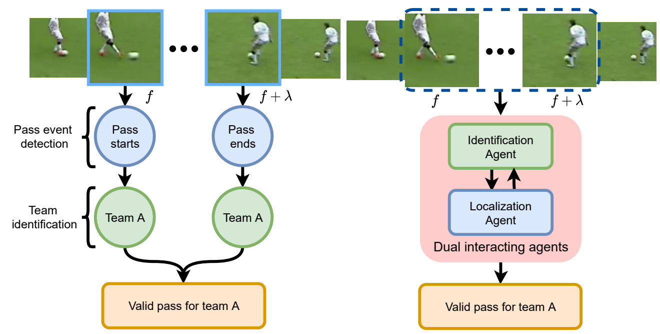
Abstract

Pass localization and team identification are two primary tasks for pass-count based possession statistics generation of a soccer match. While the existing works perform these two tasks separately, we propose dual interacting reinforcement learning agents to jointly perform these tasks. The proposed model has a localization agent, that decides which direction to move a temporal window to localize a pass. On the other hand, there is an identification agent that decides if the temporal window contains a pass for team-A (or team-B), or the localization agent needs to readjust the temporal window further. In this multi-agent setup, an agent may communicate by sharing some message to guide the other agent to achieve its task. To achieve this inter-agent communication, we extend the Dueling DQN architecture and share the value of a state as a message to the other agent. Two agents watch, act independently and cooperate with each other in order to detect a valid pass in a soccer video. A novel reward function is proposed that helps the agents to learn the optimal policy. Experiments performed on online videos show that our method is 3% better at localization of pass than the competitive methods.

1. Introduction

The ball possession statistics is one of the key determining factors that differentiates a winning team from a losing team. A recent study [4] on the impact of ball possession statistics on match outcome in the UEFA Champions League (from 2014 to 2019) reveals that teams with higher ball possession won 49.25%, draw 22.04%, and lost 28.71% of the matches.

Pass-count based approach is one of the most popular methods to calculate the possession statistics of two teams. In this method, the possession stats of team-A is calculated by dividing the number of valid passes of team-A with the



(a) Existing method (b) The proposed method

Figure 1. Comparison between existing and proposed methods.

total number of valid passes of both teams, as follows,

$$Possession(team\ A) = \frac{passA}{passA + passB}, \quad (1)$$

where $passA$ and $passB$ are valid pass counts for team-A and team-B respectively. The numerator of (1) is replaced by $passB$ for calculation of possession stat for team-B. A valid pass for a team (say team-A) is defined as the passing of the ball between two players of that team (team-A).

The main challenge in the pass-count based method is how to detect a valid pass in a soccer video. The existing works [16–18] solve the problem of detection of a valid pass in two steps. First, a pass-start event (a player passing the ball) at frame f and a pass-end event (another player receiving the ball) at frame $(f + \lambda)$ are detected. Then, a pass is recognized from the frame f to $(f + \lambda)$ as a pass-start followed by a pass-end event. In the next step, the team information of the passer and receiver are analyzed based on jersey color and a valid pass is recognized if two players are of the same team. The existing approach of pass detection is shown in Fig. 1a.

Note that the existing approaches have detected passes by detecting pass events (pass-start and pass-end). Therefore, the accuracy of pass detection depends on the accuracy of detection of pass events. As a result, a wrong pass event could lead us to a wrong pass detection as well as wrong possession statistics. Also, the team information needs to be stored once a pass event is detected. The team information of a pass-start event needs to be matched with the team information of a pass-end event to determine a valid pass. This two-step process is therefore complex and irrecoverable to errors if there is a mistake in one step.

To cope with the above difficulties, we have modeled a pass as a spatio-temporal event containing a pass-start event at frame f and a pass-end event at frame $(f+\lambda)$. The pass is detected in a temporal window w of length λ frames, shown as blue dashed rectangle in Fig. 1b. The proposed model follows a predict-correct-predict cycle, therefore more flexible and recoverable to error. The length of a pass may vary, hence λ could vary from pass to pass.

Given an initial location of w, the main challenge of detecting a pass is two-fold; first how to move w to localize a pass in the video? Second, how to determine the length of w? A sliding window-based approach with a set of variable-length windows seems a natural choice to solve the problem. But due to exhaustive search, this approach will take huge processing time, therefore cannot be applicable for real-time applications. We solve this problem by applying reinforcement learning algorithm. We design a *localization agent*, a deep Q-network (DQN) agent, that watches the temporal window w, and learns the policy on how to move and re-scale w to locate a pass based on the reward obtained from the *soccer-environment* (a reinforcement learning environment that we have designed for this experiment). We also design an *identification agent* whose task is to identify a valid pass for team-A or team-B, localized by w. Two agents interact with each other to jointly detect a pass in the video, as shown in Fig. 1b. Once trained, the agents take decisions in real-time, therefore the proposed model is suitable for time-constrained applications.

Existing works: Automatic generation of ball possession statistics has recently gained considerable attention from computer vision researchers. But, the existing works primarily focused on pass event based pass detection, instead of modeling a pass as a spatio-temporal event. One such work is [16], where a passing event is detected based on the interaction-energy of the ball and players detected in the soccer frame. The interaction-energy is modeled in such a way that the energy is relatively high for a pass event, and low otherwise. The energy is calculated based on the position and velocity of the ball and players.

A graph based method is proposed in [17], where the pass event detection is modeled as split and merge of the nodes of a minimum-cost flow network. The network is constructed by the ball and players detected in two consecutive frames. The network has special nodes like appear, disappear, split and merge to detect appearance, disappearance, split and merge of the ball and players. A split followed by a merge of the ball engineered by two different players of the same team denotes a valid pass detection.

The modeling of [17] allows any object from the previous frame to get split (or merge) with any object of the current frame. This relaxation raises false pass detection. The method in [18] overcomes the above limitation by introducing the concept of the group. A group is formed by nearby objects in the frame and split and merge between members is allowed only in case they are part of a group [18]. A modified network structure and cost function is also proposed in [18], which improved the accuracy of generation of possession statistics.

In contrast to the existing approaches, we propose a single-step solution for pass detection. The proposed dual interacting model jointly localizes and identifies a valid pass in a video through a temporal window. The proposed method is elaborated in Section 2. Next, we discuss relevant works on reinforcement learning.

Reinforcement learning: Reinforcement learning (RL), powered by deep network, has been effectively used in different computer vision tasks like object detection [1, 2, 11], object tracking [20, 22, 24] and video summarization [25, 26]. Our model is inspired by the idea of temporal window based RL agents applied for action detection [7, 10, 23], but with specific novel designs motivated by multi-agent communication [6, 9].

Action detection by continuously repositioning and reshaping the temporal window over a video is proposed in [7, 10]. An RL agent is trained to sequentially move the window by taking actions like move left or right, expand left or right, shrink and jump. The movement of the window is terminated once an action instance is localized by the agent. Generally, a separate classification model is used to further classify the detected action instances. Our *localization agent* is designed based on a similar idea. But, instead of using a separate classification model, we use a DQN agent (*identification agent*) for pass identification. This improvement empowers the agents to cooperate and learn the pass detection task via a unified framework.

Multiple agents have been used to perform tasks such as object localization [9] and face tracking [6]. By treating each object detection as an individual agent, a collaborative multi-agent model is proposed in [9] for localization of objects which are under interactions. A dual-agent RL model for deformable face tracking is proposed in [6]. In this model, a tracking agent tracks the target face and an alignment agent adjusts the facial landmark points.

In a multi-agent environment, one agent may hold important information that can guide the policy of other agents. Due to this, multi-agent models with no communication shows poor performance than multiple interacting agents [12]. An agent may send a message m to guide another agent to achieve its task. Conventionally, m is handcrafted based on prior knowledge. Recent works [5, 6] learn m via back-propagation. In [9], the weights of the layers of one agent are merged into the layers of the other agent as the message. In contrast to that, we explicitly share the value of a state as the message for the other agent. We modify the Dueling DQN architecture [21] and enable it for multi-agent communication, which is discussed in Section 2.1.3.

Here we summarize our major contributions.

- We propose dual interacting RL agents to detect valid passes in soccer video.
- We propose a way of communication between the agents by modifying Dueling DQN architecture [21].
- We propose a novel reward function.

The rest of the paper is organized as follows. The proposed method is discussed in Section 2. Proposed dual interacting agents, their communication mechanism and pass detection process are also discussed in Section 2. In Section 3, we present comprehensive experiments on openly available soccer videos and compare our results with existing works. We conclude in Section 4.

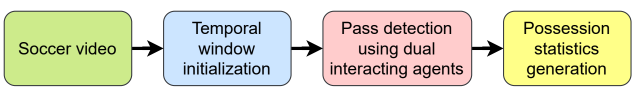


Figure 2. Block diagram of the proposed method.

2. Proposed method

The block diagram of the proposed method is shown in Fig. 2. We take a broadcast soccer video as input. We then initialize a temporal window $w = \{st, en\}$ on it, where st and en are the starting and ending frame indices. The dual agents then process the video and detect the valid pass for team-A and team-B. In the final step, we generate the possession statistics from the pass counts of two teams using (1).

The process of pass detection via a dual interacting model is shown in Fig. 3. The model contains two DQN agents, a *localization agent* and a *identification agent*. Given a temporal window w (blue dashed rectangle of Fig. 3), the identification agent decides if w contains a pass for team-A or team-B. If w does not contain a pass, the localization agent moves w to the new position w' (red dashed rectangle) that may contain a pass. This identification followed by localization continues until a pass is detected. Once a pass is detected, the temporal window is repositioned to a new position $w = w + l$ by shifting the window by l number of frames and the search for the next pass in the video continues. We next describe the dual interacting agent model.

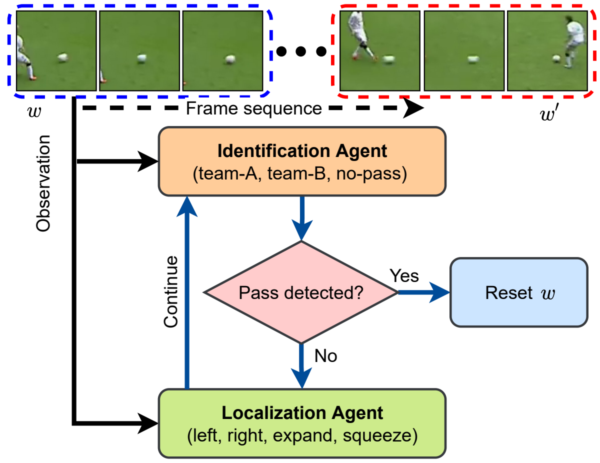


Figure 3. Pass detection using dual agent model.

2.1. Dual interacting agents

Given an initial window $w = (st, en)$, we select n number of frames using a frame selection strategy and concatenate them to obtain the feature vector $s \in \mathbb{R}^{n \times d \times h} = \{f_1, f_2, ..., f_n\}$, where f is a frame of width d and height h. The feature vector s is the current state and servers as the observation to the agents through which an agent decides to choose an action.

2.1.1 Localization agent

The localization agent observes s and takes an action a_L to move w to new position w'. In our case, we define four actions for the localization agent, which are $A_L = \{$*left, right, expand, squeeze*$\}$. Given the current window $w = (st, en)$, the *left* action moves the current window to the left, where the transformed window is $w' = (st - u, en - u)$ and u is a fixed number of frames. Similarly, the *right* move transformed w to $w' = (st + u, en + u)$. The *expand* move expands the current window w to $w' = (st - u, en + u)$. Opposite to the expand move, the *squeeze* move shrinks the current window to $w' = (st + u, en - u)$. We experimentally found that the above four actions are sufficient enough to localize a pass in a soccer video.

The goal of the localization agent is to detect a pass. Therefore, we design a reward function $R_L(s, a_L \in A_L)$ in such a way that $R_L(s, a_L)$ is positive if the action a_L moves the current window towards a pass. Upon taking an action, the localization agent receives a reward $r_L \in R_L(s, a) = +1$ if the intersection over union (IoU) between w' and a ground truth pass $w_g = (st_g, en_g)$ is greater than some threshold τ. Otherwise, we provide a small reward $r_L = +0.1$ if the translated window w' moves towards any of the ground truth window w_g. We define boundary distance between a window w and a ground truth pass w_g as $D(w, w_g) = min(|st - st_g|, |st - en_g|, |en - st_g|, |en - en_g|)$. If the boundary distance between the current window and ground truth pass reduces after moving from w to w', or IoU increases, the agent receives a reward $r_L = 0.1$. For all other cases, the agent receives a penalty $r_L = -1$, as shown below.

$$R_L(s, a_L) = \begin{cases} +1 & \text{if IoU}(w', w_g) \geq \tau, \\ +0.1 & \text{else if} \\ & \text{IoU}(w', w_g) > \text{IoU}(w, w_g) \text{ OR} \\ & D(w, w_g) > D(w', w_g), \\ -1 & \text{otherwise.} \end{cases} \tag{2}$$

2.1.2 Identification agent

Unlike the localization agent, the primary task of the identification agent is to decide whether the current window w captures a pass (for team-A or B) or not. Therefore, we choose three actions for the identification agent, which are $A_I = \{$*team-A*, *team-B*, *no-pass*$\}$. The identification agent chooses the action *team-A* or *team-B* to denote a valid pass for team-A or team-B respectively. Otherwise, the agent chooses *no-pass*.

Typically most of the time w will contain a no-pass event. To encourage the agent to detect a valid pass for team-A or team-B, we provide a positive reward $r_I \in R_I(s, a_I \in A_I) = +1$ for correct identification of a pass for team-A or team-B. We set $r_I = 0$ for a correct identification of no-pass event. For the rest of the cases, we set $r_I = -1$. Let the current window w overlaps with a ground truth pass w_g and the function $team(w_g)$ returns the team information of w_g. We define,

$$R_I(s, a_I) = \begin{cases} +1 & \text{if IoU}(w, w_g) \geq \tau \text{ AND} \\ & a_I == team(w_g), \\ +0 & \text{if IoU}(w, w_g) < \tau \text{ AND} \\ & a_I == \textit{no-pass}, \\ -1 & \text{otherwise.} \end{cases} \tag{3}$$

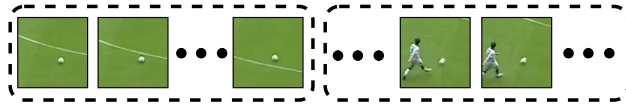
The effect of taking an action a_i by an agent i (localization or identification) in a state s, known as Q-value, is numerically measured as [13],

$$Q_i(s, a_i) = r_i + \gamma \max_{a'_i} Q_i(s', a'_i), \tag{4}$$

where r_i is the reward for the action a_i, s' and a'_i are the next state and action respectively. The maximum Q-value for the next state is discounted by the factor $0 \leq \gamma < 1$.

The Dueling DQN [21] represents the Q-value in (4) as the sum of the value of a state $V(s)$ and the advantage of taking an action in that state $A(s, a_i)$, subtracted by the average value of all actions a_k,

$$Q_i(s, a_i) = V_i(s) + (A_i(s, a_i) - \frac{1}{k} \sum_{a_k} A_i(s, a_k)), \tag{5}$$



(a) Frames contain only the ball (b) A player is passing the ball

Figure 4. Two scenarios. The scenario in (b) may have a higher state value than (a).

where k is the number of actions of the agent i.

The value of a state $V_i(s)$ quantifies how valuable a state s is to an agent i irrespective of any action, whereas the advantage $A_i(s, a_i)$ learns how much extra reward could be gained from the state s for a particular action a_i. The value of a state $V_i(s)$ for the agent i is shared to the other agent $i-$ as a message for inter-agent communication, which is discussed next.

2.1.3 Communication between agents

The performance of the identification agent heavily depends on how well a pass is localized by the localization agent. On the other hand, the identification agent can instruct the localization agent whether a pass is localized or not. This inter-agent communication is achieved by sharing a message from one agent to another. The message is either handcrafted [19] or learned via back-propagation [5, 6]. We, on the other hand, have proposed the idea of implementing inter-agent communication by sharing the state value ($V_i(s)$) of an agent to the other.

The core idea is that the state value $V_i(s)$ of a state s for an agent can help the other agent to decide its policy. Imagine two scenarios shown in Fig. 4. First, the temporal window contains just a ball (Fig. 4a) and second, a player is kicking the ball (Fig. 4b). In the first scenario, the localization agent has no clue regarding which direction to move the window to locate a pass. Hence, this state may be less important for the agent. As a result, the localization agent may move w to right and instruct the identification agent that no pass is localized. For the second scenario, the window contains a player with the ball. Therefore the localization agent may extend the window to locate the pass. Hence, Fig. 4b is more valuable than Fig. 4a for the agents.

A higher state value for the localization may convey the message to the identification agent that a pass is localized. Hence, the identification agent may choose either team-A or team-B. Alternatively, the identification agent may instruct the localization agent to move the temporal window to the right if a player kicked the ball.

Based, on the above intuition, we modify (5) and define the Q-value of an agent i as the sum of state value $V_i(s)$, the state value of the other agent $V_{i-}(s)$ and action advantage

$A_i(s, a_i)$,

$$Q_i(s, a_i) = \beta V_i(s) + (1-\beta)V_{i-}(s) + (A_i(s, a_i) - \frac{1}{k}\sum_{a_k} A_i(s, a_k)), \quad (6)$$

where β is a scaling parameter that controls the contribution of $V_i(s)$ and $V_{i-}(s)$ in $Q_i(s, a_i)$.

We modify the Dueling DQN [21] architecture to learn the Q-value of (6). The modified architecture with two agents is shown in Fig. 5. Aiming to localize a pass, the localization agent consists of a vertical stack of two bidirectional convolutional LSTM layers (Bi-ConvLSTM1 and Bi-ConvLSTM2) to capture spatio-temporal features from the sequence of frames. The vertically stacked Bi-ConvLSTMs add more depth to the architecture and create hierarchical feature representation which is assumed to improve the accuracy of the localization agent.

Guided by the fact that a soccer frame sequence is symmetric with respect to time, the bidirectional connections of the Bi-ConvLSTM layer captures temporal features in both forward and backward directions of time. The output of the last time-step of Bi-ConvLSTM2 is flattened and connected to a fully connected layer (FC1). The original Dueling architecture [21] contains two branches, one to learn $V_i(s)$ and another one to learn $A_i(s, a_i)$. In our case, we add one more branch to learn β, as shown in Fig. 5.

The identification agent consists of a 3D Convolution layer followed by a 2D Convolution layer to capture the spatial features from the frame sequence for the identification task. Note that, we are intended to capture only spatial features from the frame sequence to detect a valid pass or a miss-pass. Hence, we have used 3D and 2D Convolution layers instead of recurrent layers for the identification agent. The output of the 2D convolution layer is connected to a fully connected layer (FC1), which is divided into three branches similar to the localization agent. At a time step t, an agent reads the current state s and state value of the other agent $V_{i-}(s)$. The state value $V_{i-}(s)$ is multiplied with $(1-\beta)$ and added with $V_i(s)$ and $A_i(s, a_i)$ in the final layer (FC5) of the network. Finally, the Q-value (6) of all actions are generated in the final layer of the network, as shown in Fig. 5. The parameters β, $V_i(s)$ and $A_i(s, a_i)$ are learned via back-propagation.

2.2. Learning the optimal policy

The Q-value in (6) is approximated using a deep neural network $Q(s, a_i) \approx Q(s, a_i; \theta)$, where θ is the parameters of the network. Q-value of each agent i is learned by minimizing a loss function $\mathcal{L}_i(\theta)$, where,

$$\mathcal{L}_i(\theta) = [(r_i + \gamma \max_{a'} Q_i(s', a'; \theta)) - Q_i(s, a_i; \theta)]^2, \quad (7)$$

The loss function is defined as the squared difference between current Q-value $Q_i(s, a_i; \theta)$ and the target Q-value

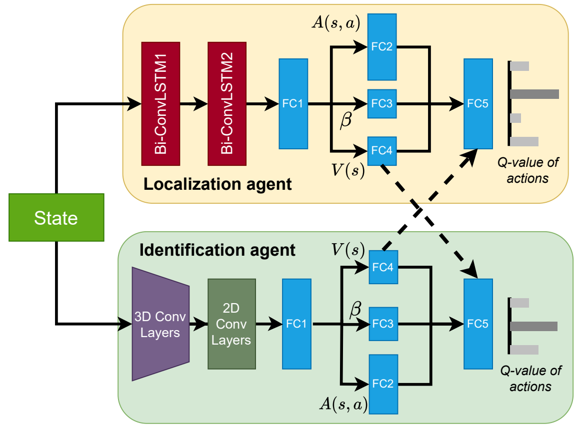


Figure 5. Communication between the agents.

$Q_i(s', a'; \theta)$ as in [13]. The parameter θ can be optimized using the gradient descent method as follows,

$$\theta_{c+1} = \theta_c + \alpha(Y_c - Q_i(s, a_i; \theta_c))\nabla_{\theta_c} Q_i(s, a_i; \theta_c), \quad (8)$$

where $Y_c = r + \gamma \max_{a'} Q_i(s', a'; \theta_c)$ and α is the learning rate.

Finally, once the Q-value of all state-action pairs is learned, the optimal policy π_i^* is obtained by greedily selecting the action that returns the maximum Q-value,

$$\pi_i^* = \text{argmax}_a\, Q_i(s, a_i). \quad (9)$$

2.3. Generation of possession statistics

Once the training process is complete, the agents have learned the optimal policy and know which action to select in which state. Now, given a soccer video as input, the steps to generate possession statistics of two teams are as follows. We initialize w at the beginning of the video and obtain the current state s. We take two counters, *passA* and *passB* to keep track of valid passes for team-A and team-B respectively.

The identification agent observes s and chooses an action $a_I = \text{argmax}_a\, Q_I(s, a_i)$. If $a_I =$ *team-A*, we say a pass for team-A is identified and *passA* is incremented by one. Similarly, *passB* is incremented by one if we get $a_I =$ *team-B*. The case $a_I =$ *no-pass* indicates no-pass situation. We then move to the localization agent to reposition w. The localization agent selects an action $a_L = \text{argmax}_a\, Q_L(s, a_i)$ and w is moved accordingly. This identification followed by localization process continues either maximum ρ number of times or a valid pass is detected. We then shift w by l frames to the right. Once the entire video is processed, the possession statistics is generated using (1). The steps for the generation of the possession statistics are shown in Algorithm 1.

Algorithm 1: Generation of possession statistics

Input: Soccer video; **Output**: Ball possession stats
Initialize the temporal window w,
Set $passA = 0, passB = 0$
while *(Video not complete)* **do**
  Set $itr = 0$
  while $(itr \leq \rho)$ **do**
    $a_I = \text{argmax}_a \, Q_I(s, a_i)$
    `// Identification agent checks if there is a pass`
    if *Valid pass detected* **then**
      $passA = passA + 1$ `// team-A`
      or
      $passB = passB + 1$ `// team-B`
      break
    else
      $a_L = \text{argmax}_a \, Q_L(s, a_i)$
      `// Localization`
    end
    $itr = itr + 1$
  end
  Reposition $w = w + l$
end
Generate ball possession statistics using (1)

3. Experimental details

Dataset: Due to the lack of any benchmark dataset on ball possession statistics, we evaluate the performance of our model on openly available broadcast soccer match videos[1] [2] [3]. We extract 8 video clips (V1-V8) from the match videos, each is encoded in mp4 format stored at 25 frames per second. Additionally, we experiment on 8-pass goal[4] (B1) and 44-pass goal[5] (B2). We implement [15] for the identification of long-shot frames in a soccer video. The long-shot frames provide a global view of the soccer field and have a consistent camera zooming effect, therefore used for our experiments.

We prepare ground truth by marking each video clip with the tuple $\{st_g, en_g, \psi\}$, where st_g and en_g are the starting and ending frame index of a pass and $\psi = \{\textit{team-A}, \textit{team-B}\}$ is the team label of the pass. For our experiment, we design the *soccer environment*, an RL environment responsible for the execution of the actions of localization and identification agents. The *soccer environment* also returns the rewards R_L and R_I to the localization and identification agents respectively.

Implementation details: Given a soccer frame, we crop a region of interest (RoI) of width $d = 100$ pixels and height $h = 100$ pixels surrounding the ball. The frames shown in Fig. 4 are examples of cropped RoI. The RoI typically contains the soccer ball and possessing player, therefore has sufficient image information for the localization and identification task. We experimentally set $u = 3$ in Section 2.1, therefore w is shifted by 3 frames for different actions of the localization agent. We also experimentally set $l = 15$ frames in Section 2.3, therefore once a pass is detected, w is shifted by 25 frames.

Given a soccer video, we initialize the window $w = (st, en)$ with $st = 0$ and $en = 20$ and select $n = 6$ frames that serve as the state s to the agents. In case of a valid pass, a player passes the ball to another player of the same team. Therefore, if w properly localizes a pass, the starting frames of w would have a player passing the ball. Also, the ending frames of w would have a player receiving the ball. Based on this fact, we select the first 3 frames out of 6 frames from the beginning of w (st to $st + 2$). The rest of the 3 frames are selected from the end of w ($en - 2$ to en). We concatenate the 6 frames to obtain the feature vector of shape $s \in \mathbb{R}^{6\times100\times100}$.

The first bidirectional convolutional LSTM layer (Bi-ConvLSTM1) of the localization agent contains 64 convolutional filters of kernel size 5×5 and strides (1,1), whereas the Bi-ConvLSTM2 contains 32 convolutional filters of kernel size 3×3 and strides (1,1). The output of the last time step of Bi-ConvLSTM2 is reduced using 2D max-pooling operation of kernel size 2×2, which is then connected to a fully connected layer (FC1 of Fig. 5) of size 512. The localization agent has 4 actions, therefore the size of FC2 is set to 4. Similarly, $V(s)$ and β are scalar values, therefore the size of FC3 and FC4 are set to a vector of length 1. FC2, FC3 and FC4 are merged in FC5 of size 4.

For the identification agent, the 3D conv layer contains 64 convolutional filters of kernel size $3 \times 3 \times 3$ and strides (1,1,1). The 2D conv layer contains 64 convolutional filters of kernel size 3×3 and strides (1,1). The output of the 2D conv layer is reduced using 3D max-pooling operation of kernel size $2 \times 2 \times 2$, which is then connected to a fully connected layer of size 512. The localization agent has 3 actions, therefore the size of FC2 and FC5 are set to 3. The size of FC3 and FC4 are set to 1. We assign linear activation to FC5 to generate real-valued Q-values but sigmoid activation to FC3 to generate the values between 0 to 1 for β. All other layers have ReLu activation.

Both agents are trained using the Adam [8] optimizer.

[1] Real Madrid vs Valencia, LaLiga, 04-Jan-2016; URL: https://www.youtube.com/watch?v=7uHGd7yNm6I.
[2] Manchester United vs Everton, FA Cup, 23-Apr-2016; URL: https://www.youtube.com/watch?v=cC18Y--L-7w.
[3] Gladbach vs. Bayern, Bundesliga, 18-May-2013; URL: https://www.youtube.com/watch?v=GGhhbMOp6yY.
[4] Bayer Leverkusen vs Werder Bremen, DFB Pokal, 07-Feb-2018; URL: https://www.youtube.com/watch?v=xdfac1tUuN8.
[5] Manchester City vs Manchester United, Premier League, 11-Nov-2018; URL: https://www.youtube.com/watch?v=DboajdmSHJM.

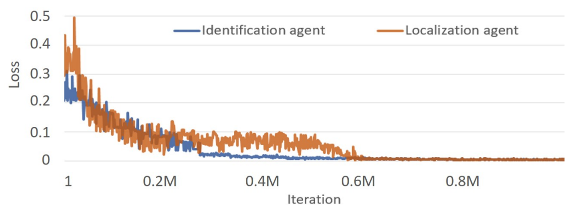


Figure 6. Change of loss during the training of the agents.

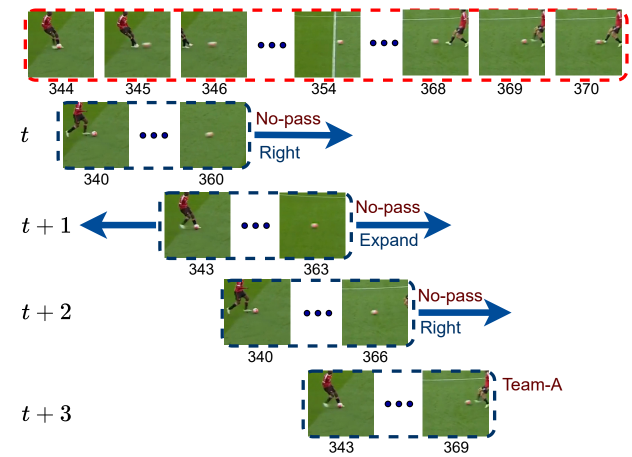


Figure 7. Visualization of different steps of a pass detection by our method. A pass (343,369) is detected by the agents at step $t+3$. Actions selected by the identification and localization agents are shown above and below the arrow respectively.

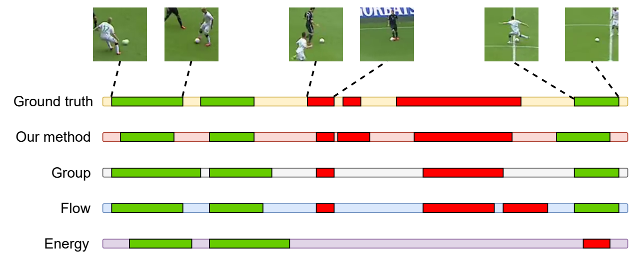


Figure 8. Visualization of pass detection by different methods. The green and red bars represent valid passes for team-A and team-B respectively.

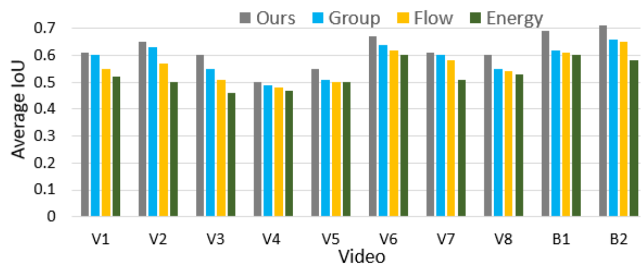


Figure 9. Comparison of IoU of the detected passes of different methods. The proposed method has higher IoU.

The initial learning rate α in (8) is set to 1.e-5, which is linearly reduced to 1.e-8 over 1M iterations. For each iteration, we take a batch of size 64 for our experiment. The training is stopped when the loss $\mathcal{L}(\theta) \leq 0.001$ for both the agents. A typical change of the loss value of both agents during training over an iteration of 1M is shown in Fig. 6. We see that the loss is gradually decreasing near to zero, which empirically shows the convergence of the agents. We implement ε-greedy and experience replay [14] for the training of the agents.

Qualitative analysis: Typical steps of a pass detection by the agents is visualized in Fig. 7. A ground truth pass that starts at frame 344 and ends at 370 is shown at the top of the figure. At time t, the identification and localization agents observe the window (340, 360) (blue dashed rectangle) and select *no-pass* and *right* actions respectively. Next, the identification agent selects *expand* move at $t+1$ and *right* move at $t+2$. The actions selected by the identification agent remain same at $t+1$ and $t+2$. At time $t+3$, the identification agent observes the window (343, 369) and selects the action *team-A*, which denotes detection of a pass for team-A.

A visualization of pass detection by different methods on V6 is shown in Fig. 8. The frame sequence is represented by a straight line and the valid passes for team-A and team-B are represented via red and green bars respectively. The length of each bar represents the span of a pass. We can see that our method has successfully detected all the passes marked in the ground truth. But the methods of Group [18], Flow [17] and Energy [16] have failed to detect all the passes. Also, we can see that our method has higher intersection over union ratio (IoU), which confirms that our method is better at pass localization than the other methods.

Quantitative analysis: Our model is better at localizing a pass than the other methods. The plot of the average IoU of detected passes by different methods is shown in Fig. 9. The average IoU of our method is 0.61, which is 0.58 for Group [18], 0.56 for Flow [17] and 0.52 for Energy [16]. To evaluate our model quantitatively, we define two metrics namely pass detection accuracy and possession statistics accuracy. Let a pass be detected between the frame-interval (st, en), where st is the starting frame and en is the ending frame. The pass is said to be successfully detected if the IoU between the pass (st, en) and a ground truth pass (st_g, en_g) is greater than a threshold ζ. We experimentally set $\zeta = 0.4$ in our case.

Given a soccer video with ι number of ground truth passes and κ number of detected passes, the pass count error is defined as $(\frac{|\iota-\kappa|}{\iota} \times 100)\%$. Then, the pass count accuracy is measured as $(100 - \text{pass count error})\%$. Similarly, we define the possession statistics error as $(\frac{|\chi-\psi|}{\chi} \times 100)\%$, where χ is the ground truth possession stat and ψ is the estimated possession stat derived from (1). The possession statistics accuracy is measured as $(100 - \text{possession statistics error})\%$.

A graphical comparison of the pass detection accuracy of our method with the other methods is shown in Fig. 10. The average accuracy of our pass detection with respect to ground truth is 81.6%, which is approximately 0.5% lower than Group [18]. But, our method is 7.9% better than Flow [17] and 15.8% better than Energy [16] in pass detection. Table 1 shows the ball possession statistics generated from different methods. The comparison of the accuracy of different methods is shown in Fig. 11. The average error of pass detection, as well as possession statistics calculation is shown in Table 2. The average error of possession statistics calculation of our method is 13.35%, which is 12.10% for Group [18], 15.35% for Flow [17] and 18.85% for Energy [16]. Although our method has marginally higher error on possession stat calculation, our method is significantly faster than Group [18].

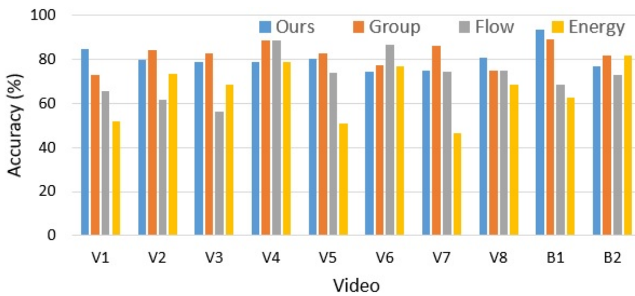


Figure 10. Comparison of the accuracy of pass detection by the proposed method with Group [18], Flow [17] and Energy [16].

Table 1. Comparison of possession stat.

Video	Ground Truth		Ours		Group		Flow		Energy	
	A (%)	B (%)	A (%)	B (%)	A (%)	B (%)	A (%)	B (%)	A (%)	B (%)
V1	80	20	75	25	74	26	73	27	71	29
V2	55	45	45	55	46	54	33	67	66	34
V3	43	57	52	48	53	47	62	38	30	70
V4	37	63	28	72	32	68	32	68	28	72
V5	30	70	38	62	27	73	30	70	31	69
V6	47	53	51	49	39	61	41	59	57	43
V7	76	24	85	15	79	21	76	24	67	33
V8	45	55	45	55	35	65	35	65	31	69
B1	100	0	90	10	100	0	93	7	78	22
B2	100	0	94	6	96	4	95	5	87	13

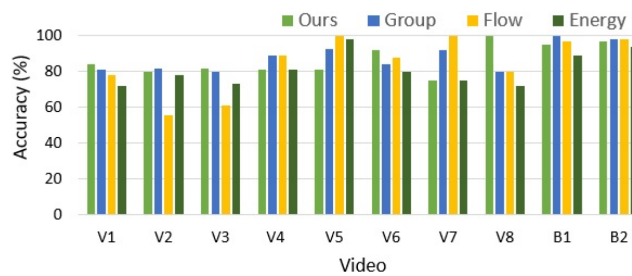


Figure 11. Comparison of the accuracy of calculation of possession statistics by different methods.

Table 2. Comparison of error (pass detection and possession stat) and processing time.

Method	Pass detection error (%)		Possession stat error (%)		Processing time (sec)
	team-A	team-B	team-A	team-B	
Ours	20.5	16.4	13.3	13.4	0.05 (GPU)
Group	11.8	24.0	11.7	12.5	21.8
Flow	26.7	25.9	15.3	15.4	6.86
Energy	33.0	35.4	18.8	18.9	0.08

Our method takes approximately 0.05 seconds to process a frame on GPU. This is a major advantage of the proposed method. The proposed model is implemented in Keras [3] with Tensorflow backend. The training of the network is done on NVIDIA TITAN RTX GPU with 24 GB memory, CUDA 10.1 and cuDNN 7.6. with an unoptimized Python 3.6 code on a PC with Intel i9 3.3 GHz processor, 64 GB RAM and Linux operating system.

4. Conclusions

This paper shows the application of reinforcement learning in generation of possession statistics in telecasted soccer videos. The proposed dual interacting agent model shows competitive results compared to the state-of-the-art method. This paper also shows an effective way of communication between agents by utilizing the state-value of a Dueling DQN. Experimental results show that the proposed method is significantly faster than the competitive methods, therefore applicable to real-time applications. We plan to modify the reward function of the agents in the future and add contextual information like the motion of the ball and player to improve the accuracy of pass localization. We also plan to explore the applicability of the proposed model in other ball-based games like field hockey and basketball.

5. Acknowledgments

This work is partially supported by DRDO, Government of India (project grant: ERIP/ER/202009001/M/01/1781).

References

[1] Miriam Bellver, Xavier Giro-i Nieto, Ferran Marques, and Jordi Torres. Hierarchical object detection with deep reinforcement learning. In *Deep Reinforcement Learning Workshop, NIPS*, December 2016. 2

[2] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2488–2496, 2015. 2

[3] François Chollet et al. Keras. `https://keras.io`, 2015. 8

[4] Vinicius Martins Farias, Wesley Bierhals Fernandes, Gabriel Gustavo Bergmann, and Eraldo dos Santos Pinheiro. Relationship between ball possession and match outcome in uefa champions league. *Motricidade*, 16(4):1–7, 2020. 1

[5] Jakob N Foerster, Yannis M Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016. 3, 4

[6] Minghao Guo, Jiwen Lu, and Jie Zhou. Dual-agent deep reinforcement learning for deformable face tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 768–783, 2018. 2, 3, 4

[7] Jingjia Huang, Nannan Li, Tao Zhang, Ge Li, Tiejun Huang, and Wen Gao. Sap: Self-adaptive proposal model for temporal action detection based on reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[9] Xiangyu Kong, Bo Xin, Yizhou Wang, and Gang Hua. Collaborative deep reinforcement learning for joint object search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1695–1704, 2017. 2, 3

[10] Nannan Li, Hui-Wen Guo, Yang Zhao, Thomas Li, and Ge Li. Active temporal action detection in untrimmed videos via deep reinforcement learning. *IEEE Access*, 6:59126–59140, 2018. 2

[11] Stefan Mathe, Aleksis Pirinen, and Cristian Sminchisescu. Reinforcement learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2894–2902, 2016. 2

[12] Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012. 2

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 4, 5

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015. 7

[15] Saikat Sarkar, Sazid Ali, and Amlan Chakrabarti. Shot classification and replay detection in broadcast soccer video. In *Advanced Computing and Systems for Security*, pages 57–66. Springer, 2020. 6

[16] Saikat Sarkar, Amlan Chakrabarti, and Dipti Prasad Mukherjee. Estimation of ball possession statistics in soccer video. In *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*, pages 1–8, 2018. 1, 2, 7, 8

[17] Saikat Sarkar, Amlan Chakrabarti, and Dipti Prasad Mukherjee. Generation of ball possession statistics in soccer using minimum-cost flow network. In *IEEE CVPR Workshop CVSports*, pages 2515–2523, 2019. 1, 2, 7, 8

[18] Saikat Sarkar, Dipti Prasad Mukherjee, and Amlan Chakrabarti. From soccer video to ball possession statistics. *Pattern Recognition*, page 108338, 2022. 1, 2, 7, 8

[19] Howard M Schwartz. *Multi-agent machine learning: A reinforcement approach*. John Wiley & Sons, 2014. 4

[20] James Supancic III and Deva Ramanan. Tracking as online decision-making: Learning a policy from streaming videos with reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 322–331, 2017. 2

[21] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016. 3, 4, 5

[22] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015. 2

[23] Wanru Xu, Zhenjiang Miao, Jian Yu, and Qiang Ji. Deep reinforcement learning for weak human activity localization. *IEEE Transactions on Image Processing*, 29:1522–1535, 2019. 2

[24] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2711–2720, 2017. 2

[25] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 2

[26] Kaiyang Zhou, Tao Xiang, and Andrea Cavallaro. Video summarisation by classification with deep reinforcement learning. In *British Machine Vision Conference*, page 298, 2018. 2