Automatic Pass Annotation from Soccer Video Streams Based on Object Detection and LSTM

Danilo Sorano[1], Fabio Carrara[2], Paolo Cintia,[1]
Fabrizio Falchi[2], and Luca Pappalardo[2] ✉

[1] Department of Computer Science, University of Pisa, Italy
[2] ISTI-CNR, Pisa, Italy
luca.pappalardo@isti.cnr.it

Abstract. Soccer analytics is attracting increasing interest in academia and industry, thanks to the availability of data that describe all the spatio-temporal events that occur in each match. These events (e.g., passes, shots, fouls) are collected by human operators manually, constituting a considerable cost for data providers in terms of time and economic resources. In this paper, we describe PassNet, a method to recognize the most frequent events in soccer, i.e., passes, from video streams. Our model combines a set of artificial neural networks that perform feature extraction from video streams, object detection to identify the positions of the ball and the players, and classification of frame sequences as passes or not passes. We test PassNet on different scenarios, depending on the similarity of conditions to the match used for training. Our results show good classification results and significant improvement in the accuracy of pass detection with respect to baseline classifiers, even when the match's video conditions of the test and training sets are considerably different. PassNet is the first step towards an automated event annotation system that may break the time and the costs for event annotation, enabling data collections for minor and non-professional divisions, youth leagues and, in general, competitions whose matches are not currently annotated by data providers.

Keywords: Sports Analytics · Computer Vision · Applied Data Science · Deep Learning · Video Semantics Analysis.

1 Introduction

Soccer analytics is developing nowadays in a rapid way, thanks to sensing technologies that provide high-fidelity data streams extracted from every match and training session [10,20,22]. In particular, the combination of video-tracking data and soccer-logs, which describe the movements of players and the spatio-temporal events that occur during a match, respectively, allows sophisticated technical-tactical analyses [27,3,18,19,5,6]. However, from a data provider's perspective, the collection of soccer-logs is expensive, time-consuming, and not free from errors [16]. It is indeed still performed manually through proprietary software for the annotation of events (e.g., passes, shots, fouls) from video streams,

a procedure that requires around three human operators and about two hours per match [20]. Given these costs and the enormous number of matches played every day around the world, data providers collect data regarding relevant professional competitions only, of which they sell data to clubs, companies, websites, and TV shows. For all these reasons, an automated event annotation system would provide many benefits to the sports industry. On the one hand, it would bring a reduction of errors, time, and costs of annotation for data providers: an automatic annotation system may substitute one of the human operators, or it may be used to check the reliability of the events collected manually. On the other hand, it would enable data collections for non-professional divisions, youth leagues and, in general, competitions whose matches data providers have no economic convenience to annotate.

Most of the works in the literature focus on video summarization, i.e., the detection from video streams of salient but infrequent episodes in matches, such as goals, replays, highlights, and play-breaks [1,12,23,17,28]. Another strand of research focuses on the identification of players through their jersey number [9] or on the detection of ball possession [14]. Nevertheless, there are no approaches that focus specifically on the recognition of *passes* from video streams, although passes correspond to around 50% of all the events in a soccer match [20]. It hence goes without saying that a system that wants to drastically reduce errors, time and economic resources required by event annotation must be able to accurately recognize passes from video streams.

In this paper, we propose PassNet, a computer vision system to detect passes from video streams of soccer matches. We define a *pass* as a game episode in which a player kicks the ball towards a teammate, and we define *automatic pass detection* as the problem of detecting all sequences of video frames in which a pass occurs. PassNet solves pass detection combining three models: ResNet18 for feature extraction, a Bidirectional-LSTM for sequence classification, and YOLOv3 for the detection of the position of the ball and the players. To train PassNet, we integrate video streams of four official matches with data describing when each pass begins and ends, collected manually through a pass annotation application we implement specifically to this purpose. We empirically prove that PassNet overtakes several baselines on different scenarios, depending on the similarity of conditions of the match's video stream used for training, and that it has good agreement with the sets of passes annotated by human operators of a leading data collection company. Given its flexibility and modularity, PassNet is a first step towards the construction of an automated event annotation tool for soccer.

2 Related Work

De Sousa et al. [24] group event detection methods for soccer in low-level, medium-level, and high-level analysis. The low-level analysis concerns the recognition of basic marks on the field, such as lines, arcs, and goalmouth. The middle-level analysis aims at detecting the behavior of the ball and the players. The high-level analysis regards the recognition of events and video summarization.

Video summarization. Most of the works in the literature focus on the detection from video streams of salient actions such as goals, replays, highlights, and play-breaks. Bayat et al. [1] propose a heuristic method to detect goals from video streams based on the audio intensity, color intensity, and the presence of goalmouth. Zawbaa et al. [29] perform video summarization through shot-type classification, play-break classification, and replay detection, while Kapela et al. [13] can detect scores and near misses that do not result in a score. Jiang et al. [12] detect goals, shots, corner kicks, and cards through a combination of convolutional and recurrent neural networks that proceeds in three progressive steps: play-break segmentation, feature extraction, and event detection. Yu et al. [28] use deep learning to identify replays and associated events such as goals, corners, fouls, shots, free-kicks, offsides, and cards. Saraogi et al. [23] develop a method to recognize notable events combining generic event recognition, event's active region recognition, and shot classification. Liu et al. [17] use 3D convolutional networks to perform play-break segmentation and action detection from video streams segmented with shot boundary detection. Similarly, Fakhar et al. [8] address the problem of highlight detection in video streams in three steps: shot boundary detection, shot view classification, and replay detection.

Ball, player and motion detection. Kahn et al. [14] use object detection methods based on deep learning to identify when a player is in possession of the ball. Gerke et al. [9] use a convolutional neural network to recognize the jerseys from labeled images of soccer players. Carrara et al. [4] use recurrent neural networks to annotate human motion streams, in which each step represents 31 joints of a skeleton, each described as a point in a 3D space.

Contribution of our work. An overview of the state of the art cannot avoid noticing that there are no approaches that focus on the recognition of *passes* from video streams. Nevertheless, automatic pass detection is essential, considering that passes correspond to around 50% of all the events in a soccer match [20]. In this paper, we fill this gap by providing a method, based on a combination of artificial neural networks, that can recognize passes from video streams.

3 Pass detection problem

An *event* is any relevant episode that occurs at some point in a soccer match, e.g., pass, shot, goal, foul, save. The type of events annotated from video streams is similar across different data collection companies, although there may be differences in the way annotated events are structurally organized [16,20]. From a video stream's perspective, an event is the sequence of n frames $\langle k_t, k_{t+1}, \ldots, k_{t+n} \rangle$ in which it takes place.

Nowadays, events are annotated from video streams through a manual procedure performed through a proprietary software (the tagger) by expert video analysts (the operators) [20]. For example, the company Wyscout[3] uses one operator per team and one operator acting as a responsible supervisor of the

[3] `https://wyscout.com/`

output of the whole match [20]. Each operator annotates each relevant episode during the match, hence defining the event's type, sub-type, coordinates on the pitch, and additional attributes [20]. Finally, the operators perform quality control by checking the coherence between the events that involve both teams, and through manually scanning the annotated events. Manual event annotation is time-consuming and expensive: since the annotation of a single match requires about two hours and three operators, the effort and the costs needed to tag an entire match-day are considerable [20].

We define automatic event detection as the problem of annotating sequences of frames in a video stream with a label representing the corresponding event that occurs during that frame sequence. In particular, in this paper we focus on *automatic pass detection*: detecting all sequences of video frames in which a *pass* occurs. A pass is a game episode in which a player in possession of the ball tries to kick it towards a teammate.

4 PassNet

Our solution to the automatic pass detection problem is PassNet, the architecture of which is shown in Figure 1.[4] It combines three tasks: *(i) feature extraction* reduces the dimensionality of the input using ResNet18 (Section 4.1); *(ii) object detection* detects the players and the ball in the video frames using YOLOv3 (Section 4.2); *(iii) sequence classification* classifies sequences of frames as containing a pass or not using a Bi-LSTM [4] (Section 4.3).

In PassNet, each frame has dimension $3 \times 352 \times 240$, where 3 indicates the RGB channel, and 352×240 is the size of an input frame. The sequence of frames that composes a video stream (Figure 1a) is provided in input to *(i)* a feature extraction module (Figure 1b), which outputs a sequence of vectors of 512 features, and *(ii)* to an object detection module (Figure 1c), which outputs a sequence of vectors of 24 features describing the positions of the ball and the closest players to it. The two outputs are combined into a sequence of vectors of 536 features (Figure 1d) and provided as input to a sequence classification module (Figure 1e), which generates a pass vector (Figure 1f) that indicates, for each frame of the original sequence, whether or not it is part of a pass sequence.

4.1 Feature Extraction

The sequence of frames is provided in input to the Feature Extraction module frame by frame, each of which is transformed into a feature vector by the image classification model ResNet18 [11]. In the end, the feature vectors are combined again into a sequence. ResNet18 consists of a sequence of convolution and pooling layers [11]. Convolution layers use convolution operations to produce a feature map by sliding a kernel of fixed size over the input tensor and computing the dot-product between the covered input and the kernel weights. Pooling layers

[4] PassNet's code and data are available at `github.com/jonpappalord/PassNet`

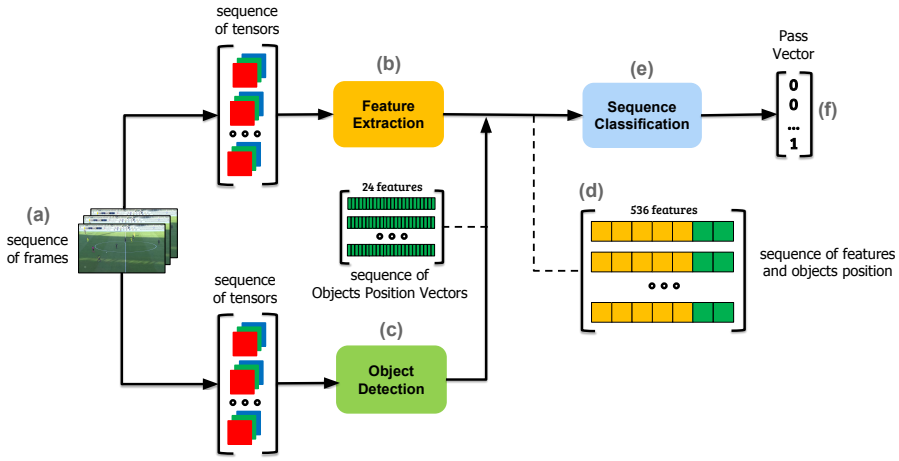


Fig. 1. Architecture of PassNet. The sequence of frames of the video stream (a) is provided to a Feature Extraction module (b) that outputs a sequence of vectors of 512 features, and to an Object Detection module (c) that outputs a sequence of vectors of 24 features describing the position of the ball and the closest players to it. The two outputs are combined into a sequence of vectors of 536 features (d) and provided to a Sequence Classification module (e), which outputs a pass vector (f) that indicates, for each frame of the original sequence, whether or not it is part of a pass sequence.

reduce the dimensionality of each feature map while retaining the most important information. ResNet18 returns as output a feature vector of 1,000 elements, which represents all possible classes (objects) in the ImageNet-1K data set [7] used to train the model. For our purpose, we take the output of the last average pooling layer of ResNet18, which generates a vector of 512 features.

4.2 Object Detection

To identify the position of relevant objects in each frame, such as the players and the ball, we use YOLOv3 [21], a convolutional neural network for real-time object detection. In particular, we use a version of YOLOv3 pre-trained on the COCO dataset [15], which contains images labeled as balls or persons. YOLOv3 assigns to each object detected inside a frame a label and a bounding box, the center of which indicates the position of the object inside the frame. Figure 2 summarizes the process of extraction of the position of the objects using YOLOv3.

We provide to YOLOv3 a match's video stream frame by frame. Based on the objects identified by YOLOv3, we convert each frame into a vector of 24 elements, that we call *Objects Position Vector* (OPV). OPV combines six vectors of length four, describing the ball and the five closest players to the ball. In particular, each of the six vectors has the following structure:

– the first element is a binary value, where 0 indicates that the vector refers to the ball and 1 that it refers to a player;

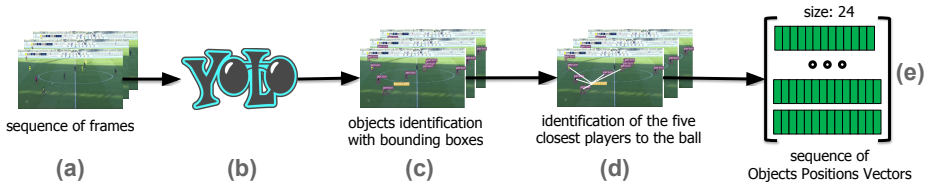


Fig. 2. Object Detection module. The sequence of frames (a) is provided to YOLOv3 [21] (b), which detects the players and the ball in each frame (c). The five closest players to the ball are identified (d) and a vector of 24 elements is created describing the positions of the objects (e).

- the second element has value 1 if the object is detected in the frame, and 0 that the vector is a flag vector (see below);
- the third and the fourth elements indicate the coordinates of the object's position, normalized in the range $[-1, +1]$, where the center of the frame has coordinates $(0, 0)$ (Figure 3a).

For example, vector $[0, 1, 0.8, -0.1]$ indicates that YOLOv3 detects that ball in a frame at position $(0.8, -0.1)$, while $[1, 1, -0.1, 0.4]$ indicates that YOLOv3 detects a player at position $(-0.1, 0.4)$. Note that YOLOv3 may identify no objects in a frame, even if they are actually present [21].[5] When an object is not detected, we substitute the corresponding vector with a *flag vector*. Specifically, if in a frame the ball is not detected, we describe the ball using flag vector $[0, 0, 0, 0]$.

We detect the five closest players to the ball by computing the distance to it of all the players detected in a frame (Figure 3b). If less than five players are detected, we describe a player using the flag vector $[1, 0, 2, 2]$. When no player or ball is detected, we use flag vectors for both the ball and the players. If at least one player is detected, but the ball is not detected, we assume that the ball is located at the center of the frame and identify the five closest players to it.

4.3 Sequence Classification

The outputs of the Feature Extraction module and the Object Detection module are combined into a sequence of vectors of 536 features and provided in input to the Sequence Classification module (Figure 4). We use a sliding window δ to split the sequence of vectors into sub-sequences of length δ. Each sub-sequence (Figure 4a) goes in input to a Bidirectional-LSTM (Figure 4b), followed by two dense layers (Figure 4c) that output a vector of δ values. Each element of this vector is transformed into 1 (`Pass`) or 0 (`No Pass`) according to a sigmoid activation function (Figure 4d) and an activation threshold (Figure 4e). The best hyper-parameter values of the Sequence Classification module (e.g., number of hidden units in the dense layers, value of activation threshold) are determined experimentally.

[5] In our experiments, we find that this situation happens for 0.66% of the frames.

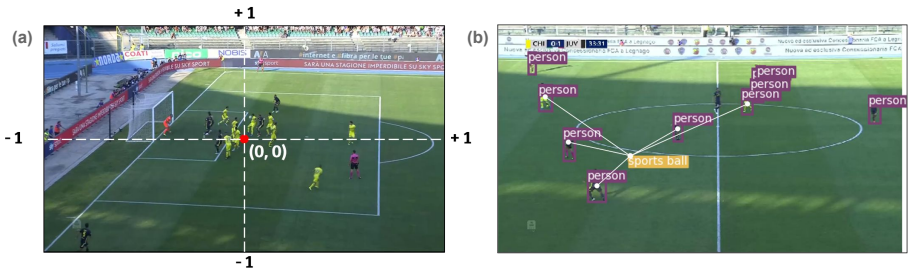


Fig. 3. Construction of the Object Position Vectors. (a) Normalization of the coordinates in the range $[-1, +1]$, where the center of the frame has coordinates $(0, 0)$. (b) The five players identified by YOLOv3 with the shortest distance from the ball.

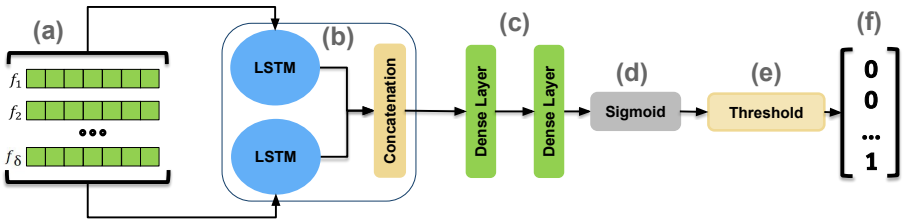


Fig. 4. Sequence Classification Module. Each sub-sequence of δ vectors (a) is provided to a Bi-LSTM (b), which processes the sub-sequence in two directions: from the first frame f_1 to the last frame f_δ and from f_δ to f_1. The output of the Bi-LSTM goes to two dense layers (c) with ReLu activation functions and dropout=0.5. A sigmoid activation function (d) and an activation threshold (e) are used to output the pass binary vector, in which 1 indicates the presence of a pass in a frame.

5 Data sets

We have video streams corresponding to four matches in the Italian first division: AS Roma vs. Juventus FC, US Sassuolo vs. FC Internazionale, AS Roma vs. SS Lazio from season 2016/2017, and AC Chievo Verona vs. Juventus FC from season 2017/2018. All of them are video broadcasts on TV and have resolution 1280×720 and 25 frames per second. We eliminate the initial part of each video, in which the teams' formations are presented and the referee tosses the coin, and we split the videos into the first and second half. For computational reasons, we reduce the resolution of the video to 352×240 and 5 frames per second.

We associate each video with an external data set containing all the spatio-temporal events that occur during the match, including passes. These events are collected by Wyscout through the manual annotation procedure described in Section 3. In particular, each pass event describes the player, the position on the field, and the time when the pass occurs [20].

We use the time of a pass to associate it with the corresponding frame in the video. Unfortunately, an event indicates the time when the pass starts, but not

when it ends. Moreover, by comparing the video and the events, we note that the time of an event is often misaligned with the video. We overcome these drawbacks by annotating manually the passes through an application we implement specifically to this purpose (see Section 5.1).

After the manual annotation, for each match, we construct a vector with a length equal to the number of frames in the corresponding video. In this vector, each element can be either 1, indicating that the frame is part of a sequence describing a pass (`Pass`), or 0, indicating the absence of a pass in the frame (`No Pass`). For example, vector [0011111000] indicates that there are five consecutive frames in which there is a pass.

5.1 Pass Annotation Application

We implement a web application that contains a user interface to annotate the starting and ending times of a pass.[6] Figure 5 shows the structure of the application's visual interface. When the user loads a match using the appropriate dropdown (Figure 5a), a table on the right side shows the match's passes and related information (Figure 5b). On the left side, the interface shows the video and buttons to play and pause it, to move forward and backward, and to tag the starting time (`Pass Start`) and the ending time (`Pass End`) of a pass (Figure 5c). When the user clicks on a row in the table to select the pass to annotate, the video automatically goes at the frame two seconds before the pass time. At this point, the user can use the `Pass Start` and `Pass End` buttons to annotate the starting and ending times, that will appear in the table expressed in seconds since the beginning of the video. In total, we annotate 3,206 passes, which are saved into a file that will be used to train PassNet and evaluate its performance.

6 Experiments

We compare PassNet with the following models:

- ResBi uses just the Feature Extraction module and the Sequence Classification module, i.e., it does not use the Object Detection module for the recognition of the position of the ball and the players;
- Random predicts the label randomly;
- MostFrequent predicts always the majority class, `No Pass` (71% of the frames);
- LeastFrequent predicts always the minority class, `Pass` (29% of the frames).

To classify a frame as `Pass` or `No Pass` we try several activation thresholds (Figure 1g): 0.5, 0.9, and the threshold that maximizes the Youden Index (YI) [2], where YI $= Rec + TrueNegativeRate - 1$, and YI $\in [0, 1]$. YI represents in the model's ROC curve the farthest point from the random classifier's curve. We compare the models in terms of accuracy (Acc), F1-score ($F1$), precision on

[6] The application is developed using python framework Flask, and is available at `https://github.com/jonpappalord/PassNet`

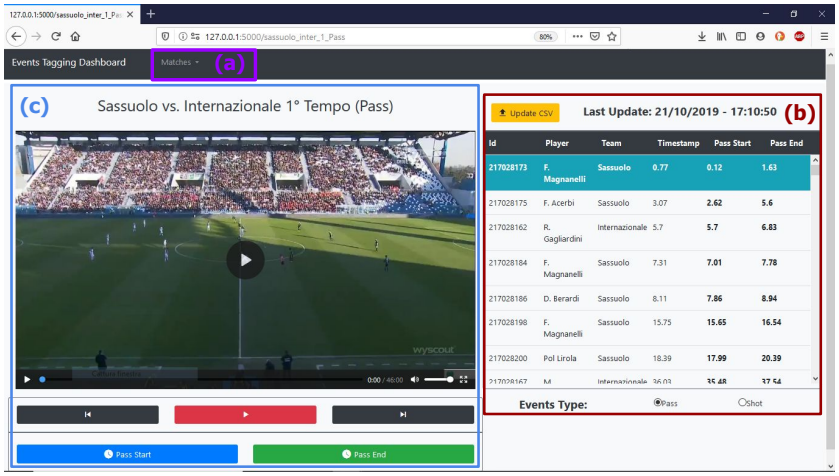


Fig. 5. Visual interface of the manual annotation application. The user can load a match using the appropriate dropdown (a). On the right side (b), a table shows all the pass events of the match and related information. On the left side (c), the interface shows the video and buttons to start and pause it, to move backward and forward, and to annotate the starting and ending times. When the user clicks on a row in the table, the video moves to two seconds before the event. A video that illustrates the functioning of the application is here: `https://youtu.be/v098f3XuTAU`.

class `Pass` ($Prec$), precision on class `No Pass` ($PrecNo$), recall on class `Pass` (Rec), recall on class `No Pass` (RecNo) [25].

We perform the experiments on a computer with CPU Intel(R) Core(TM) i7-6800K CPU @ 3.40GHz, 32 GB RAM, GPU GeForce GTX 1080. The time required to train the model (11 epochs) on a match half is around 11 hours; the time required for the annotation of a match half is on average 30 minutes.

6.1 Results

We fix some hyper-parameter values and use the first half of AS Roma vs. Juventus FC to tune the learning rate, the sequence dimension δ, and the hidden dimension of the Sequence Classification module of PassNet and ResBi (Table 1). We use each hyper-parameter configuration to train PassNet and ResBi on a validation set extracted from the training set and test the configuration on the second half of AS Roma vs. Juventus FC. In particular, we try values 128, 256 and 512 for the hidden dimension, $\delta = 10, 25$, and values 0.01, 0.001 and 0.0001 for the learning rate. We evaluate the performance of each configuration in terms of average precision (AP), the weighted mean of precision at several thresholds. Formally, $AP = \sum_n (Rec_n - Rec_{n-1}) Prec_n$, where $Prec_n$ and Rec_n are precision and recall at the n-th threshold, respectively, and Rec_{n-1} is the recall at the $(n-1)$-th threshold [25]. Table 1 shows the hyper-parameter values corresponding to the best configuration of PassNet and ResBi.

We test PassNet, ResBi and the baselines on four scenarios, in which we use: *(i)* the same match used for training the model (`Same` scenario); *(ii)* a match that shares similar video conditions as the match used for training the model (`Similar` scenario); *(iii)* matches with different teams and light conditions (`Different` scenario); *(iv)* a mix of matches with similar and different conditions (`Mixed` scenario).

	fixed hyper-parameters							**tuned hyper-parameters**			
	input dim	yolo dim	layer dim	dense dim	drop out	batch size	opti-mizer	hidden dim	seq dim	learn. rate	best epoch
PassNet	512	24	1	2	0.5	1	adam	**128**	**25**	**0.0001**	**6**
ResBi	512	-	1	2	0.5	1	adam	**256**	**25**	**0.0001**	**4**

Table 1. Hyper-parameter values of the best configuration of PassNet (AP=74%) and ResBi (AP=75%). Tuning performed using the first half of AS Roma vs. Juventus FC.

In the `Same` scenario we use two matches: we train the models on the first half of AS Roma vs. Juventus FC and test them on the second half of the same match; similarly we train the models on the first half of US Sassuolo vs. Internazionale FC and test them on the second half of the match. On AS Roma vs. Juventus FC, PassNet and ResBi have similar performance in terms of F1-score ($F1_{\text{PassNet}} = 70.21\%$, $F1_{\text{ResBi}} = 70.50\%$), and they both outperform the baseline classifiers with an improvement of 21.24% absolute and 43% relative with respect to the best baseline, `LeastFrequent` ($F1_{\text{Least}} = 49.26\%$, see Table 3). On US Sassuolo vs. FC Internazionale, PassNet has lower performance than the other match but still outperforms ResBi and the baselines ($F1_{\text{PassNet}} = 54.44\%$, $F1_{\text{ResBi}} = 53.72\%$), with an improvement of 15.75% absolute and 41% relative with respect to `LeastFrequent` ($F1_{\text{Least}} = 38.69\%$, Table 3).

We then test the models on the `Similar` scenario using the first half of AS Roma vs. Juventus FC as training set and the first half of match AS Roma vs. SS Lazio as test set. Note that this match is played by one of the teams that played the match used for training the model (AS Roma), and it is played in the same stadium (Stadio Olimpico, in Rome) and with the same light conditions (in the evening). PassNet outperforms ResBi and the baselines ($F1_{\text{PassNet}} = 61.74\%$, $F1_{\text{ResBi}} = 59.34\%$), with an improvement of 20.19% absolute and 48% relative w.r.t. `LeastFrequent` ($F1_{\text{Least}} = 41.55\%$, Table 4, right).

PassNet outperforms all models on the `Mixed` scenario, too. Here we train the models using the first halves of AS Roma vs. Juventus FC and US Sassuolo vs. FC Internazionale and test them on AC Chievo Verona vs. Juventus FC. We obtain $F1_{\text{PassNet}} = 63.73\%$ and $F1_{\text{ResBi}} = 58.17\%$, with an improvement of 11.96% absolute and 23% relative with respect to `LeastFrequent` ($F1_{\text{Least}} = 51.77\%$, see Table 4, left). Finally, we challenge the models on the `Different` scenario, in which we use match AS Roma vs. Juventus FC to train the models and we

test them on matches AC Chievo vs. Juventus FC and US Sassuolo vs. FC Internazionale. PassNet and ResBi have similar performance in terms of F1-score (Table 5) and they both outperform `LeastFrequent`. Figure 6a compares the ROC curves of PassNet and ResBi on the four experimental scenarios.

	scenario	YI	TH=.5	TH=.9
AS Roma vs. Juventus FC 2H	`Same`	70.21 (.19)	69.64	65.20
US Sassuolo vs. FC Inter 2H	`Same`	54.44 (.0005)	40.60	28.26
US Chievo vs. Juventus FC 1H	`Mixed`	63.73 (.003)	42.55	27.22
AS Roma vs. SS Lazio 1H	`Similar`	61.74 (.15)	61.70	55.24
US Sassuolo vs. FC Inter 2H	`Different`	53.92 (.004)	47.00	34.62
US Chievo vs. Juventus FC 1H	`Different`	59.51 (.0003)	23.17	8.45

Table 2. F1-score (in percentage) of PassNet at different thresholds for each match/scenario. The best value for each combination of metric and threshold is highlighted in grey. For YI, we specify the value of the threshold in parenthesis.

	2H of AS Roma vs. Juventus FC					**2H of US Sassuolo vs. FC Inter**				
	PassNet	ResBi	**baselines**			PassNet	ResBi	**baselines**		
	YI=.19	YI=.05	`Random`	`Most`	`Least`	YI=.0005	YI=.01	`Random`	`Most`	`Least`
Acc	76.07	77.18	50.09	67.32	32.68	63.74	67.88	50.61	76.02	23.98
*F*1	70.21	70.50	39.74	0.0	49.26	54.44	53.72	32.81	0.0	38.69
Prec	59.17	61.03	32.82	0.0	32.68	38.96	41.04	24.35	0.0	23.98
Rec	86.32	83.45	50.36	0.0	100	90.34	77.73	50.29	0.0	100
PrecNo	91.45	90.22	67.46	67.32	0.0	94.78	90.21	76.38	76.02	0.0
RecNo	71.09	74.13	49.95	100	0.0	55.34	64.77	50.70	100	0.0

Table 3. Comparison of PassNet, ResBi and the baselines (`Random`, `Most`, `Least`), on matches AS Roma vs. Juventus FC (left) and US Sassuolo vs. FC Internazionale (right), of the `Same` scenario. The metrics are specified in percentage. YI = Youden Index.

In summary, PassNet and ResBi significantly outperform the baselines on all four scenarios, indicating that our approach is able to learn from data to annotate pass events. We find that PassNet outperforms ResBi on all scenarios but the `Different` one, on which the two models have similar performance in terms of F1-score. In particular, PassNet achieves the best performance on the `Same` scenario ($AUC = 0.87$), followed by the `Similar` ($AUC = 0.84$), the `Mixed` ($AUC = 0.79$), and the `Different` ($AUC = 0.72$) scenarios (Figure 6a).

In general, our results highlight that the detection of the ball and the closest players to it makes a significant contribution to pass detection. This is not surprising on the `Same` scenario, in which we use the second half of the match the

	1H of AC Chievo vs. Juventus FC					1H of AS Roma vs. SS Lazio				
	PassNet	ResBi	baselines			PassNet	ResBi	baselines		
	YI=.003	YI=.0064	`Random`	`Most`	`Least`	YI=.15	YI=.07	`Random`	`Most`	`Least`
Acc	70.00	63.37	49.99	65.08	34.92	73.13	72.32	50.30	73.78	26.22
*F*1	63.73	58.17	39.59	0.0	51.77	61.74	59.34	34.90	0.0	41.55
Prec	55.15	48.38	32.82	0.0	34.92	49.26	48.25	26.58	0.0	26.22
Rec	75.48	72.95	49.89	0.0	100	82.68	77.05	50.81	0.0	100
PrecNo	83.60	80.05	67.13	65.08	0.0	91.89	89.65	74.14	73.78	0.0
RecNo	67.06	58.23	50.04	100	0.0	69.73	70.63	50.12	100	0.0

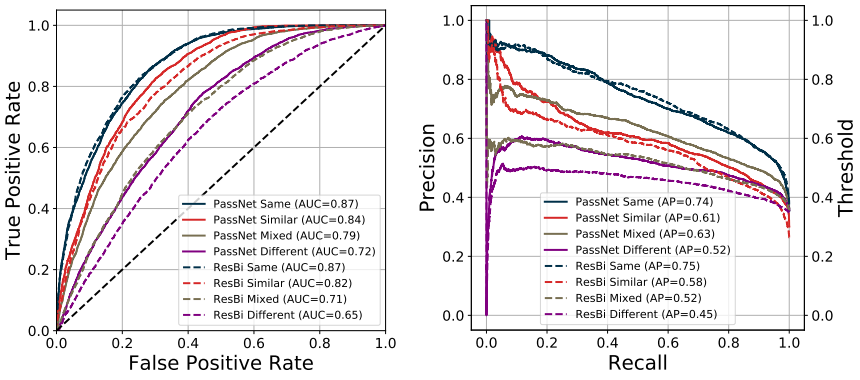
Table 4. Comparison of PassNet, ResBi and the baselines (`Random`, `Most`, `Least`), on matches AC Chievo vs. Juventus FC (left, `Mixed` scenario) and AS Roma vs. SS Lazio (right, `Similar` scenario). The metrics are specified in percentage. YI = Youden Index.

	1H of AC Chievo vs. Juventus FC					2H of US Sassuolo vs. FC Inter				
	PassNet	ResBi	baselines			PassNet	ResBi	baselines		
	YI= .0003	.00001	`Random`	`Most`	`Least`	YI= .004	.0005	`Random`	`Most`	`Least`
Acc	63.33	59.82	49.99	65.08	34.92	66.25	67.13	50.61	76.02	23.98
*F*1	59.51	53.42	39.59	0.0	51.77	53.91	54.08	32.81	0.0	38.69
Prec	48.43	44.88	32.82	0.0	34.92	40.08	40.66	24.35	0.0	23.98
Rec	77.16	65.97	49.89	0.0	100	82.34	80.71	50.29	0.0	100
PrecNo	82.02	75.58	67.13	65.08	0.0	91.65	91.17	76.38	76.02	0.0
RecNo	55.91	56.52	50.04	100	0.0	61.17	62.85	50.70	100	0.0

Table 5. Comparison of PassNet, ResBi and the baselines (`Random`, `Most`, `Least`) on AC Chievo vs. Juventus FC (left) and US Sassuolo vs. FC Internazionale (right), of the `Different` scenario. The metrics are specified in percentage. YI = Youden Index.

first half of which is used for training. However, the fact that the performance on the `Similar` scenario is better than the performance on the `Mixed` and the `Different` scenarios suggests that, provided the availability of matches for an entire season, we may build different models for different teams, for different light conditions, or a combination of both. Moreover, the similar performance of PassNet and ResBi on the `Different` scenario suggests that the contribution of the object detection module is weaker for matches whose video conditions differ significantly from those of the match used for training the model. Note that the threshold with the maximum Youden Index provides also the best results in terms of F1-score (Table 2).

Figure 6b shows how recall and precision change varying the threshold used to construct the passing vector, a useful tool for possible users of PassNet, such as data collection companies: if they want to optimize the precision of pass detection, the plot indicates that high thresholds must be used. In contrast, if they want to optimize recall, thresholds in the range $[0.0, 0.4]$ must be preferred.



(a) (b)

Fig. 6. Classification Performance. (a) ROC curves and (b) precision and recall vayring the threshold, for PassNet and ResBi on the `Same` (AS Roma vs. Juventus FC), `Similar`, `Mixed` and `Different` (AC Chievo vs. Juventus FC) scenarios.

To visualize the limits of PassNet, we create some videos that show the results of its predictions as the match goes by. In particular, we make publicly available video clips for US Sassuolo vs. FC Internazionale (`Different` scenario) and AS Roma vs. Juventus FC (`Same` scenario).[7] Figure 7a shows the structure of these videos. On the left side, we show the match, in which a label "Pass" appears every time PassNet detects a pass. On the right side, we show two animated plots that compare the real label with the model's prediction. In these plots, value 0 indicates no pass, value 1 indicates that there is a pass.

The observation of these videos reveals that PassNet sometimes classifies consecutive passes that come in a close interval of time as a single pass (AS Roma vs. Juventus FC). This is presumably because the YI threshold cannot detect the slightest changes that occur between two consecutive passes. Interestingly, the videos also reveal the presence of errors during the manual annotation. For example, in US Sassuolo vs. FC Internazionale, PassNet recognizes passes that actually take place but that were not annotated by the human operators. Another error, that may constitute room for future improvement, is that PassNet usually misclassifies as passes situations in which a player runs in possession of the ball.

As a further assessment of the reliability of the annotation made by our system, we evaluate the degree of agreement on matches in Tables 3, 4, and 5 between PassNet and Wyscout's human operators computing the Inter-Rater Agreement Rate, defined as IRAR=$1 - \frac{1-p_o}{1-p_e} \in [0, 1]$, where p_o is the relative agreement among operators (ratio of passes detected by both) and p_e is the probability of chance agreement [16]. In order to compute IRAR, we first associate each pass annotated by PassNet at time t with the Wyscout pass (if any)

[7] `https://youtu.be/l4Qt1tjfE_8`, `https://youtu.be/sOxYG4Fduoc`

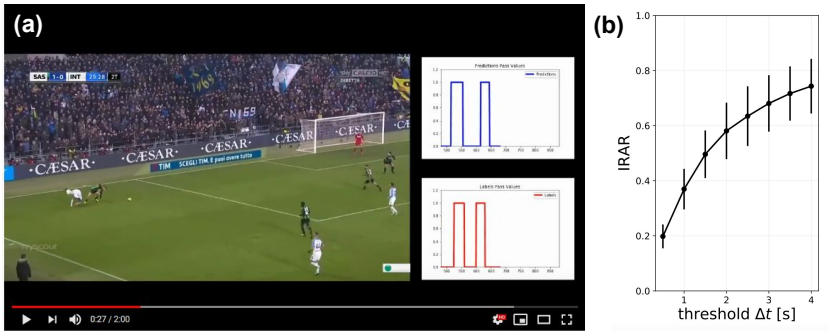


Fig. 7. PassNet in action. (a) Structure of the video showing how PassNet annotates US Sassuolo vs. FC Internazionale as the match goes by. The left side shows the match, a label "Pass" appears every time a pass is detected. The right side shows two animated plots comparing the real (red) and the predicted (blue) labels. (b) Average IRAR w.r.t. Wyscout operators varying the time threshold Δt.

in the time interval $[t-\Delta t, t+\Delta t]$ (see [16,19]). Figure 7b shows how the mean IRAR varies with Δt: at Δt=1.5s, mean IRAR $\approx$ 0.50, referred to as "moderate agreement" in [26], at Δt=3s, mean IRAR $\approx$ 0.70, referred to as "good agreement". These results are promising considering that the typical agreement between two Wyscout human operators with $\Delta t = 1.5$ is IRAR=0.70 [19].

7 Conclusion

In this article, we presented PassNet, a method for automatic pass detection from soccer video streams. We showed that PassNet outperforms several baselines on four different scenarios, and that it has a moderate agreement with the sets of passes annotated by human operators.

PassNet can be improved and extended in several ways. First, in this article, we use broadcast videos that contain camera view changes and play-breaks such as replays, checks at the VAR, and goal celebrations. These elements may introduce noise and affect the performance of the model. The usage of fixed camera views and play-break detection models may be used to clean the video streams, reduce noise, and further improve the performance of PassNet. Second, we use a pre-trained YOLOv3 that can recognize generic persons and balls. Although our results show that it provides a significant contribution to the predictions, we may build an object detection module to recognize specifically the soccer players and ball. Given the flexibility of YOLOv3's architecture, this may be achieved simply by training the model on a labeled data set of soccer players and balls. Moreover, we may integrate in PassNet existing methods to detect the identity of players, for example by recognizing their jersey number. Finally, PassNet can be easily adapted to annotate other crucial events, such as shots, fouls, saves, and tackles, or to discriminate between accurate and inaccurate passes. Given

the modularity of our model, this may be achieved simply by training the model using frames that describe the type of event of interest.

In the meanwhile, PassNet is a first step towards the construction of an automated event detection tool for soccer. On the one hand, this tool may reduce the time and cost of data collection, by providing a support to manual annotation. For example, event annotation may be partially delegated to the automatic tool and human annotators can focus on data quality control, especially for complex events such as duels and defending events. On the other hand, it would consent to extend the data acquisition process to unexplored directions, such as matches in youth and non-professional leagues, to leagues far away in the past, and to practice matches in training sessions, allowing researchers to compare technical-tactical characteristics across divisions, times and phases of the season.

Acknowledgments

This work has been supported by project H2020 SoBigData++ #871042.

References

1. Bayat, F., Moin, M.S., Bayat, F.: Goal detection in soccer video: Role-based events detection approach. International Journal of Electrical & Computer Engineering (2088-8708) **4**(6) (2014)
2. Berrar, D.: Performance measures for binary classification. In: Encyclopedia of Bioinformatics and Computational Biology, pp. 546 – 560 (2019)
3. Bornn, L., Fernandez, J.: Wide open spaces: A statistical technique for measuring space creation in professional soccer. In: MIT Sloan Sports Analytics Conference (2018)
4. Carrara, F., Elias, P., Sedmidubsky, J., Zezula, P.: Lstm-based real-time action detection and prediction in human motion streams. Multimedia Tools and Applications pp. 1–23 (2019)
5. Cintia, P., Giannotti, F., Pappalardo, L., Pedreschi, D., Malvaldi, M.: The harsh rule of the goals: Data-driven performance indicators for football teams. In: IEEE International Conference on Data Science and Advanced Analytics. pp. 1–10 (2015)
6. Decroos, T., Bransen, L., Van Haaren, J., Davis, J.: Actions speak louder than goals: Valuing player actions in soccer. In: 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 18511861 (2019)
7. Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
8. Fakhar, B., Kanan, H.R., Behrad, A.: Event detection in soccer videos using unsupervised learning of spatio-temporal features based on pooled spatial pyramid model. Multimedia Tools and Applications **78**(12), 16995–17025 (2019)
9. Gerke, S., Muller, K., Schafer, R.: Soccer jersey number recognition using convolutional neural networks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 17–24 (2015)
10. Gudmundsson, J., Horton, M.: Spatio-temporal analysis of team sports. ACM Computing Surveys **50**(2) (2017)

11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Jiang, H., Lu, Y., Xue, J.: Automatic soccer video event detection based on a deep neural network combined cnn and rnn. In: 28th IEEE International Conference on Tools with Artificial Intelligence. pp. 490–494 (2016)
13. Kapela, R., McGuinness, K., Swietlicka, A., OConnor, N.E.: Real-time event detection in field sport videos. In: Computer vision in Sports, pp. 293–316 (2014)
14. Khan, A., Lazzerini, B., Calabrese, G., Serafini, L.: Soccer event detection. In: 4th International Conference on Image Processing and Pattern Recognition. pp. 119–129 (2018)
15. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755 (2014)
16. Liu, H., Hopkins, W., Gmez, A.M., Molinuevo, S.J.: Inter-operator reliability of live football match statistics from opta sportsdata. International Journal of Performance Analysis in Sport **13**(3), 803–821 (2013)
17. Liu, T., Lu, Y., Lei, X., Zhang, L., Wang, H., Huang, W., Wang, Z.: Soccer video event detection using 3d convolutional networks and shot boundary detection via deep feature distance. In: International Conference on Neural Information Processing. pp. 440–449 (2017)
18. Pappalardo, L., Cintia, P.: Quantifying the relation between performance and success in soccer. Advances in Complex Systems **20**(4) (2017)
19. Pappalardo, L., Cintia, P., Ferragina, P., Massucco, E., Pedreschi, D., Giannotti, F.: Playerank: Data-driven performance evaluation and player ranking in soccer via a machine learning approach. ACM Trans. Intell. Syst. Technol. **10**(5) (2019)
20. Pappalardo, L., Cintia, P., Rossi, A., Massucco, E., Ferragina, P., Pedreschi, D., Giannotti, F.: A public data set of spatio-temporal match events in soccer competitions. Nature Scientific Data **6**(236) (2019)
21. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
22. Rossi, A., Pappalardo, L., Cintia, P., Iaia, F.M., Fernàndez, J., Medina, D.: Effective injury forecasting in soccer with gps training data and machine learning. PLoS One **13**(7), 1–15 (2018)
23. Saraogi, H., Sharma, R.A., Kumar, V.: Event recognition in broadcast soccer videos. In: Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing. p. 14. ACM (2016)
24. de Sousa, S.F., Araújo, A.d.A., Menotti, D.: An overview of automatic event detection in soccer matches. In: IEEE Workshop on Applications of Computer Vision. pp. 31–38 (2011)
25. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to data mining. Pearson Education India (2016)
26. Viera, A.J., Garrett, J.M.: Understanding interobserver agreement: the kappa statistic. Family medicine **37 5**, 360–3 (2005)
27. Wei, X., Sha, L., Lucey, P., Morgan, S., Sridharan, S.: Large-scale analysis of formations in soccer. In: 2013 International Conference on Digital Image Computing: Techniques and Applications. pp. 1–8 (2013)
28. Yu, J., Lei, A., Hu, Y.: Soccer video event detection based on deep learning. In: International Conference on Multimedia Modeling. pp. 377–389. Springer (2019)
29. Zawbaa, H.M., El-Bendary, N., Hassanien, A.E., Kim, T.h.: Event detection based approach for soccer video summarization using machine learning. International Journal of Multimedia and Ubiquitous Engineering **7**(2), 63–80 (2012)