# TORSO-21 Dataset: Typical Objects in RoboCup Soccer 2021

Marc Bestmann, Timon Engelke, Niklas Fiedler, Jasper Güldenstein, Jan Gutsche, Jonas Hagge, and Florian Vahl*

Hamburg Bit-Bots, Department of Informatics, Universität Hamburg, Vogt-Kölln-Straße 30, 22527 Hamburg, Germany
{bestmann, 7engelke, 5fiedler, 5guelden, 7gutsche, 5hagge, 7vahl}@informatik.uni-hamburg.de
https://robocup.informatik.uni-hamburg.de

**Abstract.** We present a dataset specifically designed to be used as a benchmark to compare vision systems in the RoboCup Humanoid Soccer domain. The dataset is composed of a collection of images taken in various real-world locations as well as a collection of simulated images. It enables comparing vision approaches with a meaningful and expressive metric. The contributions of this paper consist of providing a comprehensive and annotated dataset, an overview of the recent approaches to vision in RoboCup, methods to generate vision training data in a simulated environment, and an approach to increase the variety of a dataset by automatically selecting a diverse set of images from a larger pool. Additionally, we provide a baseline of YOLOv4 and YOLOv4-tiny on this dataset.

**Keywords:** Computer Vision · Vision Dataset · Deep Learning.

## 1 Introduction

In recent years, similar to other domains, the approaches for computer vision in the RoboCup soccer domain moved nearly completely to deep learning based methods [2]. Still, a quantitative comparison between the different approaches is difficult, as most approaches are evaluated using their custom-made dataset. The presented performance of the approaches is therefore not only related to its detection quality but also the specific challenge posed by the used dataset. Especially, if images are only from a single location or without natural light, they can hardly be an indicator for actual performance in a competition.

Outside of the RoboCup domain, this problem is addressed by creating standardized datasets for various challenges in computer vision [7,9,20,32]. These datasets are used as a benchmark when comparing existing approaches with each other, allowing a quantitative evaluation (e. g. [4]). Tsipras et al. investigated how well results of evaluations with the ImageNet dataset [7] reflect the

* All authors contributed equally.

performance of approaches in their actual tasks [31]. They observed that in some cases, the scores achieved in the ImageNet challenge poorly reflect real-world capabilities. In the RoboCup domain, participants are challenged with restricted hardware capabilities resulting in computer vision approaches specifically designed for the given environment (e. g. [28]). Thus, existing datasets are even less applicable to evaluate the vision pipelines designed for RoboCup soccer.

We propose a standardized dataset for the RoboCup Humanoid Soccer domain consisting of images of the Humanoid League (HSL) as well as the Standard Platform League (SPL). We provide two image collections. The first one consists of images from various real-world locations, recorded by different robots. It includes annotations for the ball, goalposts, robots, lines, field edge, and three types of line intersections. The second collection is generated in the Webots simulator [22] which is used for the official RoboCup Virtual Humanoid Soccer Competition[1]. Additionally to the labels of the first collection, labels for the complete goal, crossbar, segmentation images for all classes, depth images, 6D poses for all labels, as well as the camera location in the field of play, are provided. For both collections, we give a baseline using YOLOv4 [4].

Most of the existing popular image datasets are only designed to compare image classification approaches. In RoboCup Soccer, object localization, as well as segmentation, are also commonly used (see Table 1).

While the creation and sharing of datasets were already facilitated by the ImageTagger platform [13], it did not help increase the comparability of vision pipelines since teams use different parts of the available images. Furthermore, many teams published the datasets that they used in their publications (see Table 1). Still, none of these papers have compared their work directly to others.

While this lack of using existing datasets could simply result from missing knowledge about their existence, since they are often only mentioned briefly as a side note in the publications, this is not probable. In our experience, we chose to create a new dataset for our latest vision pipeline publication [14] since the other datasets did not include the object classes required. Another issue is a lack of variety in some sets, e. g. only including the NAO robot or being recorded in just one location. Furthermore, the label type of the dataset may also limit its uses, e. g. a classification set is not usable for bounding box based approaches.

The remainder of this paper is structured as follows: Our methods of image collection and annotation are presented in Section 2 and Section 3 respectively. We evaluate and discuss the proposed dataset in Section 4 followed by a conclusion of our work in Section 5.

## 2 Image Collection

The dataset presented in this work is composed out of images recorded in the real world as well as in simulation using the Webots simulator. In the following, we describe the methods of image collection and also our method to reduce the number of similar images for greater variety in the dataset.

[1] https://humanoid.robocup.org/hl-2021/v-hsc/ (last accessed: 2021/06/14)

**Table 1.** Comparison of approaches to vision in RoboCup Humanoid Soccer leagues. Detection types are abbreviated as follows: classification (C), bounding box (B), segmentation (S), keypoints (K). Detection classes are abbreviated as follows: ball (B), goal (G), goalpost (P), field (F), robot (R), obstacles (O), lines (L), line intersections (I). The ∘ sign means the data is publicly available, but the specific dataset is not specified. (✓) means it is partially publicly available. The sources are as follows: ImageTagger (IT), SPQR NAO image dataset (N), self created (S), not specified (?). The Locations are competition (C), lab (L), and not specified (?).

| Approach | | | | | Dataset | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Year | Paper | League | Detection Type | Classes | # Images | Synthetic | Source | Public | Location |
| 2016 | [1] | SPL | C | B,G,R | 6,843 | × | N | ✓ | ? |
| | [10] | HSL | B,C,K | R | 1,500 | × | ? | ? | ? |
| | [25] | HSL-K | S | B | 1,160 | × | S | × | ? |
| 2017 | [24] | HSL-A | K | B,I,P,O,R | 2,400 | × | S,YouTube | × | C,L |
| | [6] | SPL | C | R | 6,843 | × | N | ✓ | ? |
| | [17] | SPL | C | B,F,P,R | 100,000 | ✓ | S | ? | – |
| | [21] | SPL | C | B | 16,000 | × | S | × | ? |
| | [18] | HSL | S | R | 4,000 | × | S,N | ✓ | ? |
| 2018 | [27] | SPL | S | B,R,P,L | syn: 5,000 real: 570 | (✓) | S | × | C,L |
| | [12] | SPL | C | B | 40,756 | × | S | × | C,L |
| | [15] | HSL-T | B,S | B,F | ? | × | S | ? | ? |
| | [8] | HSL-K | S | B | 1,000 | × | IT | ✓ | C |
| | [11] | HSL-A | K | B,P,R | 3,000 | × | (IT) | (∘) | ? |
| | [26] | HSL-K | S | B | 35,327 | × | IT | ✓ | C,L |
| 2019 | [19] | HSL-A | K | B | 4,562 | × | S | ✓ | C,L |
| | [30] | HSL-K | B | B | 1,000 | × | S | × | C,L |
| | [16] | HSL-K | C | B,P | ? | × | Rhoban Tagger | ∘ | C |
| | [23] | SPL | B | R | syn: 28,000 real: 7,000 | (✓) | IT, SimRobot | ✓ | C,L |
| | [3] | HSL-K | B | B,P | 1,423 | × | S, IT | ✓ | C |
| | [14] | HSL-K | B,S | B,F,L,O,P,R | ? | × | IT | ✓ | C |
| | [28] | SPL | B | B,I,P,R | syn: 6,250 real: 710 | (✓) | Unreal Engine, S | ✓ | C,L |
| 2021 | [5] | SPL | B | B,R,I | ? | (✓) | N, S | ✓ | ? |
| | [29] | SPL | B,S | B,P,R,L,I | syn: 6,250 real: 3,000 | (✓) | Unreal Engine, S | ✓ | C,L |
| | Ours | HSL-K SPL | B,S,K | B,P,F,R,L,I,(G) | syn: 10,000 real: 10,464 | (✓) | IT, Webots | ✓ | C,L |

## 2.1 Reality

To create a diverse dataset, we collected images from multiple sources. First, from our recordings during different RoboCup competitions and our lab. Second, we investigated the data other teams uploaded publicly to the ImageTagger. Finally, we asked other teams to provide images especially from further locations, and for situations that were not already represented in the existing images. While this provided a large set of images, most of them had to be excluded to prevent biasing the final dataset. First, the number of images from the SPL was limited, as these only include the NAO robot and this could have easily lead to an over-

representation of the robot model. Many imagesets were excluded to limit one of the following biases: the ball is always in the image center, the camera is stationary, there are no robots on the field, other robots are not moving, or the camera always points onto the field. Generally, the selection focus was on including images that were recorded by a robot on a field, rather than images that were recorded by humans from the side of the field. Using images recorded by different teams is also crucial to include different camera and lens types, locations, and perspectives.

### 2.2 Simulation

As the RoboCup world championship in the HSL is held virtually in 2021, we deemed a data collection recorded in a simulated environment necessary. Diverse data can be generated as required. We chose the Webots simulator because it is used for the official competition. The official KidSize environment of the competition including the background, ball, goals, and turf as well as lighting conditions was used. During data generation, we used six robot models (including our own) which were published for this year's competition in the league.

Four setups are used per matchup of these robots. These vary by the robot team marker color and the team from whose perspective the image is captured. Scenes were generated in four scenarios per setup. In the first one, images are taken at camera positions uniformly distributed over the field. To prevent a bias of always having the ball included, we created a second scenario without a ball, but the same distribution of camera positions. Similar to the previous two, we also include two scenarios with the camera position normally distributed around a target on the field with and without a ball present. These last two scenarios imitate a robot that is contesting the ball.

We generated 100 images for each of the presented scenarios resulting in a total of: $\binom{6}{2} \cdot 2 \cdot 2 \cdot 4 \cdot 100$ images $= 24000$ images. The data is split into a 85% training and 15% test set.

For each image, a new scene is generated. The scenes are set up randomly by first sampling a target position. The ball is placed at the target position or out of sight depending on the scenario. Then the field players are placed by sampling from a normal distribution around the target position since it occurs often that multiple robots are grouped. To prevent robots from standing inside of each other, we resample the robot's position in the case of a collision. The heading of each robot is sampled from a normal distribution around facing the ball for the images where the ball is at the target position and from a uniform distribution otherwise. We assume each team to have a goalie, which stands on a random position on the goal line with its heading sampled from a normal distribution with the mean being the robot looking towards the field. The postures of the robots with our own robot model are each sampled from a set of 260 postures. These were recorded while the robot was performing one of six typical actions. The sampling is weighted by the estimated probability of an action occurring in a game (walking: 50%, standing: 20%, kicking: 10%, standup: 10%, falling: 5%,

fallen: 5%). We chose these samplings to cover the majority of situations a robot could realistically face in a match.

The camera position is sampled in the field either from a uniform distribution or from a normal distribution around the target position depending on the scenario. The camera floats freely in space instead of being mounted on a robot. We chose to do this to be able to simulate various robot sizes. Thus, edge cases such as the robot looking at its shoulder, are not included in this collection. The camera is generally oriented towards the target position. To avoid a bias towards the ball being in the center of the image, the camera orientation is offset so the position of the target position is evenly distributed in the image space.

Since the robot sizes (and thereby camera height) and fields of view (FOVs) are very different in the league, we decided to also model this in the dataset. We collected these parameters from the robot specifications from the last RoboCup competition. On this basis, we calculated the mean FOV and height as well as its standard deviations (FOV: $\mu = 89.3°, \sigma^2 = 28.1$, height: $\mu = 0.64\text{m}, \sigma^2 = 0.12$) for the HSL-KidSize. Based on this, for each image, we sample an FOV and a camera height from a normal distribution around the mean and with the standard deviation of the league. If the sampled FOV or height is smaller or larger than the extreme values used by a team (FOV: $60° - 180°$, height: $0.45\text{m} - 0.95\text{m}$), we resample for a new value.

### 2.3 Avoiding Similarity

How well a dataset represents a domain is not only related to its size but also the diversity between the images. In the Pascal VOC dataset [9] special attention was put on removing exact and near-duplicate images from the set of images to reduce redundancy and unnecessary labeling work. This approach worked well on their raw data taken from a photo-sharing website. However, the RoboCup domain poses additional challenges, as the images are typically recorded in a sequence. Therefore, most images are similar to the one before, since the robots only move slowly or are even standing (especially prevalent with goalkeepers). While a naive approach of taking every nth image can address this problem, it can also remove short events which rarely occur in the dataset. Additionally, the robots typically have some version of capture bias such as continuously tracking the ball or looking back to positions where they expect objects to be. Finally, the position of the robot on the field is not evenly distributed. Positions like the goal area, the center circle, and the sides of the field where the robots walk in are more commonly represented than for example the corners.

To avoid these issues, we used unsupervised machine learning to train a variational autoencoder. It was trained on a dataset consisting of low-resolution images ($128 \times 112$ pixels) from the various imagesets we decided to include. The autoencoder has $3,416,987$ trainable parameters and is based upon the *conv-vae*[2] GitHub repository using the original model architecture. We trained it to represent the images of this domain in a 300 dimensional latent space. To prune

[2] https://github.com/noctrog/conv-vae (last accessed: 2021/06/14)

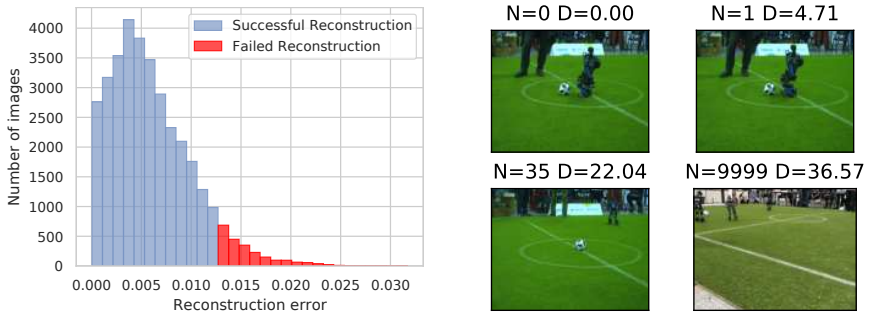


**Fig. 1.** Distribution of the reconstruction error from the variational autoencoder on the unfiltered dataset (**left**) and exemplary images with distance to a reference point in latent space (**right**). **D** describes the Euclidean distance in the latent space of the **N**'th distant neighbor of the reference image.

similar images, we used this latent space representation to remove images with close proximity to a given image. Neighbors within a given Euclidean distance were determined using a k-d tree. During the greedy sampling process, we start with the set $E$ containing all the unfiltered images and a k-d tree representing the latent space relations. An image is randomly selected from $E$ and all its close neighbors including the image itself are removed from $E$ while the sampled image itself is added to our filtered set $O$. We repeat this process until $E$ is empty and $O$ contains our filtered imageset. This algorithm is based on the assumption that the variational autoencoder can represent a given image in its latent space. This may not be the case for edge cases. Therefore we check the reconstruction performance of the autoencoder on a given image by comparing the original image against the decoder output and calculating the mean squared error between both of them. Outliers with an error of more than $1.64\sigma$ (which equals 10% of the dataset) are added to $O$ regardless of their latent space distance to other images. The error distribution is shown in Figure 1. Since a high error implies that a situation is not represented significantly in our existing dataset to be encoded into the latent space, it is assumed to be sufficiently distinct from the other images in the set. To filter our real-world dataset, we used 44,366 images as an input to this selection algorithm and reduced it to 10,464 images.

## 3 Image Annotation

In the following, we define the label types we used in the dataset. Additionally, we explain the process of image annotation for both the images gathered in the real world and in simulation. We provide labels for the classes ball, robot, goalpost, field area, lines, T-, L-, and X-line intersections. Only features that are relevant for a robot in a RoboCup Soccer game were labeled. Thus, no balls or robots outside of the current field and no other fields are labeled. Parts of the recording robot, e. g. its feet, are not labeled. Additionally, each label might be marked

as concealed or blurred. Concealed means that the object is partially covered by another object that is in front of it. Labels of objects, that are truncated as they are located on the border of the image, are not marked as concealed. The exception to this are the line crossings, they are concealed if they are not entirely visible. A blurred annotation is affected by either motion or camera blur, resulting in a significantly changed appearance of the object, e. g. a ball might appear oval rather than circular. A concealed or blurred object is significantly harder to detect. For example, this information could be used in the calculation of the loss function to specifically focus on also detecting blurred and concealed objects. It could also be used to focus on them less since a team might have no issues with motion blur because they use a different camera setup.

To avoid ambiguities, we define each object class in detail:
**Ball**: The ball is represented as a bounding box. It is possible to compute a near pixel-precise ellipse from the bounding box [26]. In some images, multiple balls are present on the field of play. We label all of them even though this would not occur in a regular game.
**Robot**: We define robot labels as a bounding box. Unlike the ball, it is not as easy to generate an accurate shape of the robot with just a bounding box, because the form of a robot is not as easy to define. However, to make labeling feasible, we compromise by using bounding boxes.
**Goalpost**: The label for goalposts is a four-point polygon. This allows to accurately describe tilted goalposts. Because the polygon encompasses the goalpost tightly, this method allows the computation of a segmentation image, the middle point, and the lowest point, which is required for the projection of the goalpost position from image space into Cartesian space. Only the goalposts on the goal line are labeled, excluding other parts of the goal.
**Field Area**: The field area is relevant as everything outside of the field provides no useful information for the robot. We define it with a series of connected lines, the ends are connected to the right and left borders of the image, assuming the end of the field area is visible there. A segmentation is computed from the area between the lines and the bottom of the image.
**Lines**: We offer a segmentation image for lines as the ground truth because there is no other option to annotate lines with sufficient precision, as their width in image space is highly variable.
**Field Features**: We define the T-Intersections, L-Intersections, and X-Intersections (including penalty mark and center point) of lines as field features. For this feature, we only define a single point in the center of the intersection.

To create labels for the real-world images, we used the ImageTagger. It provides all the necessary labeling types we used, other than for the annotation of lines. For these, we created a specialized tool. First, it allows the user to specify smoothing and adaptive threshold parameters for a given image. Based on this, a proposed segmentation mask is generated which can then be corrected manually. In the last step, all balls, robots, goalposts, and the area above the field are excluded from the line segmentation using the existing labels for these classes.

In general, features are only labeled when they are detectable by a human considering only the respective image (i. e. no context information from a previous image should be necessary). Some imagesets were already labeled for some of the classes. The rest of the labels were created by us. Additionally, we manually verified all labels in the set.

One of the main advantages of training data generation in simulated environments is the availability of ground truth data. Webots offers the functionality to generate bounding boxes and segmentation images for the objects present in a scene. Each individual object has a distinct color in the segmentation image. Furthermore, we provide bounding box annotations for the classes ball and robot, four-point polygons for goalposts and the goal top bar, and a single image coordinate for T-, L-, and X-intersections. Since the bounding boxes provided by Webots were inaccurate in some cases, we computed the minimum enclosing rectangle from the segmentation images. The bounding boxes for robots and the ball are calculated from this. For goalposts, we used rotated bounding boxes to account for the fact they may be tilted yielding similar annotations as the 4-point polygons used for manual annotation of real-world images. Line intersections were annotated by projecting their known positions into the image space of the camera. To detect whether they are visible, we verify in the segmentation image that lines are visible close to the intersection point. If the intersection is occluded, we still include it in the annotations, but mark it as "not in image". The remaining classes (i.e. goal, field area, and lines) are only provided in the segmentation images.

## 4 Evaluation

To evaluate the dataset, we performed a statistical analysis of the data and assess the performance of a YOLOv4 on it. We focus our evaluation on the real-world images since the images from simulation were generated as described in Section 2.2. The real-world dataset contains 10,464 images and 101,432 annotations in eight different classes. In Table 2 we showcase metadata about the images and annotations present in the collection. Figure 2 shows exemplary annotations on

**Table 2.** Detailed distribution of annotations (**left**) and general statistics (**right**) of the real-world collection.

| Annotation Type | Total | Annotations per Image | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 | 5+ |
| Ball | 10,959 | 46.6% | 49.6% | 2.9% | 0.9% | 0.0% | 0.0% |
| Goalpost | 12,780 | 46.8% | 31.4% | 21.6% | 0.3% | 0.0% | 0.0% |
| Robot | 14,383 | 64.2% | 18.6% | 7.6% | 4.3% | 2.1% | 3.1% |
| L-Intersection | 15,458 | 48.6% | 19.3% | 22.4% | 5.6% | 3.1% | 1.1% |
| T-Intersection | 13,479 | 46.1% | 35.1% | 12.7% | 3.5% | 1.8% | 0.9% |
| X-Intersection | 13,445 | 59.0% | 24.0% | 9.6% | 4.2% | 2.3% | 0.9% |
| Field Area | 10,464 | Segmentation Mask | | | | | |
| Lines | 10,464 | | | | | | |
| Sum | 101,432 | | | | | | |

| | |
|---|---|
| Ball Types | 6 |
| Locations | 12 |
| Camera Types | 5 |
| On Robot | 76% |
| During Game | 59% |
| Natural Light | 38% |
| League | |
| SPL | 25% |
| HSL | 75% |
| Perspective | |
| Field | 93% |
| Goalie | 4% |
| From Outside | 3% |

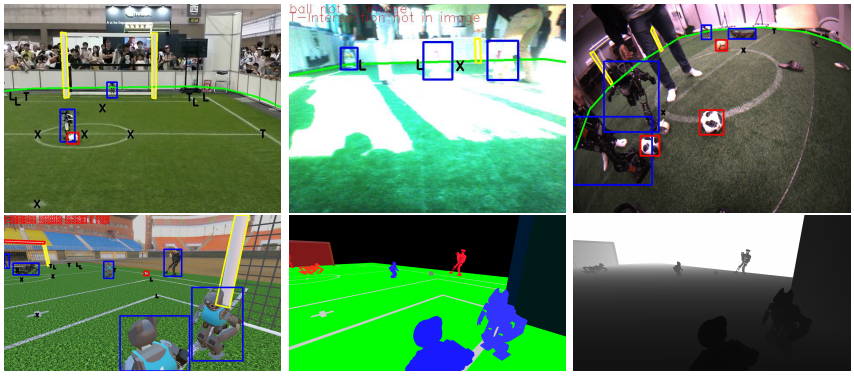

**Fig. 2.** Examples from the dataset. First row from the real-world collection, second row shows one image of the simulation collection with the corresponding segmentation and depth images.

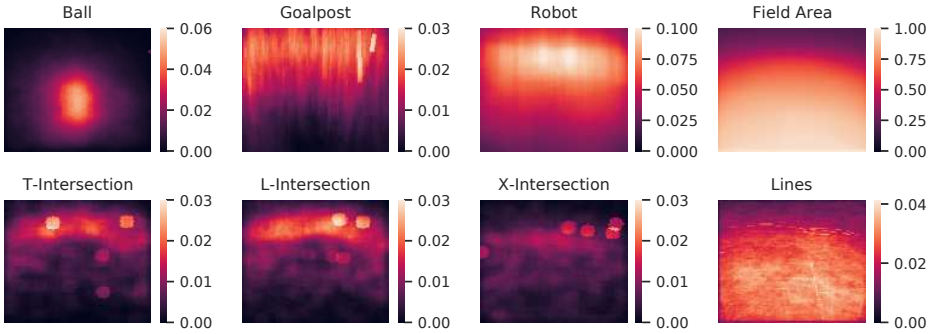


**Fig. 3.** Visualization of the position density of the respective annotations in the image space over all images of the real-world collection.

images from the dataset. Also, we investigated the positions of annotations in image space. This was done by plotting the heatmaps shown in Figure 3. Many of the patterns evident in the heatmaps are caused by the typical positions of the robots in the field and especially prominent by their head behavior as they are often programmed to look directly at a ball.

Based on metrics used in related work, we decided to present detection results using the mean average precision (mAP) and intersection over union (IoU) metrics. The IoU metric compares how well pixels of the ground truth and the detection overlap. Since the ball is round, but the labels are rectangular, we computed an ellipse in the bounding box as the ground truth for the IoU. Similarly, the line intersections are labeled as a single coordinate, but since the intersection itself is larger, we assumed a bounding box with height and width as 5% of the image height and -width respectively. In case of a true negative, we set the value of the IoU to 1. With the IoU, pixel-precise detection methods can achieve

**Table 3.** Mean average precision of YOLOv4 and YOLOv4-tiny on this dataset. For the intersections, we used a bounding box of 5% of the image size. The mAP values for the goalpost and crossbar are calculated from a bounding box that fully encompasses the polygon. The values are IOU, mAP with IOU threshold of 50%, mAP with IOU threshold of 75%. The floating point operations (FLOPS) required by YOLOv4 and YOLOv4-tiny per sample are 127 billion FLOPS and 6.79 billion FLOPS respectively.

| Environment | Approach | Metric | Ball | Goalpost | Robot | T-Int. | L-Int. | X-Int. | Crossbar |
|---|---|---|---|---|---|---|---|---|---|
| Real World | YOLOv4 [4] | IoU | 91.1% | 70.0% | 91.7% | 77.3% | 79.2% | 83.6% | - |
| | | mAP(50%) | 98.8% | 91.9% | 96.0% | 95.1% | 94.4% | 93.5% | |
| | | mAP(75%) | 89.7% | 54.9% | 72.7% | 23.6% | 23.8% | 23.1% | |
| | YOLOv4-tiny | IoU | 89.2% | 69.9% | 89.3% | 75.5% | 75.8% | 82.2% | - |
| | | mAP(50%) | 97.5% | 89.6% | 91.4% | 89.8% | 88.8% | 92.6% | |
| | | mAP(75%) | 80.0% | 42.9% | 47.7% | 43.3% | 39.7% | 38.9% | |
| Simulation | YOLOv4 | IoU | 88.5% | 51.2% | 87.2% | 70.5% | 69.3% | 78.9% | 58.1% |
| | | mAP(50%) | 92.1% | 94.2% | 93.7% | 97.9% | 97.2% | 98.6% | 89.5% |
| | | mAP(75%) | 84.6% | 76.4% | 82.7% | 86.7% | 87.1% | 91.1% | 66.2% |
| | YOLOv4-tiny | IoU | 85.1% | 51.4% | 82.5% | 63.1% | 60.9% | 73.2% | 58.9% |
| | | mAP(50%) | 80.4% | 91.0% | 83.5% | 89.8% | 85.8% | 91.7% | 91.2% |
| | | mAP(75%) | 59.5% | 64.8% | 57.8% | 55.4% | 50.6% | 63.4% | 59.2% |

higher scores than bounding box based approaches. The mAP metric classifies a detection as true positive if the ground truth and predicted bounding box have an IoU of at least e. g. 75%. It also represents how many of the individual objects were correctly found, especially when pixel-precise detection is less important. We present exemplary results of a YOLOv4 on the dataset in Table 3.

We would like to note that the dataset does not include images of the HSL AdultSize league. This is caused by the lack of available images and robot models for simulation. However, we expect the dataset to be still usable as a benchmark as the HSL KidSize and AdultSize leagues are visually very similar from a robot's perspective.

## 5 Conclusion

Efforts to share training data between teams have eased the transition to machine learning based approaches and were a good starting point for new teams. However, as we have shown in Table 1, many of the existing and new approaches were hard to compare quantitatively to each other as there was no common benchmark available. This work closes this gap by providing a benchmark dataset that is specific to the RoboCup Humanoid Soccer domain. Additional contributions of this paper are a system for vision training data generation in a simulated environment and an approach to increase the variety of a dataset by automatically selecting a diverse set of images from a larger pool.

The quality of the dataset is limited by the availability of images. Therefore, we hope that more teams start recording images on their robots during games and publish them so that future datasets can profit from this. Future datasets could include image sequences to allow detection of robots' actions, e.g. a kick, and include images of outdoor fields with real grass. This dataset could also be used as a qualification metric for future competitions.

The dataset and tools used to create it are available at https://github.com/bit-bots/TORSO_21_dataset.

## References


1. Albani, D., Youssef, A., Suriani, V., Nardi, D., Bloisi, D.D.: A deep learning approach for object recognition with nao soccer robots. In: Robot World Cup. pp. 392–403. Springer (2016)
2. Asada, M., von Stryk, O.: Scientific and technological challenges in robocup. Annual Review of Control, Robotics, and Autonomous Systems **3**, 441–471 (2020)
3. Barry, D., Shah, M., Keijsers, M., Khan, H., Hopman, B.: Xyolo: A model for real-time object detection in humanoid soccer on low-end hardware. In: International Conference on Image and Vision Computing New Zealand (IVCNZ). pp. 1–6. IEEE (2019)
4. Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
5. Cruz, N., Leiva, F., Ruiz-del Solar, J.: Deep learning applied to humanoid soccer robotics: playing without using any color information. Autonomous Robots pp. 1–16 (2021)
6. Cruz, N., Lobos-Tsunekawa, K., Ruiz-del Solar, J.: Using convolutional neural networks in robots with limited computational resources: detecting nao robots while playing soccer. In: Robot World Cup. pp. 19–30. Springer (2017)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE conference on computer vision and pattern recognition. pp. 248–255. IEEE (2009)
8. van Dijk, S.G., Scheunemann, M.M.: Deep learning for semantic segmentation on minimal hardware. In: Robot World Cup. pp. 349–361. Springer (2018)
9. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
10. Farazi, H., Behnke, S.: Real-time visual tracking and identification for a team of homogeneous humanoid robots. In: Robot World Cup. pp. 230–242. Springer (2016)
11. Farazi, H., Ficht, G., Allgeuer, P., Pavlichenko, D., Rodriguez, D., Brandenburger, A., Hosseini, M., Behnke, S.: Nimbro robots winning robocup 2018 humanoid adult-size soccer competitions. In: Robot World Cup. pp. 436–449. Springer (2018)
12. Felbinger, G.C., Göttsch, P., Loth, P., Peters, L., Wege, F.: Designing convolutional neural networks using a genetic approach for ball detection. In: Robot World Cup. pp. 150–161. Springer (2018)
13. Fiedler, N., Bestmann, M., Hendrich, N.: Imagetagger: An open source online platform for collaborative image labeling. In: Robot World Cup: XXII. pp. 162–169. Springer (2018)
14. Fiedler, N., Brandt, H., Gutsche, J., Vahl, F., Hagge, J., Bestmann, M.: An open source vision pipeline approach for robocup humanoid soccer. In: Robot World Cup: XXIII. pp. 376–386. Springer (2019)
15. Gabel, A., Heuer, T., Schiering, I., Gerndt, R.: Jetson, where is the ball? Using neural networks for ball detection at robocup 2017. In: Robot World Cup. pp. 181–192. Springer (2018)
16. Gondry, L., Hofer, L., Laborde-Zubieta, P., Ly, O., Mathé, L., Passault, G., Pirrone, A., Skuric, A.: Rhoban football club: Robocup humanoid kidsize 2019 champion team paper. In: Robot World Cup. pp. 491–503. Springer (2019)

17. Hess, T., Mundt, M., Weis, T., Ramesh, V.: Large-scale stochastic scene generation and semantic annotation for deep convolutional neural network training in the robocup spl. In: Robot World Cup. pp. 33–44. Springer (2017)
18. Javadi, M., Azar, S.M., Azami, S., Ghidary, S.S., Sadeghnejad, S., Baltes, J.: Humanoid robot detection using deep learning: A speed-accuracy tradeoff. In: Robot World Cup. pp. 338–349. Springer (2017)
19. Kukleva, A., Khan, M.A., Farazi, H., Behnke, S.: Utilizing temporal information in deep convolutional network for efficient soccer ball detection and tracking. In: Robot World Cup. pp. 112–125. Springer (2019)
20. 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. Springer (2014)
21. Menashe, J., Kelle, J., Genter, K., Hanna, J., Liebman, E., Narvekar, S., Zhang, R., Stone, P.: Fast and precise black and white ball detection for robocup soccer. In: Robot World Cup. pp. 45–58. Springer (2017)
22. Michel, O.: Cyberbotics ltd. webots™: professional mobile robot simulation. International Journal of Advanced Robotic Systems **1**(1), 5 (2004)
23. Poppinga, B., Laue, T.: Jet-net: real-time object detection for mobile robots. In: Robot World Cup. pp. 227–240. Springer (2019)
24. Schnekenburger, F., Scharffenberg, M., Wülker, M., Hochberg, U., Dorer, K.: Detection and localization of features on a soccer field with feedforward fully convolutional neural networks (fcnn) for the adult-size humanoid robot sweaty. In: Proceedings of the 12th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots, Birmingham. sn (2017)
25. Speck, D., Barros, P., Weber, C., Wermter, S.: Ball localization for robocup soccer using convolutional neural networks. In: Robot World Cup. pp. 19–30. Springer (2016)
26. Speck, D., Bestmann, M., Barros, P.: Towards real-time ball localization using cnns. In: Robot World Cup. pp. 337–348. Springer (2018)
27. Szemenyei, M., Estivill-Castro, V.: Real-time scene understanding using deep neural networks for robocup spl. In: Robot World Cup. pp. 96–108. Springer (2018)
28. Szemenyei, M., Estivill-Castro, V.: Robo: Robust, fully neural object detection for robot soccer. In: Robot World Cup. pp. 309–322. Springer (2019)
29. Szemenyei, M., Estivill-Castro, V.: Fully neural object detection solutions for robot soccer. Neural Computing and Applications pp. 1–14 (04 2021)
30. Teimouri, M., Delavaran, M.H., Rezaei, M.: A real-time ball detection approach using convolutional neural networks. In: Robot World Cup. pp. 323–336. Springer (2019)
31. Tsipras, D., Santurkar, S., Engstrom, L., Ilyas, A., Madry, A.: From imagenet to image classification: Contextualizing progress on benchmarks. In: International Conference on Machine Learning. pp. 9625–9635. PMLR (2020)
32. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)

**Acknowledgments.** Thanks to all individuals and teams that provided data and labels or helped to develop and host the ImageTagger.
This research was partially funded by the Ministry of Science, Research and Equalities of Hamburg as well as the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in project Crossmodal Learning, TRR-169.