



ImageTagger: An Open Source Online Platform for Collaborative Image Labeling

Niklas Fiedler^{1,2(✉)}, Marc Bestmann^{1,2}, and Norman Hendrich²

¹ Hamburg Bit-Bots, Department of Informatics, University of Hamburg,
Vogt-Kölln-Straße 30, 22527 Hamburg, Germany
{[5fiedler](mailto:5fiedler@informatik.uni-hamburg.de), [bestmann](mailto:bestmann@informatik.uni-hamburg.de)}@informatik.uni-hamburg.de

² TAMS, Department of Informatics, University of Hamburg,
Vogt-Kölln-Straße 30, 22527 Hamburg, Germany
hendrich@informatik.uni-hamburg.de
<http://robocup.informatik.uni-hamburg.de>

Abstract. The need for labeled training data for object recognition in RoboCup increased due to the spread of deep learning approaches. Creating large sets of training images from different environments and annotating the recorded objects is difficult for a single RoboCup team.

This paper presents our tool *ImageTagger* which facilitates creating and sharing such data sets. The tool is already being successfully used in RoboCup Soccer, and a large amount of labeled data is publicly available. Other leagues are invited to use this tool to create data for their contexts.

Keywords: RoboCup · Open source · Image labeling · Deep learning

1 Introduction

The approaches for object recognition in RoboCup Soccer evolved significantly during the last few years. This was triggered by rule changes in multiple leagues, replacing the simple color-coded environment with a realistic one. Combined with the increase of the field size and the change to artificial grass ball and goal recognition got more difficult, and previously popular algorithms, e. g. [5] were not able to reliably detect objects over large distances anymore. Therefore, many teams started to use different kinds of machine learning techniques [4, 11, 12]. A huge amount of labeled images is needed for training deep neural networks, which requires a lot of image recording and labeling. Furthermore, in order to achieve good training sets, multiple recording locations, e. g. different RoboCup competitions and various objects have to be used.

This makes it very difficult for a single team, especially for a new one, to achieve large, high-quality training sets. This problem also exists in other leagues, e. g. RoboCup@Home, where objects and environment change from year to year.

The workload can be lowered by either providing a tool which enables faster labeling or by sharing training data with other teams. We tried to achieve both

by implementing an online tool called *ImageTagger*. It provides intuitive user interfaces for labeling images, for verifying annotations and for managing image sets. The labels are saved internally in a common format to ensure compatibility but can be exported in user-defined formats, so that no changes in the existing training processes of the teams are needed. The integrated user and permission management system allows the team admins to decide which of their image sets are be public or private. While these features allow collaboration in any RoboCup league, this paper will focus on our results in the soccer context.

The remainder of the paper is structured as follows: first, already existing tools are compared to ImageTagger in Sect. 2. The core features of our tool are explained in Sect. 3 and an evaluation of the tool is provided in Sect. 4. The paper concludes with a summary and an outlook to future work in Sect. 5.

2 Related Work

Manual labeling of images for object recognition is a common task since it is needed for many supervised learning approaches. Therefore, many different tools

Table 1. List of commonly used image labeling tools compared by compliance to requirements for collaboration of RoboCup teams.

| | Online | Open Source | Label Types (shapes) | Label Categories | Export Format | User/Permission Management | Verification | Label Import | Image Preloading | Image Upload | Image Management |
|--------------------|--------|-------------|----------------------|------------------|----------------|----------------------------|--------------|--------------|------------------|--------------|------------------|
| sloth [6] | | ✓ | user definable | user definable | self impl. | 0 | | 0 | 0 | 0 | 0 |
| Ratsnake [8] | | (free) | polygon, grid | user definable | multiple | 0 | | 0 | 0 | 0 | 0 |
| LabelImg [9] | | ✓ | polygon | user definable | PASCAL VOC | 0 | | 0 | 0 | 0 | 0 |
| via [7] | ✓ | ✓ | multiple | free text | JSON, CSV | | | | 2 | ✓ | |
| Rhoban Tagger [3] | ✓ | ✓ | binary | admin defined | JSON | 1 | ✓ | | | | |
| LabelMe [10] | ✓ | ✓ | polygon | free text | XML | | | | | ✓ | ✓ |
| Labelbox [2] | ✓ | | admin defined | admin defined | JSON, CSV | ✓ | | | ✓ | ✓ | ✓ |
| ImageTagger | ✓ | ✓ | multiple | admin defined | user definable | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

⁰ Not applicable in an offline tool

¹ Users start in training mode

² Images are stored in the browser cache

already exist. Though, none of the programs presented in Table 1 offers the combination of labeling with other features (e. g. online image access, user, and annotation management) needed to allow collaboration between teams.

Proprietary products are difficult to use in a community like RoboCup because they are not customizable enough to fit the specific requirements of the environment. Furthermore, teams cannot contribute to the development and implement desired features for the whole community.

Offline tools are afflicted by multiple problems such as the installation process and compatibility issues. The main reason to exclude them from our options is the image and label management. Multiple team members need to coordinate their work and progress, and the files they are working with.

Most of the competing online tools require a lot of server-side management of images and annotations, which makes it difficult for multiple teams and users to work together efficiently. While a cloud could handle the image exchange between multiple users, it lacks useful metadata about these image collections (e. g. location or description of the situation) and most notably the labels in a universal format. For online tools, the image preloading feature is essential to overcome the latency to the server which gets significantly high for large images and high distances between the users and the server. The ability to export the labels in a format defined by the teams themselves is required for sharing the image and label data. In this domain, the ImageTagger offers customizability comparable to self-implementation of the export for every user. In the RoboCup environment, the amount of label categories is rather limited. It proved to be a faster way to label only one category at a time, which is easily applicable in the RoboCup environment.

3 ImageTagger Overview and Features

ImageTagger provides an efficient browser-based user interface for all required tasks: image labeling, verifying annotations, up- and downloading images/labels, managing users and teams, and the definition of image and label categories. The

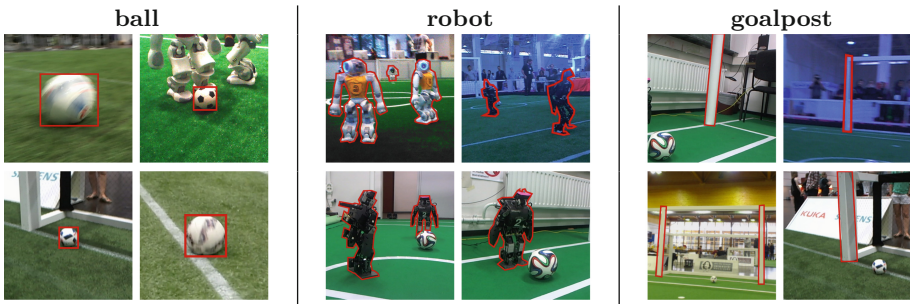


Fig. 1. Exemplary labels from categories common in RoboCup soccer. Precise labels are created to allow learning of exact object localization.

software is written in Python, using the web framework Django. Its key features are explained in the following subsections.

3.1 Manual Labeling

The *annotation view* allows the user to create labels (Fig. 1) on images. The ImageTagger offers tools to create bounding box, polygon, line and point annotations (Fig. 2). Since this is a highly repetitive task, it has to be done as fast as possible. Therefore, the images are sorted in a list which can be filtered by existing label types (e. g. ball). The user then iterates through the images using shortcuts, leaving the mouse free for creating annotations. Successive images get preloaded while the user creates an annotation for the current image, allowing a fast transition to the following image. An option to keep the last annotation enables faster labeling since the image sets are often created sequentially and require only a small adaption in the position of the label between two images. Labels can be marked as “blurred” and “concealed”. Existing annotations are listed below the image and can be drawn into the image if needed. The option to label a category as “not in the image” allows users to create negative data.

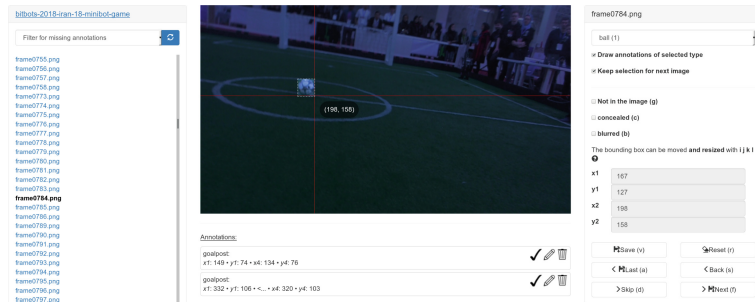


Fig. 2. The *annotation view* while creating a ball annotation. On the **left** is a list of images which can be filtered for missing annotations. In the **center**, the image is displayed and an annotation can be created. On the **right**, controls are provided, most of which can also be accessed via keyboard shortcuts for speed.

3.2 Automated and Offline Labeling

ImageTagger enables users to upload existing labels to its database. This upload feature allows users to share labels between multiple instances of ImageTagger, to restore local backups of labels and to migrate existing training data which was created with other tools.

Some deep learning methods, e. g. deep FCNNs, are not applicable during RoboCup games due to their runtime, but they can be used to create labels automatically. The results do not need to be optimal since users can verify the labels after uploading them (cf. Sect. 3.3).

3.3 Label Verification

To ensure sufficient accuracy and quality of the image labels, ImageTagger includes a special mode for label verification. The *verification view* allows permitted users to inspect a label and give it a positive or negative verification. As the annotation view, it preloads the images and annotations to reduce the perceived latency. Additionally, it is optimized for mobile usage. A positive verification increases the verification level of a label, while a negative one decreases it. The verification is a binary decision; thus it is much faster than the labeling process itself.

A manually created annotation is automatically positively verified by the user creating it. This results in label data where one positive verification means that at least one human considered the annotation as acceptable. Therefore, a verification count of two or more can be considered as sufficient for most use cases.

3.4 Image Management

To keep the high number of images required for most deep learning approaches manageable, images are grouped into *image sets*. Each set has a context, e. g. the same ball and field type, and belongs to a team. The *image set view* consists of an image list like the annotation view, general information, a management section, an export section and links to manage the annotations of the set. Members of the team that owns the image set can update the name, location, and description of the set and upload new images or labels. The *image-lock* option can be selected to disable further image upload to keep sets in a static state, e. g. to provide an immutable benchmarking set.

3.5 Collaboration

The labeling process is easy for humans, but it is a tedious and time-consuming task. Thus, collaboration is necessary to reduce the workload on each team. However, different algorithms [4, 11, 12] usually expect their own specific categories and data representations, leading to a high variance in requirements that the tool has to accommodate. To make labels usable for multiple teams, use cases and processing approaches, ImageTagger allows the creation of custom export formats. For every export format, the creator chooses which label categories to include and whether blurred or concealed labels are included. The user employs placeholders (cf. Table 2) which get replaced with the corresponding values in the export creation. All values representing a measure or coordinate in the image are available in an absolute or relative (to the size of the image) form. Resulting from the variable amount of points in a label, the list of x- and y-values has to be generated following a user-defined pattern. The “concealed” and “blurred” flags can be exported by defining text which is only included in the export of the label when the corresponding flag is (not) set.

Table 2. The available placeholders according to their contexts. Each value is provided in an absolute pixel value or as a value relative to the image size. The file name format specifies the format of the name of the export file, which the user can download. The image placeholders are only used when the user selects the option to aggregate the labels by images. The vector placeholders are used to generate the list of x/y values for label types with a variable number of points.

| file name placeholders | label placeholders |
|---|--|
| name, team, location and unique id of the image set | name of the image/image set |
| | width/height of the image |
| | label category |
| | the amount of verifications for the label |
| | width/height, center/upper left/lower right point, mean height/diameter of the label |
| | representation of the vector |
| | alternative text for "not in image" labels |
| image set placeholders | vector placeholders (optional) |
| name, team, location, description of the image set | the number of the current point |
| the content of the image/label format | x/y-coordinate of the current point |
| image placeholders (optional) | |
| name of the image/image set | |
| width/height of the image | |
| number of labels for the image | |
| the content of the label format | |

In addition to a simple list of annotations, the option to concatenate the annotations (cf. Fig. 3) by images is given. Depending on the chosen option, the user has to define an image set format, an image format (only when the label concatenation is used) and a label format. Created export formats can be saved privately or publicly and can be used by other users.

While most of the data management is handled by ImageTagger, some tasks need to be processed locally on a system, e. g. the creation of rosbags or writing the label data into the metadata of the images.

Tools, designed for those tasks can be shared using ImageTagger since some of them can be useful for multiple teams in a shared instance.

A permission management system for users and teams is needed to be able to create and share training data in a controlled way. Read and write permissions of image sets can be set by the owning team.

The variety of collaboration options allows teams to define whether they work together with the whole community in every way, to just share the training data without the possibility to collaborate or to keep the set completely private. The distinction between users and admins in a team helps to coordinate the members. To motivate the users to keep labeling and to detect wrongly labeling users, a scoring system was introduced. The score of a user is the sum of positive minus the sum of the negative verifications that were made on the annotations created by the user. This way good annotations are rewarded and wrong annotations

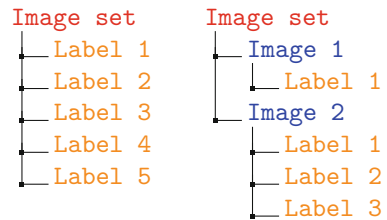


Fig. 3. export format composition hierarchy with (right) and without (left) label aggregation by image

are penalized. In the user explore view and the team view, the users are ranked, based on their score. The team view offers a 30-day user high score to focus on a shorter timespan, to make it possible for new team members to compete with the rest and engage users to keep up their effort over a longer time period.

4 Evaluation

We provide a public ImageTagger instance for the RoboCup Soccer environment. The server is open for everyone to log in, to download existing images and annotations, or to upload images, label them and verify the labels. See Table 3 for our current server statistics. At the moment, there are 119 public image sets recorded in the RoboCup environment available for download. They belong to participating teams, most notably Hamburg Bit-Bots, Nao Devils and WF Wolves. Based on these sets, team Bit-Bots recently proposed a ball localization challenge, which gives teams a benchmark to compare their approaches [1]. The training data was used to train neural networks proposed by Daniel Speck [13].

Table 3. Current numbers on our instance of the ImageTagger of images and labels (**left**) as well as users and teams (**right**).

| | image sets | images | ball labels | robot labels | goalpost labels | users (50+ labels) | teams | active teams (2+ active users) |
|--------|------------|---------|-------------|--------------|-----------------|--------------------|-------|--------------------------------|
| all | 171 | 171,057 | 95,352 | 41,016 | 18,800 | 50 | 61 | 6 |
| public | 119 | 155,602 | 92,588 | 37,981 | 17,004 | | | |

5 Conclusion and Further Work

In this paper, we presented a tool which facilitates the production and sharing of labeled image data for supervised learning in object recognition. It is already actively used by multiple teams in the Humanoid and Standard Platform League, and image sets from future RoboCup competitions will be uploaded.

Currently, the tool is only used in the soccer context, but it would be possible to use it for other areas as well, e. g. labeling household objects for the @Home League. The modular design allows the adaption of the labeling interface also for the labeling of RGB-D data while keeping the rest of the framework. In the future, the usability of the tool should be improved for usage on mobile systems to enable precise labeling on tablets and smartphones.

We encourage other RoboCup Soccer teams to use the public ImageTagger instance hosted on our server, to download training sets, and to upload further images and labels: <https://imageragger.bit-bots.de>

The project source code is available at:

<https://github.com/bit-bots/imageragger>

Acknowledgments. Thanks to the RoboCup teams Hamburg Bit-Bots, Nao Devils and WF Wolves for taking part in the development. Thanks to the Server AG. Thanks to the contributors, particularly to Timon Engelke, Rebecca Glaser, Jonas Hage, Jennifer Meyer, Daniel Speck, and Pascal Wichmann.

This research was partially funded by the German Research Foundation (DFG) and the National Science Foundation of China (NSFC) in project Crossmodal Learning, TRR-169.

References

1. Bit-bots ball localization challenge. <https://robocup.informatik.uni-hamburg.de/en/documents/bit-bots-ball-localization-challenge-2018/>. Accessed 31 Mar 2018
2. Labelbox. <https://www.labelbox.io/>. Accessed 25 May 2018
3. Rhoban tagger. <http://rhoban.com/tagger/index.php>. Accessed 30 Mar 2018
4. Albani, D., Youssef, A., Suriani, V., Nardi, D., Bloisi, D.D.: A deep learning approach for object recognition with NAO soccer robots. In: Behnke, S., Sheh, R., Sarel, S., Lee, D.D. (eds.) RoboCup 2016. LNCS (LNAI), vol. 9776, pp. 392–403. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68792-6_33
5. Budden, D., Fenn, S., Walker, J., Mendes, A.: A novel approach to ball detection for humanoid robot soccer. In: Thielscher, M., Zhang, D. (eds.) AI 2012. LNCS (LNAI), vol. 7691, pp. 827–838. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35101-3_70
6. Bäuml, M.: Sloth documentation (2014). <http://sloth.readthedocs.io/en/latest/>
7. Dutta, A., Gupta, A., Zissermann, A.: VGG image annotator (VIA) (2016). <http://www.robots.ox.ac.uk/~vgg/software/via/>. Accessed 10 Mar 2018
8. Iakovidis, D., Goudas, T., Smailis, C., Maglogiannis, I.: Ratsnake: a versatile image annotation tool with application to computer-aided diagnosis. *Sci. World J.* (2014). <https://doi.org/10.1155/2014/286856>
9. Lin, T.: Labeling. <https://github.com/tzutalin/labelImg>. Accessed 29 Mar 2018
10. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. *Int. J. Comput. Vis.* **77**(1–3), 157–173 (2008). <https://doi.org/10.1007/s11263-007-0090-8>
11. Schneckeburger, 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 (2017)
12. Speck, D., Barros, P., Weber, C., Wermter, S.: Ball localization for robocup soccer using convolutional neural networks. In: Behnke, S., Sheh, R., Sarel, S., Lee, D.D. (eds.) RoboCup 2016. LNCS (LNAI), vol. 9776, pp. 19–30. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68792-6_2
13. Speck, D., Bestmann, M., Barros, P.: Towards real-time ball localization using CNNs. In: Holz, D. et al. (eds.) RoboCup 2018. LNAI, vol. 11374, pp. xx–yy. Springer, Heidelberg (2018)