Formation2Vec: Exploring a Representation for Formation Segmentation and Detection in Soccer Games

Sarit Kraus
Department of Computer Science, Bar-Ilan University, Israel
pcchair@ijcai19.org

Abstract

In the soccer game, formation depicts the role of the soccer players on the pitch, revealing the basic tactic of the teams. During a game, a team will switch between offensive and defensive formations as they get or lose the ball. Although the analysis of formation changing is important in soccer, there are few methods to detect and classify the formations automatically. In this paper, we analogize the formation detection to the human action segmentation and classification problem. Referring to the general framework to solve such a problem, we firstly adopt a neighborhood aware method to model the positional data and obtain the topological structure of the players, which can help represent the position. Our experiments prove that the representation outperforms the alternative representations, e.g. player positions and parameterized pitch. Then by a convolutional network, the temporal patterns will be captured, enabling us to segment the game into the periods with specific formations. The accuracy of the method achieves a satisfactory level in action segmentation and classification area. Compared with the existing methods in formation detection, our method supports fine-grained analysis of formation change and shows a wider application prospect.

1 Introduction

Soccer is one of the most popular sports in the world with huge commercial value. This 11-a-side competition shows a high degree of confrontation and dynamic. The players on the pitch (except goalkeeper) are organized by team **formation**, which is arranged by the coach considering the characteristics of the players and the situation of the game. Formation indicates the role of the players on the pitch. For example, formation 4-4-2 means 4 backwards, 4 midfielders and 2 forwards, as Figure 1 shows. The players will try to keep their relative positions according to the formation. During a game, the formation changes frequently. As the team obtains the ball, the players will move forward and the number of people in the frontcourt will increase. A good formation helps the team control the midfield and generate fierce attacks. When the opponent gets the ball back and approaches the penalty area, the team will adopt a deep defensive formation, like 6-3-1, to enhance the defense. If the defensive formation is not held tightly, leaving space for the opponents, the backwards will be more likely to make mistakes and lose the game.

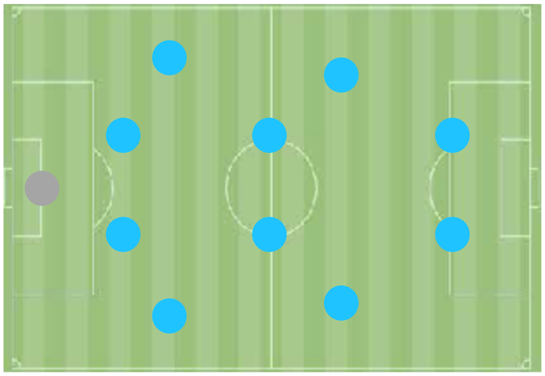


Figure 1: The relative position of the players with formation 4-4-2.

As the formation plays a key role in soccer games, many works tried to analyze the games from the perspective of formation. [Bialkowski *et al.*, 2014a] analyzed the formation of a team when it is at home and away and the influence on the results of the game. [Wei *et al.*, 2013] used formation as a prior knowledge to explore the offensive and conceding patterns of a team. These above works assume that a team generally adopts one or two formations in a game. [Wu *et al.*, 2019] developed a visual analytic system called ForVizor for dynamic formation analysis, and one of the challenges in this work is the access to fine-grained formation data. Few works in the literature focused on formation detection, and the most approved one is the algorithm proposed by [Bialkowski *et al.*, 2014b]. They used a method equivalent to k-means to aggregate the sequential positional data into gaussian distributions, and then the agglomerative clustering is adopted for formation detection. As the method is an unsupervised approach, thus predicting the formation with raw data is not feasible.

Two major challenges in formation segmentation and detection are feature selection and temporal pattern learning for dynamic data. Extracting an effective high-level feature representation characterizing the highly dynamic and unstructured positional data of the players commonly used in soccer analysis is challenging. Due to the unstructured nature of the positional data, the available feature extraction algorithms in image processing and network embedding cannot be directly applied to this problem. For segmentation, as the formation change of a team is a continuous and evolutional process, capturing the transformation of formation and ignoring the subtle changes of the features pose a significant challenge to a segmentation model. The semantic pattern in formation is hard to define.

We address the first challenge by proposing a neighborhood aware representation. The idea comes from the observation of the players on the pitch. A player is unable to see his position from an aerial view. To preserve a formation, he must keep aware of the relative positions of his teammates to determine if he is at the right position and keep a proper distance from other players. During the game, every player adjusts the position dynamically to ensure that the formations would largely hold.

To address the second challenge, we adopt a temporal convolutional network (TCN), a supervised manner, to learn the temporal patterns. TCN is a variant of convolutional network designed for sequential data. As the convolutional networks achieve good performance in image segmentation and classification, [Gehring *et al.*, 2017] from Facebook and [Kalchbrenner *et al.*, 2016] from Google successfully applied CNNs to machine translation. [Lea *et al.*, 2017] proposed TCN and achieved better results than the previous state of the art in the field of human action detection and segmentation. [Bai *et al.*, 2018] evaluated TCNs in wider range of problems and demonstrated the strengths of TCNs, but the application of TCNs regarding formation detections have not been discussed.

Our main contributions are as follows. Firstly, we define the problem of formation detection as a sequence segmentation and detection problem. Secondly, we propose a representation suitable for formation segmentation and classification. Thirdly, we apply the TCN to formation data and obtain satisfactory results compared to the base line model.

2 Problem Definition

2.1 Data Preprocessing

Our data was collected from several panoramic videos of soccer games. We used a semi-automatic collecting method to label the position of the players in the videos. To preserve the efficiency and precision, a particle filter-based tracking algorithm, developed by [Dearden *et al.*, 2006], was run under the supervision of the annotators. When occlusion happened and the tracker lost the target, the annotators would manually stop and correct the position of the tracker and make sure the tracking data is accurate. Data smoothing was adopted to avoid drifting and reduce the noise.

With the acquired position data of the players in the panoramic videos, we further used an affine projection to map the position data to a two-dimensional soccer pitch. According to the latest *Law of the Game 2018/19*[1] by the International Football Association Board (IFAB), the length of the pitch (touchline) is between 100m and 110m and the width (goal line) is between 64 and 75m. However, in many important international competitions, such as 2018 Russia World Cup, the pitch dimension is 105m by 68m. To keep the degree of accuracy, we mapped the positional data into a 1050 $\times$ 680 area.

The events on the pitch, such as foul, out of line and goal, are important for game analysis. The offensive or defensive state of a team can be interrupted by the events as the game will stop and restart. With the events, we segmented the game into small periods, and the time between the periods is stoppage time. Each period is independent to other periods. As a result, the position change within a period can be viewed as a sequential process.

To label the formation information of the game, we closely collaborated with the domain experts in soccer, two of whom were previous professional soccer players in the Chinese Football Association Super League, the first-class league in China. They watched the soccer game videos and labeled the time when formations exist with an assistant system. The rest of the time is labeled with no formation.

2.2 Problem Definition and Parameters Description

With the labeled events on the pitch, the stoppage time is eliminated and the whole game is divided into representation sequences $\{S_i\}(i = 1, 2, \cdots, N)$, and N is the amount of the sequences. The sequence $S_i = [r^i_1, r^i_2, \cdots, r^i_{N_i}], r^i_j \in \mathbb{R}^k$, where N_i is the length of the sequence and k is the dimension of the representation (or feature) space. Each sequence S_i is assigned with the labels $l_i = [l^i_1, l^i_2, \cdots, l^i_{N_i}]$ and the label $l^i_j \in L$, where label set $L =$ [No Formation, 4-4-2, 4-2-3-1, 2-3-2-3, $\cdots$]. Given a sequence, the classification function $f : \mathbb{R}^{t\times k} \to L^t$. The formation segmentation and detection is defined as an optimization problem

$$\arg\min_f \sum_{i=1}^{N} \|f(S^i) - l_i\|. \tag{1}$$

3 Proposed Method

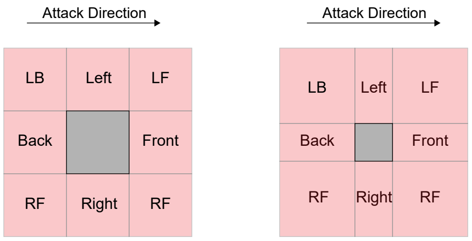
Our task is finding a suitable representation of the relative position of the players and a model to learn the temporal pattern of the representations.

3.1 Distribution Representation

Position Scaling

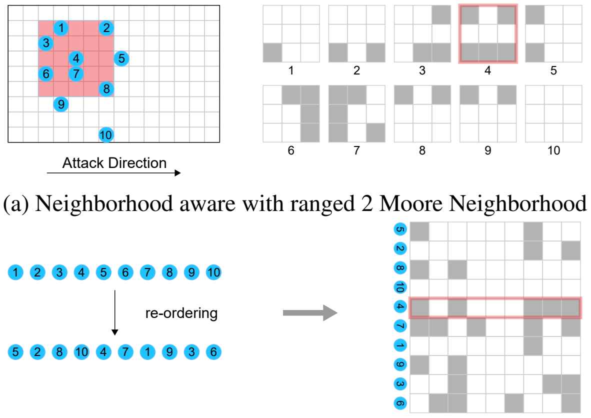
Given the preprocessed position of the players, the most direct and intuitive way to represent the position of the players is using a parameterized pitch. The soccer pitch is parameterized into $(W + 1) \times (H + 1)$ grids and the position of the players are normalized into a $[0, W] \times [0, H]$ area of integer. However, with the data dimension of $(W+1)\times(H+1)$, only 10 grids have value one, and the rest of them are filled with zeros. The sparsity of the representation affects the result of segmentation and classification. With a scaling rate s, the size of the pitch is scaled into $\lceil W/s\rceil \times \lceil H/s\rceil$, while the position of the players are scaled into $(\lceil x/s - 0.5\rceil, \lceil y/s - 0.5\rceil)$ (rounding). The scaling rate depicts the occupancy area of

[1] http://www.theifab.com/laws



(a) Range 1 (b) Range 2

Figure 2: An illustration of Moore Neighborhood



(a) Neighborhood aware with ranged 2 Moore Neighborhood

(b) flatten and x-coordinate first sorting

Figure 3: An illustration of formulating the neighborhood aware representation

a player, and a larger scaling rate means a larger occupancy area of the players. In our experiment, we set $W = 106$ and $H = 69$ as a base line, and explore how the scaling rate s affects the results.

Neighborhood Aware Representation

To capture the relative position of the players, we borrow the idea of moore neighborhood in cellular automata theory. The original definition of the moore neighborhood is the cellular itself and the surrounding eight cells, as Figure 2 shows. The mathematical definition of the Moore neighborhood with a radius k is

$$\{c_{ij} | \|c_{ij} - c\|_{L_\infty} < k\}, \tag{2}$$

where c_{ij} is the cell on the i-th row and the j-th column ,c is the center of the Moore neighborhood and L_∞ is the Chebeshev distance.

Combined with the traditional representation of different scaling rate, we further develop a neighborhood aware representation (NAR), which also takes the offensive direction of the team into consideration. As Figure 3a shows, assuming the team is attacking from left to right, the ranged 2 moore neighborhood indicates 9 relative positions of the player 4. The player 1 (and 3), 2, 6, 7 and 8 are at the LB, LF, RB, Right and RB positions of the player 4 respectively. After extracting the NAR of all players, we form a 10×9 neighboring graph. As the formation depicts the relative position along the touchline of the players (the backwards will stay closer to the back-court than the forwards), the layout of neighboring graph should be consistent with this characteristic. An $x-$coordinate first sorting is operated on the position of the players and the neighboring graph is re-organized by the order, as Figure 3b shows.

3.2 Temporal Segmentation and Detection

We explore a new method to segment and detect the formation given the representation of the player positions. A formation is not only the relative position of the players on the pitch, but also the evolution of their positions. According to the domain experts, when a team obtains the ball at the back court, the defensive formation will change into an advancing formation, trying to control the midfield and seeking a chance to pass the ball forward. If the team goes deep into the front court, the formation will become more aggressive even disappear, because the strikers have much more freedom to choose their positions for tearing off the defense of the opponents and creating chaos in front of the goal. The boundary between two formations is consequently vague, with variation in some way. Therefore, the model should be able to learn the temporal patterns of the features but should not divide the time into two segments when the position of the players changes slightly, like the text segmentation according to semantic differences instead of the variation in description. The problem is similar to the human action segmentation and detection problem but is more challenging. On the one hand, the boundaries between human actions are clearer. On the other hand, many works on image feature extraction and physical detection have laid the foundation for the derived works. Referring to the general framework in this area, the feature sequences extracted from the videos will be fed into a set of classifiers. Traditionally, Recurrent Neural Networks (e.g. LSTMs and GRUs) achieve good performances on the sequential data, benefiting from their recurrent structure. Recently, [Bai *et al.*, 2018; Lea *et al.*, 2017] showed that temporal convolutional networks have the potential to compete with RNNs on sequential data.

Temporal Convolution

Temporal convolution is a kind of 1D convolution, which means that the filters only convolve temporally. As Figure 4 shows, after convolving by a $d \times l$ filter with stride one and zero padding, the input sequence in shape $t \times d$ is transformed into a $t \times 1$ sequential feature. Note that d is the dimension of input features and l is the perception area. The filter can capture the features within l units temporally.

In our case, given a sequence S_i, we have the representation $r_j^i \in \mathbb{R}^{10 \times 9}$. With the dot product nature of convolution, the representations can be flattened into vectors and the form becomes the same as example.

Application of ED-TCN

In our paper, we adopt the Encoder-Decoder temporal convolutional network (ED-TCN) by [Lea *et al.*, 2017] for segmentation and classification, whose architecture is as Figure 5 shows. The encoder with n layers convolves and captures the temporal patterns of the representation. The decoder has

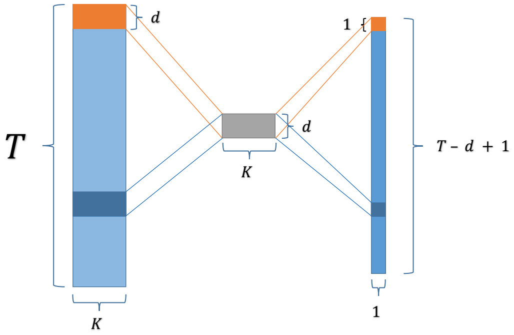


Figure 4: Temporal convolution

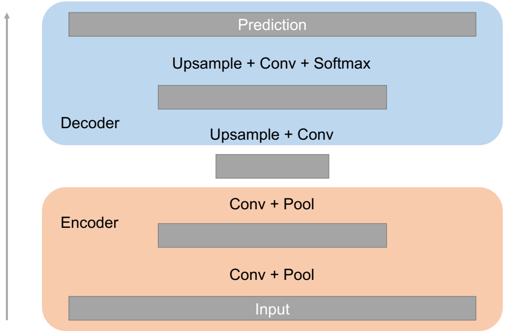


Figure 5: The architecture of the ED-TCN

the same number of layers, upsampling the high level embedding and classifying the formation. Every layer contains a set of temporal convolutional filters. The perception area of the model is $l(2^n - 1) + 1$. The model shows good performance on segmenting and classifying the human action using appearance features extracted from the videos.

3.3 Base Line Model

We adopt the model developed by [Bialkowski *et al.*, 2014b] as the base line. The model assumes that a team uses a stable formation within a period of time. Therefore, the roles on the pitch are stable and every moment each role is played by a player. A role is modeled with a 2D gaussian distribution. To obtain the final clustering, the model adopts a k-Means equivalence framework to compute the distributions. Iterating through the temporal data, player positions at each frame are assigned to the distributions bijectively using the Hungarian algorithm. After several iterations, the distributions come to convergence.

With a representation of distributions, the position of the players within a period is aggregated. The authors labeled the formation of the aggregations and performed agglomerative clustering on the them. The final agglomerations are marked as specific formations respectively, and with the results, formation detection is feasible given an unlabeled positional aggregation of the team within a period. The Figure 6 shows the framework of the model.

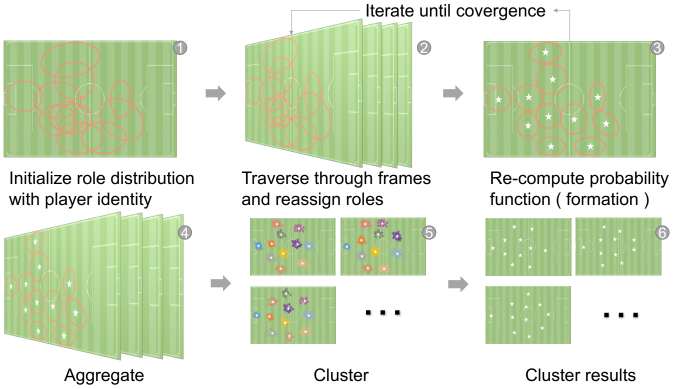

Iterate until covergence
1
2
3
Initialize role distribution with player identity
Traverse through frames and reassign roles
Re-compute probability function (formation)
4
5
6
Aggregate
Cluster
Cluster results

Figure 6: The framework of the base line model

4 Experiment

In this part, we firstly evaluated the parameter selection for feature extraction, including scaling rate, the range of the neighborhood and the number of the filters on each layer. Furtherly, we compared our result with the base line model to show the advantages of our model.

4.1 Implementation

The ED-TCN is implemented with Keras[Chollet and others, 2015] by [Lea *et al.*, 2017]. Based on the code, we made some modifications to support our data. We trained the ED-TCN on a NVIDIA 1080Ti.

We implemented the base line model by Python, with NumPy for matrix computation. Since the speed is not our focus, we do not use the GPU supported programming.

In our experiments, the data comes from a U15 game between Argentina and Brazil. The sequential positional data is truncated by the events during the game and the amount of the sequences is 294, the volume of which is comparable to the 50 Salads[Stein and McKenna, 2013]. To accelerate the training process, the sequences were downsampled temporally by 3. After downsampling, the longest sequence is 892 in length, with a feature dimension $k = 90$. Before fed into the ED-TCN, each sequence has been zero padded with mask.

4.2 Parameter Selection

Experiements on parameter selection focus on two aspects. The first aspect is about feature extraction, correlated to scaling rates and the range of NAR. In our experiment, the scaling rate ranges from 2 to 10 and NAR ranged 1 and 2. The second aspect is the number of the filters. Referring the set up of the ED-TCN, we set two combinations of the filters of the layers, (64, 96) and (96, 128). Our exploration on deeper networks did not get a good feedback. To evaluate the effect of the representation, we define a variable to measure the information density of the representation. For a representation $r \in \mathbb{R}^k$, information density

$$d(r) = \frac{\sum_{i=0}^{k} r_i}{k \cdot Max(r)}, \quad (3)$$

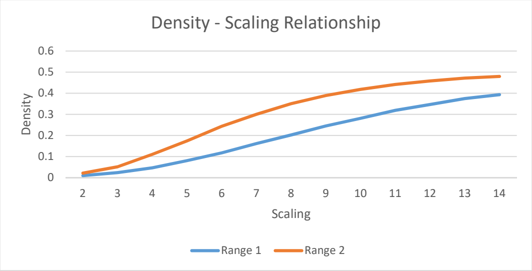


Figure 7: Information Density vs Scaling Rate

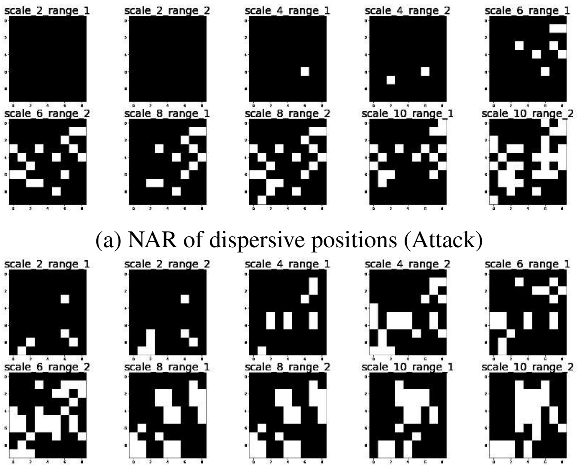
Scaling	Range	Density	(64,96)	(96,128)
2	1	0.009	44.86	41.85
	2	0.021	50.41	33.11
3	1	0.023	48.06	49.15
	2	0.051	37.84	39.16
4	1	0.046	39.05	39.69
	2	0.110	54.9	60.24
5	1	0.080	42.48	27.45
	2	0.174	41.14	34.45
6	1	0.117	41.44	55.15
	2	0.243	52.07	58.41
7	1	0.161	59.51	43.01
	2	0.300	63.8	50.32
8	1	0.202	32.15	38.54
	2	0.350	48.63	52.8
9	1	0.245	**66.8**	**59.24**
	2	0.389	52.95	49.0
10	1	0.281	57.83	53.08
	2	0.418	46.95	55.8

Table 1: Results of Parameter Selection

where $Max(r)$ means the maximum value in representation r. The larger the sampling rate, the information density is relatively larger. Theoretically, $\lim_{s\to\infty} d(r_i(s)) = 1$. In our case, when s is larger than 105, the informaiton density will reach 1.

The results are shown in Table 1. The information density in the table is the average density of the NAR with relative sampling rate and range. As the sampling rate increases from 1, the information density increases, but as the sampling rate gets larger, the increase of information density gets slower (Figure 7). From the table, we can find that the representations with higher scaling rate get better results. When the information density reaches 0.25, the accuracy increases obviously.

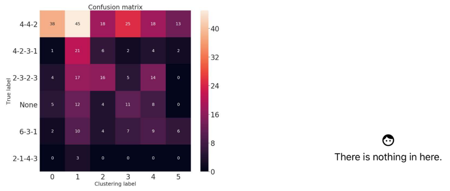
Soccer is a highly dynamic sport, and the relative distance between the players change intensely. A proper scaling rate and range is crutial to capture the relative position of the players. As Figure 8 shows, the NAR with low scaling rate is unable to capture the relative postion of the players when they get far away from each other, especially under the attack situation. However, it is not the fact that the larger the scaling rate, the larger the information density and the better the effect of the representation.



(a) NAR of dispersive positions (Attack)

(b) NAR of tight positions (Defense)

Figure 8: The NAR under different density of player positions



(a) baseline model

(b) ED-TCN

Figure 9: Comparison between two models

4.3 Control Experiment

In this section, we compare the results obtained by our model with the results by base line model. The base line model is based on the stability assumption of formation, generating only one representation of the formation every half game and experimenting on a full season data of the English Premier League. To parallelly compare the two models, we subdivide the sequential data into 330 smaller multiples, and each multiple shares a unified label for the ease of clustering. We use the ED-TCN model trained with a scaling rate 7 and ranged 1 NAR and predict the subdivided sequences.

To get the prediction result, all of the data is used for clustering. Given the agglomerations, the result is as Figure 9a assuming that the corresponding relation is built according to maximizing the total correctness. The vertical labels represent the groundth and the horizontal lables represent the result of agglomerative clustering. Cluster 0 is the agglomeration of formation 4-4-2, with an ratio of 76%, but the accuracy of other agglomerations is lower than 50%. The accuracy of the base line model on the total set is 29.79% and the $f1$ score is 23.47%, which is relatively smaller than the accuracy.

The sequences are seperated into 60% for training, 40% for validation and 10% for testing. The ED-TCN converges after 500 epoches. We tested on the model with 25, 50 and 100 filter length respectively. The evaluation of the ED-TCN is different from the base line model. Every object in the sequence will get a prediction. Evaluated on the testing data, we obtained a 49.02% accuracy, with a 14.57% f1 score. The ED-TCN model has low performance

Discussion: The base line model is not suitable for formation detection in a dynamic situation. Though the sequences are subdivided into smaller multiples with unified label, the data is not clean enough for unsupervised learning. To eliminate the spatio variation, the base line model normalizes the average position of raw data to the origin by a frame-wise mean substraction. The temporal variation is modeled by 2D gaussian distribution. Due to the human variation, the dominant formation of the labeled data within a period is correct, but abrupt changes exist. When the team loses the ball, the player nearby will snatch the ball on the spot. At the same time, the formation will start to transform into a defensive one, which takes time. The player may successfully get the ball back and the transformation will stop suddenly and get back to an offensive state. The variation is ignored by the annotators, which is unfriendly to the unsupervised clustering. The distribution representation is not robust to the variation of the sequence length because a longer sequence usually have larger temporal variation.

5 Conclusion

References

[Bai *et al.*, 2018] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018.

[Bialkowski *et al.*, 2014a] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, and Iain Matthews. Win at home and draw away: Automatic formation analysis highlighting the differences in home and away team behaviors. In *Proceedings of 8th annual MIT sloan sports analytics conference*, pages 1–7. Citeseer, 2014.

[Bialkowski *et al.*, 2014b] Alina Bialkowski, Patrick Lucey, Peter Carr, Yisong Yue, Sridha Sridharan, and Iain Matthews. Large-scale analysis of soccer matches using spatiotemporal tracking data. In *2014 IEEE International Conference on Data Mining (ICDM)*, pages 725–730. IEEE, 2014.

[Chollet and others, 2015] François Chollet et al. Keras. https://keras.io, 2015.

[Dearden *et al.*, 2006] Anthony Dearden, Yiannis Demiris, and Oliver Grau. Tracking football player movement from a single moving camera using particle filters. 2006.

[Gehring *et al.*, 2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

[Kalchbrenner *et al.*, 2016] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aäron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *CoRR*, abs/1610.10099, 2016.

[Lea *et al.*, 2017] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[Stein and McKenna, 2013] Sebastian Stein and Stephen J McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 729–738. ACM, 2013.

[Wei *et al.*, 2013] Xinyu Wei, Long Sha, Patrick Lucey, Stuart Morgan, and Sridha Sridharan. Large-scale analysis of formations in soccer. In *Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on*, pages 1–8. IEEE, 2013.

[Wu *et al.*, 2019] Yingcai Wu, Xiao Xie, Jiachen Wang, Dazhen Deng, Hongye Liang, Hui Zhang, Shoubin Cheng, and Wei Chen. Forvizor: Visualizing spatio-temporal team formations in soccer. *IEEE transactions on visualization and computer graphics*, 25(1):65–75, 2019.