Fall Detection using NAO Robot Pose Estimation in RoboCup SPL Matches

Cristian Zampino¹, Flavio Biancospino¹, Michele Brienza¹, Francesco Laus¹, Gianluca Di Stefano¹, Rocchina Romano¹, Andrea Pennisi¹, Vincenzo Suriani² and Domenico Daniele Bloisi*¹*

¹Department of Mathematics, Computer Science, and Economics, University of Basilicata, Potenza, Italy ²Department of Computer, Control, and Management Engineering, Sapienza University of Rome, Rome, Italy

Abstract

RoboCup is an International robotics initiative whose aim is to promote robotics and AI research. RoboCup's long-term goal is to create a fully autonomous humanoid robot team capable of competing and winning a soccer game against the human World champion team, in compliance with the official rules of FIFA, by 2050. In this paper, we describe a two-step method for action recognition. In the first step, we extract the pose of the robots using a pose detector trained on a novel dataset for pose estimation called *UNIBAS NAO Pose Dataset*, which is a contribution of this work. In the second step, a Spatial-Temporal Graph Convolutional Network is used for modeling the gameplay, with particular regard to fall-down detection. Experimental results show the effectiveness of our approach in detecting falls for humanoid robots.

Keywords

Fall detection, NAO robot pose estimation, Robot soccer, Sports analytics

1. Introduction

Since its foundation in 1996, one of the objectives of RoboCup was to push the boundaries of research by offering high-level challenges. In 2022, the RoboCup Standard Platform League (concerning NAO robots playing soccer 5 vs 5) has proposed an Open Research Challenge [\[1\]](#page--1-0) with the goal of creating an autonomous system for generating game statistics from the videos of the matches (see Fig. [1\)](#page--1-1). Open Research Challenge's motivation lies in the need of augmenting the data available from the *Game Controller* software, which has proven to be insufficient to extract consistent statistics from a match [\[2\]](#page--1-2). The Open Research Challenge has two main goals:

- 1. A *short-term* goal, with the purpose of calculating the extrinsic camera parameters (camera matrix) from the camera feed and tracking the ball and the robots on the field;
- 2. A *long-term* goal, aiming at creating game statistics (including time under control, successful, unsuccessful shots on goal, passes, etc.) based on the located objects and positions.

*AIRO 2022 The 9th Italian Workshop on Artificial Intelligence and Robotics, held in conjunction with AI*IA 2022 The 21st International Conference of the Italian Association for Artificial Intelligence*

 Θ andrea.pennisi@unibas.it (A. Pennisi); suriani@diag.uniroma1.it (V. Suriani); domenico.bloisi@unibas.it (D. D. Bloisi)

Orcid [0000-0002-9081-0765](https://orcid.org/0000-0002-9081-0765) (A. Pennisi); [0000-0003-1199-8358](https://orcid.org/0000-0003-1199-8358) (V. Suriani); [0000-0003-0339-8651](https://orcid.org/0000-0003-0339-8651) (D. D. Bloisi) © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CEUR Workshop [Proceedings](http://ceur-ws.org) [\(CEUR-WS.org\)](http://ceur-ws.org)

Figure 1: Example of a RoboCup SPL match recorded with a single fixed fish-eye camera. Image from <www.youtube.com/RoboCupSPL>

UNIBAS Wolves and SPQR Team participated in the Open Research Challenge presenting MARIO, an end-to-end architecture for computing visual statistics in RoboCup SPL [\[3\]](#page-7-0). One of the features of MARIO is the *modular architecture*, which allows the user to customize the system for extracting statistics. In this paper, we focus on one of the MARIO's modules, used for detecting the robot fall-down action. The novelty of this work consists in combining a NAO robot pose estimation method with a graph convolutional neural network-based approach for recognizing falling robots. A further contribution is the development of the *UNIBAS NAO Pose* Dataset^{[1](#page-1-0)}, containing labeled images and videos of NAO robot poses.

The remainder of the paper is organized as follows. Section [2](#page-1-1) contains a description of existing methods for extracting statistics from soccer matches. Section [3](#page-2-0) presents the method for detecting the robot fall-down action, including the description of the dataset used for training. Quantitative experimental results are shown in Section [4.](#page-5-0) Finally, conclusions are drawn in Section [5.](#page-6-0)

2. Related Work

The analysis of soccer matches videos aims at:

- 1. Investigating opponent's strategies;
- 2. Examining the team's collective performance;
- 3. Calculating useful statistics designed to validate game schemes.

The goal of the analysis is to collect data to improve the quality of play so as to increase the chance of winning. At the basis of conducting a video analysis of soccer matches, it is essential

¹available at https://drive.google.com/drive/folders/1wY9Xsz3O_gYc4BbGb4p_gALotynjcH-E?usp=sharing

to identify the athlete, and by employing increasingly advanced technologies, we have gone as far as identifying the type of motion and detecting interactions with opponents.

Video analysis approaches can be grouped in two main categories. The first group includes commercial systems that are semi-automatic, meaning that the labeling of players is entered manually through user interaction. Semi-automatic commercial systems are used, for example, by TVs or professional teams to observe game situations and actions that occurred during matches. The second group is made of commercial and academic systems that are designed toward eliminating user interaction.

Semi-automatic systems include Piero [\[4\]](#page-7-1) and Viz Libero [\[5\]](#page-7-2), which are capable of computing many different statistics (distances, trajectories, heatmaps, free spaces, interaction spaces, etc.). Among the group of fully-automatic systems, the system proposed by Stein et al. [\[6\]](#page-7-3) presents the labeling and tracking of players in an automatic way.

The use of supervised algorithms is a common characteristic of fully-automatic approaches. For training those systems it is crucial to collect a large dataset containing properly annotated samples. For example, SoccerNet [\[7\]](#page-7-4) is a large-scale dataset for soccer video understanding. It can be used to perform action recognition by individuating different categories of events, e.g., goals, substitutions, and yellow and red cards. An extension of SoccerNet is SoccerNet-v2 [\[8\]](#page-7-5) that includes 17 categories of actions with detection statistics every twenty-five seconds (a further improvement when compared to the 8.7 actions per hour in SoccerNet). SoccerDB [\[9\]](#page-7-6) is a soccer database similar to SoccerNet but enhanced by the use of bounding boxes for individual player detection.

In this work, we present MARIO [\[3\]](#page-7-0) a fully-automatic system specifically designed for analysing NAO soccer robot matches. MARIO ranked first, ex-aequo with the B-Human Team's system, in the Open Research Challenge at RoboCup 2022. Robot and ball tracking in MARIO are done automatically, using a combination of traditional [\[10\]](#page-7-7) and deep learning [\[11\]](#page-7-8) based computer vision methods. Game analysis can extract trajectories, passes made, and heatmaps through graphs and tables containing both traditional statistics and more advanced statistics within the field, such as falls and foul actions made by the robots. In particular, we describe two supervised algorithms included in MARIO. The first is devoted to the calculation of the poses of the NAO robots, while the second aims at detecting falling down events.

3. Proposed Method

The fall-down detection scheme includes:

- 1. A pose estimation module;
- 2. A graph convolutional neural network for detecting the movement.

Before describing our approach, it is convenient to introduce our dataset for NAO robot pose estimation, called *UNIBAS NAO Pose Dataset* [\[12\]](#page-7-9).

3.1. UNIBAS NAO Pose Dataset

NAO are humanoid robots, however, there are some differences between the NAO structure and the human one. Thus, we have created a specific dataset for detecting the pose of a NAO robot.

Figure 2: Skeleton key points used for describing the pose of a NAO robot.

In particular, we collected video frames from the RoboCup Standard Platform League (SPL) teams. Using the COCO Annotator tool [\[13\]](#page-7-10), we labeled 451 frames containing about 3,000 NAO robot instances in the well-known COCO format. All annotations share the following data structure. The pose is represented by up to 18 key points describing ears, eyes, nose, neck, shoulders, elbows, wrists, hips, knees, and ankles (see Fig. [2\)](#page-3-0). The annotations are stored using a JSON structure. UNIBAS NAO Pose Dataset is available at the following [link.](https://drive.google.com/drive/folders/1L_JNu6AYqcQWZVo-Q8nVVosUBb-DjXrV)

3.2. NAO Pose Estimation

To detect the pose of the NAOs we used a slightly optimized version of OpenPose, called Lightweight OpenPose [\[14\]](#page-7-11). In this version, the VGG feature extractor is replaced with a MobileNetV1 network and it is optimized for reaching real-time performance. In particular, all the layers are kept and a dilated convolution is used for saving the spatial resolution and reusing the backbone weights.

To produce a new estimation of keypoint heatmaps and Part Affinity Fields (PAFs), the refinement stage takes features from the backbone concatenated with the previous estimation of the keypoint heatmaps and PAFs. In order to share the computation between heatmaps and PAFs, a single prediction branch is used in the initial and refinement stage. All the layers are shared except for the last two, which directly produce keypoint heatmaps and PAFs.

To capture long-range spatial dependencies, each convolution with 7×7 kernel size is replaced by a convolutional block with the same receptive field. This block design allows having three consecutive convolutions with 1×1 , 3×3 , and 3×3 kernel size convolutional layers. To preserve the initial receptive field the last one has a dilation parameter equal to 2. Finally, a residual connection is used for speeding up the network. The final network has ∼2.5 times less complexity than convolution with 7×7 kernel.

Figure 3: ST-GCN architecture diagram.

Figure 4: Three examples of skeletons. **Left:** A rejected skeleton with 5 missing key points. **Center:** An accepted skeleton with no missing key points. **Right:** An accepted skeleton with 2 missing key points.

3.3. Action Detection

The detection of the fall-down action is carried out by using a Spatial-Temporal Graph Convolutional Network (ST-GCN). The ST-GCN model includes 9 layers of spatial-temporal graph convolution operators, namely ST-GCN units (see Fig. [3\)](#page-4-0).

The output size is 64 for the first three layers, 128 for the following three layers, and 256 for the last three layers. A dropout layer with a probability of 0.5 is present after each ST-GCN unit to prevent overfitting. The strides of the fourth and seventh temporal convolution layers are set to 2 as pooling layers. After the 9 ST-GCN units, a global pooling is performed on the resulting tensor to get a 256-dimension feature vector for each sequence. Then, the 256-feature vector is fed to a softmax classifier. Stochastic gradient descent with a learning rate of 0.01 is used as optimization method. The learning rate is programmed to decay by 0.1 every 10 epochs.

ST-GCN takes as input the output produced by the pose estimator. Since the trained ST-GCN requires 14 skeleton key points, the 18 key points coming from the NAO pose estimator are modified by removing the key points representing ears and eyes. The input for the ST-GCN model is a sequence of 30 skeletons. It is worth noticing that a skeleton is not included in the sequence if the number of key points returned by Lightweight OpenPose does not reach 80% of the expected total (see Fig. [4\)](#page-4-1).

Figure 5: Pose estimation for multiple humanoid NAO robots in different RoboCup SPL matches.

The ST-GCN model outputs the action class with the related probability score. If the probability score is greater than 50%, the action class produced by the model is considered valid, otherwise, it is rejected. For training the model, we used the *Le2i* data set proposed by Charfi *et al.* [\[15\]](#page-7-12), which includes 192 video sequences containing human fall actions.

4. Experimental Results

The performance of the proposed approach has been measured by considering separately the pose estimation module and the fall detection system.

4.1. Pose Estimation Results

Fig. [5](#page-5-1) shows the pose estimation results on four sample frames extracted from different RoboCup matches. A video showing more pose estimation results is available at [https://youtu.](https://youtu.be/QhtPp6-n3hM) [be/QhtPp6-n3hM](https://youtu.be/QhtPp6-n3hM)

In order to quantitatively evaluate the Lightweight OpenPose performance, we considered the percentage of correctly classified key points. Specifically, we computed an overall percentage of valid key points of the video input. In our calculation, we measured a percentage of 70% correctly classified key points.

4.2. Fall Detection Results

Fig. [6](#page-6-1) shows a sequence where a falling robot is correctly detected. A video showing the results of the fall-down detection system is available at <https://youtu.be/a9rZD97vOT4>

Figure 6: Fall detection. First, Lightweight OpenPose successfully detects the robot's pose. Then, the fall action is captured by the combined use of ST-GCN and the aspect ratio of the bounding box.

Table 1

Quantitative results for the fall detection system.

Table [1](#page-6-2) shows the experimental results. For evaluating the ST-GCN model, we performed several experiments using the UNIBAS NAO Pose Dataset. We calculated the number of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) detections to compute several metrics, namely Accuracy, Sensitivity, and Specificity.

4.3. Discussion

The overall performance of the system is good. However, there are some errors. The accuracy percentage of 70% in detecting the correct key points might be due either to the presence of occlusions, which means that the robots are not correctly recognized, or an unusual robot pose that is not included in the data set. In some cases, the robots are not completely facing the camera, but instead are placed to the side of the camera.

The camera position and the robot position also affect the ST-GCN system for fall-down detection. Such a problem is related to the training data set, where the annotations are taken from a different point of view with respect to the one used during the RoboCup matches.

5. Conclusions

We have presented an approach for detecting the fall-down action for NAO in RoboCup SPL matches. In particular, the approach is based on two steps: 1) the estimation of the NAO pose by using Lightweight OpenPose and 2) the detection of the fall-down action by using a Spatial-Temporal Graph Convolutional Network (ST-GCN).

For evaluating the pose estimation performance, we have conducted quantitative experiments using a novel data set called *UNIBAS NAO Pose Dataset*. For training the ST-GCN, we used publicly available data from *Le2i* dataset, while for testing it we considered a set of labeled videos of Robocup matches. The proposed method proves its effectiveness in detecting the fall-down action and provides a few false positives due to the camera position and the occlusions generated by the other robots or the referee.

As future work, we intend to create a new data set for NAO action recognition and we want to increase the number of actions recognized by the system.

References

- [1] RoboCup Technical Committee, RoboCup SPL (NAO) Rule Book, 2022.
- [2] H. Mellmann, B. Schlotter, P. Strobel, Data driven research and development in robocup collection, organization and analysis of robocup game data, 2018.
- [3] D. D. Bloisi, A. Pennisi, C. Zampino, F. Biancospino, F. Laus, G. Di Stefano, M. Brienza, R. Romano, Mario: Modular and extensible architecture for computing visual statistics in robocup spl, 2022. URL: [https://arxiv.org/abs/2209.09987.](https://arxiv.org/abs/2209.09987)
- [4] Piero sport graphics, 2022. URL: [https://www.rossvideo.com/products-services/](https://www.rossvideo.com/products-services/acquisition-production/cg-graphics-systems/piero-sports-graphics-analysis/) [acquisition-production/cg-graphics-systems/piero-sports-graphics-analysis/.](https://www.rossvideo.com/products-services/acquisition-production/cg-graphics-systems/piero-sports-graphics-analysis/)
- [5] Viz libero, 2022. URL: [https://www.vizrt.com/products/viz-libero.](https://www.vizrt.com/products/viz-libero)
- [6] M. Stein, H. Janetzko, A. Lamprecht, Thorsten, Breitkreutz, P. Zimmermann, B. Goldlucke, T. Schreck, G. Andrienko, M. Grossniklaus, D. A. Keim, Bring it to the pitch: Combining video and movement data to enhance team sport analysis (2017).
- [7] Dataset soccernet, 2022. URL: [https://www.soccer-net.org/home.](https://www.soccer-net.org/home)
- [8] A. Deliège, A. Cioppa, S. Giancola, M. J. Seikavandi, J. V. Dueholm, K. Nasrollahi, B. Ghanem, T. B. Moeslund, M. V. Droogenbroeck, Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos, 2020.
- [9] Y. Jiang, K. Cui, L. Chen, C. Wang, C. Xu, SoccerDB: A large-scale database for comprehensive video understanding, 2020.
- [10] D. D. Bloisi, A. Pennisi, L. Iocchi, Parallel multi-modal background modeling, Pattern Recognition Letters 96 (2017) 45 – 54.
- [11] D. Albani, A. Youssef, V. Suriani, D. Nardi, D. D. Bloisi, A deep learning approach for object recognition with nao soccer robots, in: RoboCup 2016: Robot World Cup XX, Springer International Publishing, 2017, pp. 392–403.
- [12] UNIBAS WOLVES, UNIBAS NAO Pose Dataset, 2022. URL: [https://drive.google.com/drive/](https://drive.google.com/drive/folders/1wY9Xsz3O_gYc4BbGb4p_gALotynjcH-E?usp=sharing) [folders/1wY9Xsz3O_gYc4BbGb4p_gALotynjcH-E?usp=sharing.](https://drive.google.com/drive/folders/1wY9Xsz3O_gYc4BbGb4p_gALotynjcH-E?usp=sharing)
- [13] J. Brooks, COCO annotator, 2019. URL: [https://github.com/jsbroks/coco-annotator.](https://github.com/jsbroks/coco-annotator)
- [14] D. Osokin, Real-time 2d multi-person pose estimation on cpu: Lightweight openpose (2018).
- [15] I. Charfi, J. Miteran, J. Dubois, M. Atri, R. Tourki, Definition and performance evaluation of a robust svm based fall detection solution, in: 8th Int. Conf. on Signal Image Technology and Internet Based Systems, 2012, pp. 218–224.