XXVIII Congresso {virtual} de Iniciação Científica da Unicamp 2020

Deep Neural Networks for multiclass detection and depth estimation in the humanoid robot soccer

Gabriel Previato[1] · **Esther Luna Colombini**[1]

Keywords Deep Learning, Object Detection, Robotics

1 Introduction

In humanoid robot soccer [1], the performance of the highest-level decision-making and localization algorithms depends on a understanding and interaction with the environment.

More recently, new approaches that make use of Machine Learning [2] [3] [4] techniques for the extraction of characteristics of the obtained images have been considered.

Such supervisioned methods need a large amount of data to be trained. Obtaining real world dataset images is quite expensive. It involves using a set of cameras and sensors that are very expensive and sensible, requiring a big effort in calibration and later data cleaning.

Even though there is a great number of images datasets for visual machine learning algorithms [5–8], in the RoboCup SPL and RoboCup Humanoid League, there is not a extensive dataset available containing a large number of images with bounding boxes, object labels and depth map, and given that the rules of these leagues are constantly changing - these changes includes field size, goalposts, number of robots and their sizes, the ball size and color - a real image dataset can easily became obsolete. Thus the benefit of using a simulated image dataset. Since simulations are easier to modify and expand, creating simulated datasets becomes a easy and feasible task.

Our work aims at creating a realistic simulated dataset in the Robocup Humanoid Soccer Leagues. We propose a pipeline that can be used to easily expand and collect images for depth estimation, object detection, classification and tracking. We further compare some state-of-the-art object detection algorithms in our proposed dataset and then evaluates these algorithms in the real world scenario.

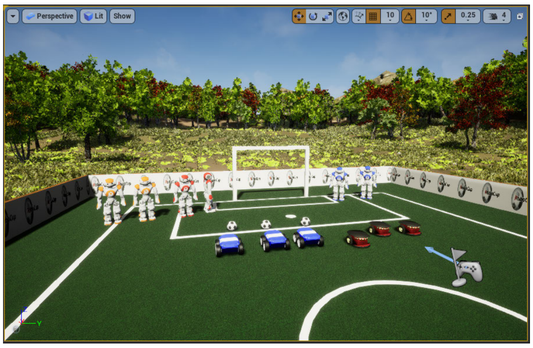


Fig. 1 Unreal Engine 4 soccer scenario.

2 The Dataset

We propose a model for a simulation-generated dataset that can be easily modified and increased, with a great amount of data with absolutely precise ground truth regarding the depth field and object class and position in the image. Our base dataset [link to dataset, inserted on publication] contains over 1 million images and their corresponding object annotations.

Using the Unreal Engine 4 [9] and the AirSim [10], we built a soccer field following the 2020 RoboCup Humanoid League field specifications for the Adult size competition.

In the Figure 1 we can see images of the built Unreal Engine 4 scenario.

Currently, our dataset consists of:

- RGB images
- Corresponding depth field images
- Text file containing all objects in the images with its class, center position and bounding box sizes
- Segmentation files for each individual object in the image

[1] Laboratory of Robotics and Cognitive Systems (LaRoCS)
Institute of Computing (IC)
University of Campinas (Unicamp), SP, Brazil
E-mail: contact@larocs.ic.unicamp.br
This project was supported by CNPq

In our dataset, all images have a resolution of 1024x640 pixels, although when doing the dataset acquisition, this resolution parameter can be changed in order to save disk space.

The depth image is a Portable Float Map image, in which each pixel has a float value representing the centimeters distance from the camera to the element at that pixel.

The segmentation image is a binary image, in which the segmented object has the value 255 while everything else has the value 0.

The objects position information is given in a JSON file, for every RGB image, there is an object JSON file containing all objects in that image. For each object, there is described its class, its center position in the image and bounding boxes dimensions. These are relative to the image size, so any re-scale won't affect the object position and bounding box information.

2.1 Dataset Acquisition

We built a script to automatically control the scene's environment and objects and also acquire data and later process it to get the already mentioned images. [link to github, inserted on publication]

We created a grid discretization of the field of 0.5m in both x and y axis and then we iterate for every grid cell, setting the camera in 8 different yaw angles, and we generate a random value for the roll an pitch angles, to simulate the head movement of a robot. For the other robots and the ball, we set then in random positions, following a normal distribution to cover the most number of possibles positions for the dataset.

With the RGB, Depth field and Segmentation images obtained with the Airsim, we than calculates all objects bounding boxes and distances to the camera, and save this information in a JSON file format.

This file saves the RGB image width and height, and all objects that are visible in the image with their class, bounding box center position, bounding box width and height. All bounding boxes positions are relative to the image size, so if the image is resized, the information of the bounding boxes will still be valid.

3 Dataset Tests

In order to validate our dataset we trained our previous proposed network for the SSNDa but we also made modifications to this network in order to improve it's object detection and depth estimation results in our new dataset. Since the main goal we want to achieve with this dataset is training detection models with simulated images that can be easily fine tuned later with real images, we propose the following test pipeline for this dataset and the previous one.

1. Train different NN models with the new and old dataset.
2. Evaluate these models performance in real world images.
3. Compare the results.

3.1 Network

For the tests in our new dataset, we used the original MODL network that we proposed in our latter work, and we also used some modifications of that network.

The MODL architecture 2 is a DCNN composed by fine tuned version of the VGG19 as a feature extractor that feed 2 different branches, one for the object detection task, and other for the depth estimation task.

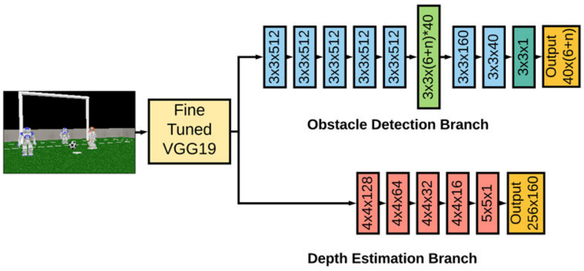


Fig. 2 Original MODL architecture.

We then made 3 more modifications on this network. We substituted the Obstacle Detection Branch with the YOLOv3 [11] network keeping the original Depth Estimation Branch, we substituted the Depth Estimation Branch with an attentional depth estimator network, proposed by Xu [12], and finally we updated both branches with the already mentioned networks.

We also tested our dataset with the J-MOD network, proposed by Mancini [4].

3.2 Results

In the Table 1 we can see the results for 5 different network compositions.

For the obstacle detection task, the MODL using a similar architecture as the YOLOv3 shows an increase in all metrics. This result was expected, since the original MODL detection branch is based on the first YOLO architecture. For the depth estimation task, the Structured Attention has also better results in comparison with the original MODL depth estimation branch. The error for the depth estimation for an object decreased by 1/3, showing that the attentional factor has a great impact on the objects itself than on the background. Since all our objects have a different shape and

	J-MOD	MODL	MODL Attention	MODL YOLOv3	MODL Attention and YOLOv3	
Detection IOU	64%	69%	69%	76%	76%	Higher
Detection Precision	71%	72%	72%	81%	80%	is
Detection Recall	84%	82%	81%	89%	89%	better
RMSE Full Depth Map [m]	1.99	1.91	1.20	1.92	1.20	Lower is
Depth RMSE on Obs.(Mean) [m]	0.29	0.37	0.13	0.37	0.13	better

Table 1 Results on the complete dataset.

color than the field and the trees in the background, this impact was expected. In the Figure 3 we can see a graph that shows the average precision and depth error values by class and distance. For the precision values we can see that the YOLOv3 architecture improves the objects precision when they are closer to the camera, while for the objects that are further, the precision is similar to the ones achieved in the original MODL architecture. For the depth error values we can see that the Attentional Depth architecture is a lot better than the original MODL approach, even for objects that are far from the camera.

In the Figure 5 we can see a RGB frames and their depth ground truth, and the respective detected obstacles in the image and the estimated depth by the network.

We also made the same tests, but instead of using the whole dataset, we only used a subset of it that contains only the NAO robots, so we could see the effect that adding more robots in the scene can cause on the final result. We can see in the Table 2 we can see the results for 5 different network compositions in the NAO subset.

In the Figure 4 we can see a graph that shows the average precision and depth error values by class and distance.

Similar to the full dataset, both the YOLOv3 architecture and the Attentional Depth architecture improves the results.

4 Real Image Tests

One of the purposes of our photo-realistic simulated dataset is to reduce the sim2real gap. In order to validate if our dataset enhance the modified MODL network in the real world scenario, we took the later model and trained it in the SSNDa and in our new proposed dataset, and run the model in a footage of the RoboCup Standard Platform League (SPL). Since these footages don't have the ground truth, we can only attest the qualitative results of the new proposed dataset. In the Figure 6 we can see some examples of obstacle detection task results of the model running on the SPL footage.

We can easily see that even though no fine tuning or post training was done, the network that was trained in our new proposed simulated dataset outperforms the one that was trained in the SSNDa for the robot and goal post detection.

5 Conclusions and future work

We conclude that, for the tested neural networks, as expected, their performance in the new proposed dataset is lower than on the SSNDa, given that our new dataset have images with more complex features and more details in the objects textures. But the networks results on the real world images are a lot better when they are trained in our new dataset, showing us that using a more photo realistic simulated dataset, we can reduce the sim2real gap.

As future works, we highlight:

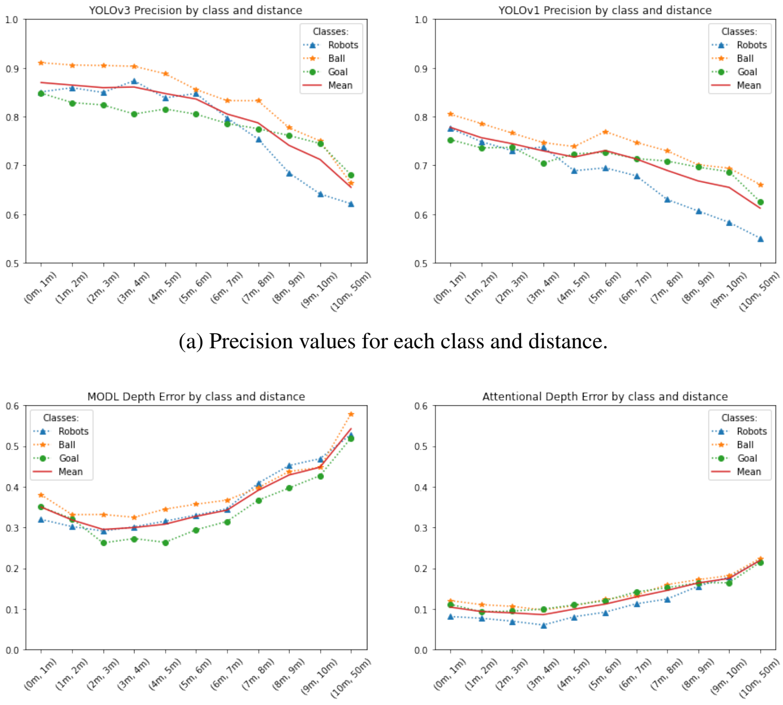
- Expand the dataset to include more variations of the soccer field (addition of indoor/outdoor arenas)
- Add more background noise, such as moving people, to the dataset
- Make a sequence of frames to obtain tracking information

Acknowledgements

The authors thank the CNPq for the funding support and the Laboratory of Robotics and Cognitive Systems at UNICAMP (LaRoCS) and the Institute of Computing at UNICAMP for the infrastructure provided to help this project.

References

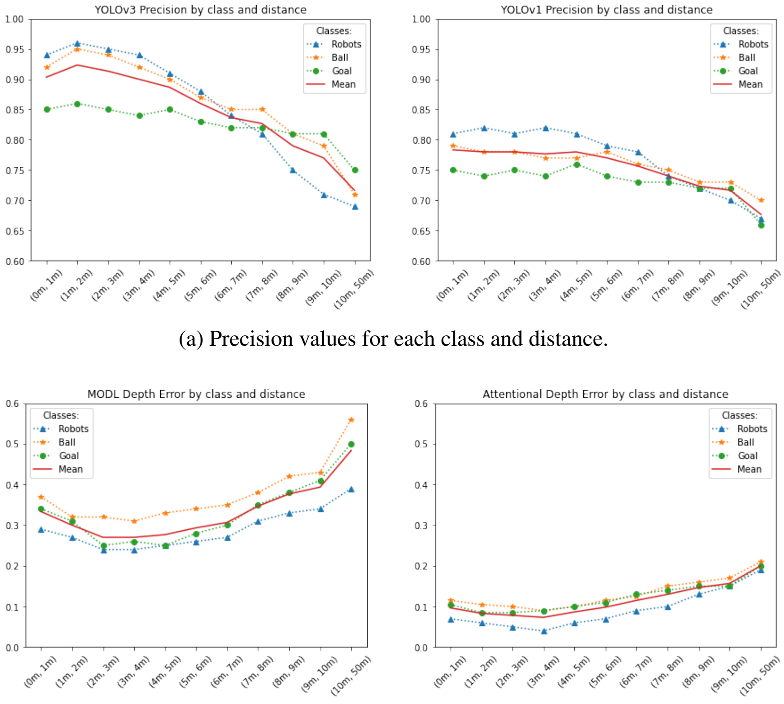
1. (2019) Robocup humanoid league. [Online]. Available: https://www.robocuphumanoid.org/
2. D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015.
3. M. Mancini, G. Costante, P. Valigi, T. A. Ciarfuglia, J. Delmerico, and D. Scaramuzza, "Towards domain independence for learning-based monocular depth estimation," in *IEEE Robotics and Automation Letters*, 2017.
4. M. Mancini, G. Costante, P. Valigi, and T. A. Ciarfuglia, "J-mod2: Joint monocular obstacle detection and depth estimation," in *IEEE Robotics and Automation Letters*, 2018.
5. T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312



(a) Precision values for each class and distance.

(b) Depth error values for each class and distance.

Fig. 3 Obstacle detection and Depth estimation metrics for each class and distance for tests on the full dataset.



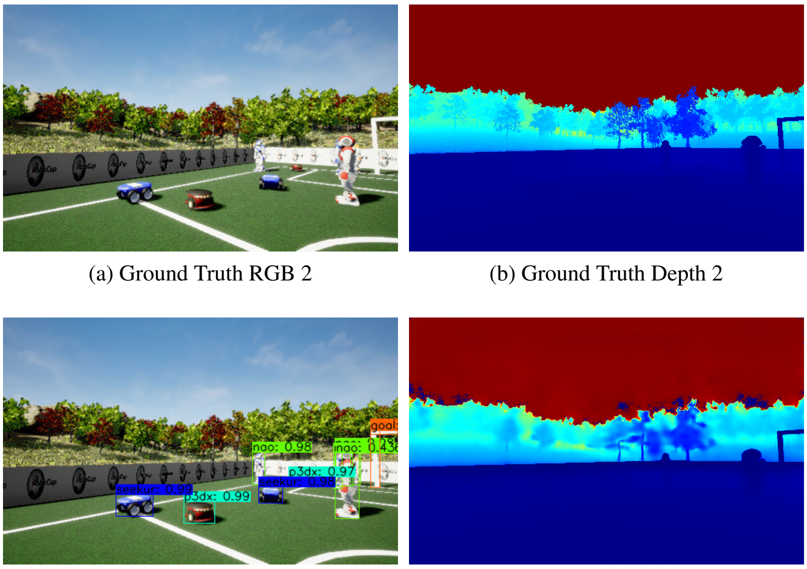
(a) Precision values for each class and distance.

(b) Depth error values for each class and distance.

Fig. 4 Obstacle detection and Depth estimation metrics for each class and distance for tests on the NAO subset.

6. A. Kuznetsova, H. Rom, N. Alldrin, J. R. R. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, “The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale,” *CoRR*, vol. abs/1811.00982, 2018. [Online]. Available: http://arxiv.org/abs/1811.00982

7. M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *Conference on Computer Vision and Pattern Recogni-*



(a) Ground Truth RGB 2

(b) Ground Truth Depth 2

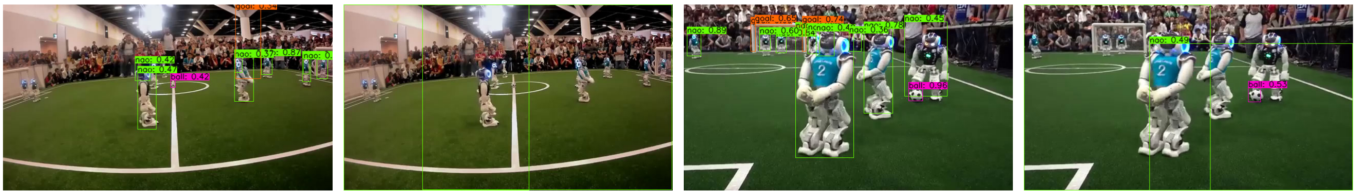
(c) Detected obstacles 2

(d) Estimated Depth 2

Fig. 5 Qualitative results of the best model prediction.

	J-MOD	MODL	MODL Attention	MODL YOLOv3	MODL Attention and YOLOv3	
Detection IOU Detection Precision Detection Recall	69% 75% 88%	70% 75% 85%	70% 75% 85%	79% 84% 91%	79% 85% 91%	Higher is better
RMSE Full Depth Map [m] Depth RMSE on Obs.(Mean) [m]	1.97 0.27	1.93 0.33	1.15 0.11	1.93 0.33	1.15 0.11	Lower is better

Table 2 Results on the NAO subset of the dataset.



(a) Frame 1 result trained on new dataset

(b) Frame 1 result trained on SSNDa

(c) Frame 3 result trained on new dataset

(d) Frame 3 result trained on SSNDa

Fig. 6 Real image examples.

tion (CVPR), 2015.

8. J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.
9. Epic Games, “Unreal engine.” [Online]. Available: https://www.unrealengine.com
10. S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065
11. J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767
12. D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, “Structured attention guided convolutional neural fields for monocular depth estimation,” in *CVPR*, 2018.