Soccer Robot Perception

CudaVision - Learning Vision Systems on Graphics Cards

Ragith Ayyappan Kutty, Lokesh Veeramacheneni

Hochschule Bonn-Rhein-Sieg
firstname.lastname@smail.inf.h-brs.de

Abstract. This paper presents the implementation of the NimbRoNet2 visual perception model used for the RoboCup 2019 by winner team NimbRo. In the RoboCup competition particularly for the soccer league, there are challenges related to the perception of ball, robots, goalposts, lines and field. The NimbRoNet2 model can perform object detection and semantic segmentation simultaneously in a single pass. In this paper we emulate the results and also provided a detailed discussion of the implementation and evaluation details.

1 Introduction

The RoboCup initiative aims to promote robotics and AI research in the form of a challenge. The ultimate aim of the RoboCup is to build a team of humanoid robots by 2050 that will be able to win a soccer game against the most recent winner of the World Cup. Building a robot that can play soccer has a little social and economic impact but it will certainly be a major accomplishment for the field as it involves many interesting problems in perception, localization and navigation.

Particularly for the soccer league research issues looks into the perception of ball and players on the field, self-localization, dynamic walking, running, and kicking a ball while maintaining balance. For RoboCup 2019, the field dimensions were increased to 14×9 m. This posed challenges to the perception system in terms of detecting further away balls and goalposts, localization in terms of robust field line detection. The NimbRoNet2 model was able to overcome these challenges that lead team NimbRo to win the soccer tournament. The model has an encoder-decoder architecture with a pre-trained ResNet-18 backbone. It has two output heads one each for object detection and segmentation.

The following section will discuss related work in Section 2 , network architecture in Section 3, implementation details in Section 4, evaluation metrics in Section 5. This is followed by result and conclusion in Section 6 and Section 7.

2 Related Work

NimbroNet2 utilizes an architecture similar to U-Net [8] which has an encoder-decoder arrangement and especially used for segmentation tasks. The two main differences between NimbroNet2 and U-Net are the intermediate connections from the encoder to decoder parts and also the multiple heads for detection and segmentation. The intermediate connections from the encoder to the decoder is inspired from SweatyNet architecture [9]. Another major difference between the NimbroNet2 architecture and Sweatynet is that the latter one uses a Bilinear upsampling in the decoder part whereas the first one implements a ConvTranspose layer.

Joint detection and segmentation type architectures are relatively new and used in multiple applications. Araújo et al. [1] proposes a network called UOLO-Net which is a combination of U-Net and Yolov2 detector. It enforces multi-task learning of detection and segmentation of blood clots from a medical image. Recently different kind of networks has been raised for simultaneous detection and segmentation. One such network is BlitzNet proposed by Yu et al. [10] which has multiple heads for detection and segmentation tasks. This was followed by improved architectures like PairNet and TripleNet as in [4]. BlitzNet, PairNet and TripleNet utilizes a similar encoder-decoder architecture as used in NimbroNet2 but has different head structures. Whereas Uolo-Net can be classified as a hybrid network.

There are other varieties of networks which can perform the task of simultaneous detection and segmentation but employs a Region Proposal Network (RPN) for segmentation task. These kind of networks are proposed in [3], [6], [11].

All the above network can perform simultaneous detection and segmentation but suffers from a problem. These networks are designed and trained to perform detection and segmentation on the same object. The implemented NimbroNet2 can perform object detection for robots, ball and goal posts and segmentation for line and field detection.

3 Network Architecture

The unified deep convolutional neural network can perform object detection and semantic segmentation in one forward pass. The visual perception architecture NimbRoNet2 is illustrated in Figure 1.

The architecture consists of two output heads: object detection and semantic segmentation. The detection head is responsible for detecting ball, robots, and goalposts. Line and field detection is the responsibility of the segmentation head. The model has an encoder-decoder architecture similar to SegNet [2] and U-Net [10] . A shorter decoder than the encoder helps overcome the computational limitations and supports real-time perception. For the encoder a modified pre-trained ResNet-18. To adapt the classification network ResNet-18 for simultaneous detection and segmentation, the Global Average Pooling and fully connected layers are removed. In the decoder part, transpose-convolution is used for representation upsampling. Additionally to use location-dependent features, a shared location-dependent bias is used for the output heads.

4 Implementation

The following sections discuss the various facets of implemented methods: dataset and dataloader, loss functions and training.

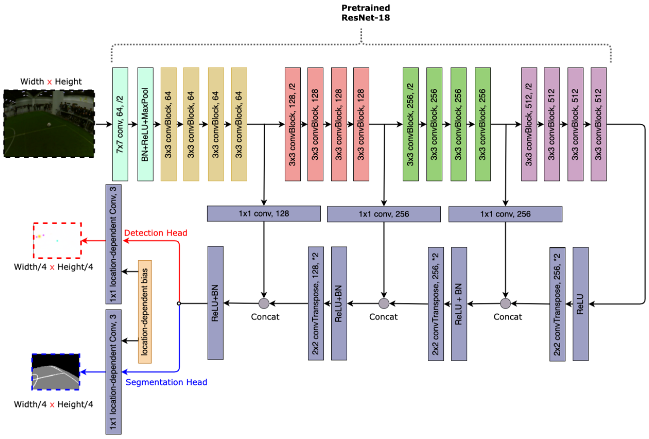


Fig. 1: NimbRoNet2 architecture [7] with pretrained ResNet-18 backbone and location-dependent bias. For simplicity, the residual connections in ResNet are not shown.

4.1 Dataset and dataloader

The architecture in Figure 1 is trained and tested on a dataset consisting of 8858 detection images and 1192 segmentation images. Detection images were trained for three classes (robot, goal post and ball) whereas the segmentation images were trained for three classes (lines, field and background).

Two different dataloaders are implemented, one each for segmentation and detection task. The data loader for the detection task includes the reading of the corresponding image XML annotation file. Then for each object (goalpost, robot, and ball), a Gaussian blob is generated with respective variance of 5, 10, 5. As discussed in [7], since the robot's canonical centre is difficult to annotate hence a bigger variance for the robot will penalize the network less for not imitating human labels. The dataloader for the segmentation task uses the ground truth image to generate a masked image denoting the various class instances. Following the original implementation, the background is changed to black, field to grey and lines to white pixel values.

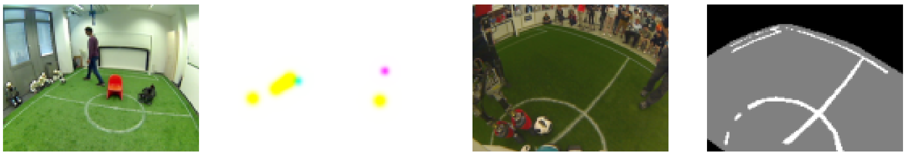


Fig. 2: Sample from the dataset. First two images represent the original image for detection and corresponding target image with robots in yellow, goalpost in magenta and ball in cyan. The later two images depict the input image for segmentation and corresponding target image with lines in white, field in gray and background in black.

4.2 Loss Function

Multiple loss functions were used for detection and segmentation tasks. The detection task involves the weighted combination of total variation loss and Mean Square Error (MSE) loss. Whereas, segmentation uses a weighted combination of total variation loss and cross-entropy loss. A weighted combination of losses is applied because of higher magnitudes of total variation loss which hinders the network convergence. The respective weights for total variation loss in detection, segmentation tasks are 2e-6, and 3e-7 respectively. To account for the intra-class imbalance in segmentation task the cross-entropy loss is weighted per class, and the weights used are 0.6, 0.5, and 0.95 for background, field and lines respectively.

Total variation loss. This loss calculates the gross absolute difference of neighbouring pixel values. Total variation loss ensures the spatial continuity between the images and also generates smooth images to avoid noisy and over-pixelated images. Other noise removal techniques like linear smoothing and median filtering although reduces noise but also smooths away higher frequency information contained in edges. When the total variation loss (1) measures the noise and hence helps in suppressing the noise when used in the optimization.

$$TVLoss = \Sigma_c \Sigma_j \Sigma_i |X_{i+1,j,c} - X_{i,j,c}| + \Sigma_c \Sigma_i \Sigma_j |X_{i,j+1,c} - X_{i,j,c}| \quad (1)$$

4.3 Training

As described in [7], we used an Adam optimizer with a learning rate of 1e-6 for ResNet-18 encoder and 1e-3 for other parameters. This network with a parameter total of 12,655,424 is trained for 300 epochs on Nvidia V100 GPU about 12 hours with a batch size of 16. After extensive training over various learning rates, training was found out to be stable with a learning rate of 1e-3. At first, we trained the detection and segmentation tasks with equal priority. This resulted in better performance of segmentation and yields in non-convergence of detection task. We found that this kind of performance is an outcome of an imbalanced dataset between detection and segmentation task. We adopted a training procedure from W-GAN training [5], where the discriminator is trained more times than the generator. Similarly, in our case, the detection head is trained for more batches compared to segmentation. Specifically, segmentation images are trained once for every seven batches of detection images. This kind of training procedure allowed better performance on detection and segmentation.

After training, post-processing is applied to detection outputs, wherein we apply a averaging kernel of size (5, 5). This removes small blobs of single-pixel size and thereby reduces false positives. Additionally, it also combines these single-pixels when they are nearby a blob.

5 Evaluation Metrics

The evaluation metrics calculated for detection tasks are accuracy, precision, recall, F1 and false discovery rate (FDR). These metrics are calculated based on the values of true positive, false positive and false negative. When the predicted blob's centroid is within 10 pixels of the ground truth blob's centroid, then it is accounted as a true positive else false positive. A missed detection is accounted as a false negative. For segmentation, accuracy and IOU are calculated on a pixel-level. The respective evaluation metrics for detection and segmentation are described in Table 1.

Detection metrics	**Segmentation metrics**

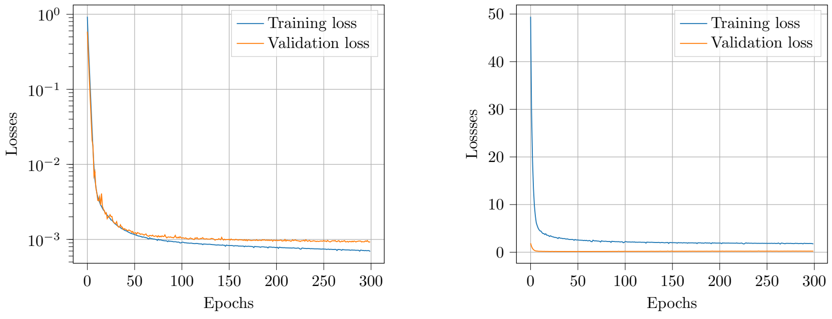
Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
Precision = $\frac{TP}{TP+FP}$
Recall = $\frac{TP}{TP+FN}$
F1 score = $\frac{2*Precision*Recall}{Precision+Recall}$
FDR = $\frac{FP}{TP+FP}$
	Accuracy = $\frac{target \cap prediction}{target}$
	IOU = $\frac{target \cap prediction}{target \cup prediction}$

Table 1: Evaluation metrics. First column represents the metrics for detection task and last column represents the metrics for segmentation task. FDR represents the False Discovery Rate

6 Results

This section portrays the training and evaluation details for detection and segmentation tasks. Figure 3 details the progression of training and validation loss over 300 epochs for detection and segmentation task.

The evaluation results for detection task is presented in Figure 4. After extensive debugging, tuning of learning rate and batch size, we achieved the best result possible. In the case of segmentation, the evaluation results are depicted in Figure 5 and we emulated the results from [7]. Figure 6 illustrates the output of segmentation and detection tasks.



(a) Detection loss

(b) Segmentation loss

Fig. 3: Training and validation loss over 300 epochs for detection and segmentation.

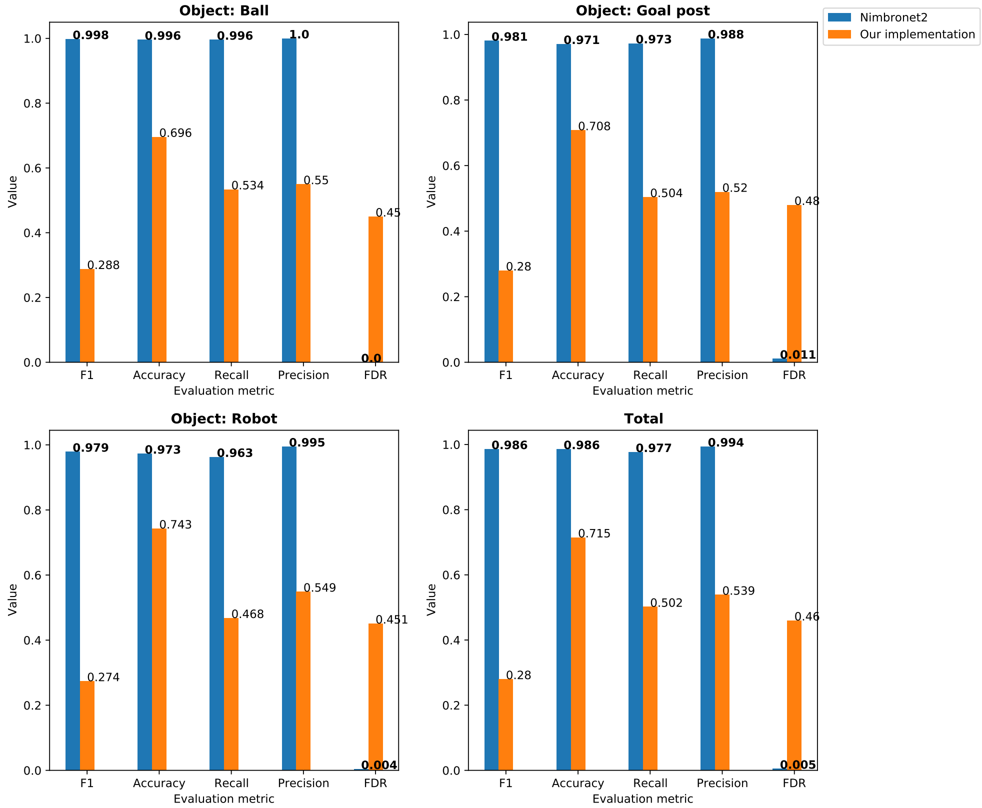


Fig. 4: Comparative evaluation metric results for detection task between our implementation and [7]. Subplot represents values for the each object and each bar in a subplot represents the corresponding detection evaluation metrics.

7 Conclusion

This paper emulates the results of the [7] for detection and segmentation tasks. A different kind of training procedure has been implemented because of the dataset size imbalance between the detection images and segmentation images. This is done by training the segmentation head once for every seven batches of training for the detection head.

Future Work. As an upgrade to the dataloader, implementing caching would save time for generating the annotations from the XML parser. The current model lacks network training with progressive resizing, it would be interesting to study the network performance with progressive resizing.

Lessons Learned. We learnt the following lessons during the course of the project.

- How to systematically train and debug an artificial neural network
- How to deal with multiple datasets of various lengths

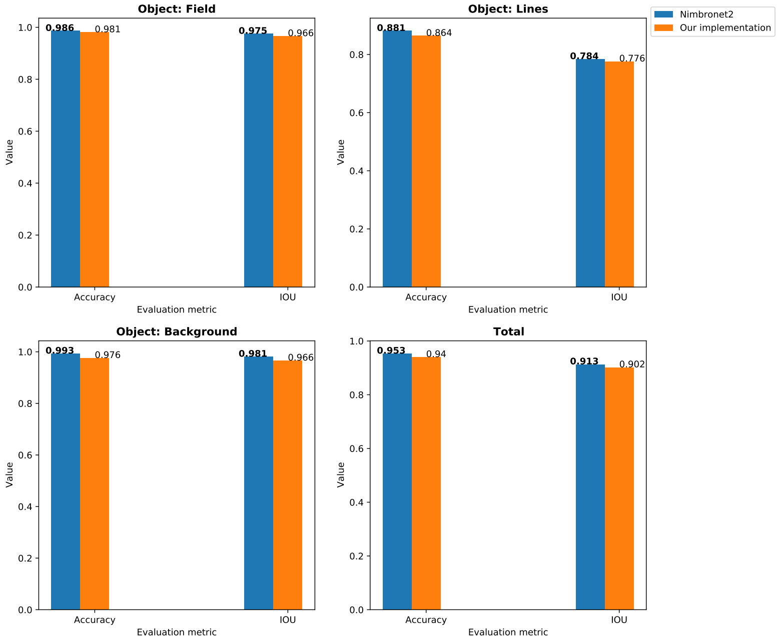


Fig. 5: Comparative evaluation metric results for segmentation task between our implementation and [7].

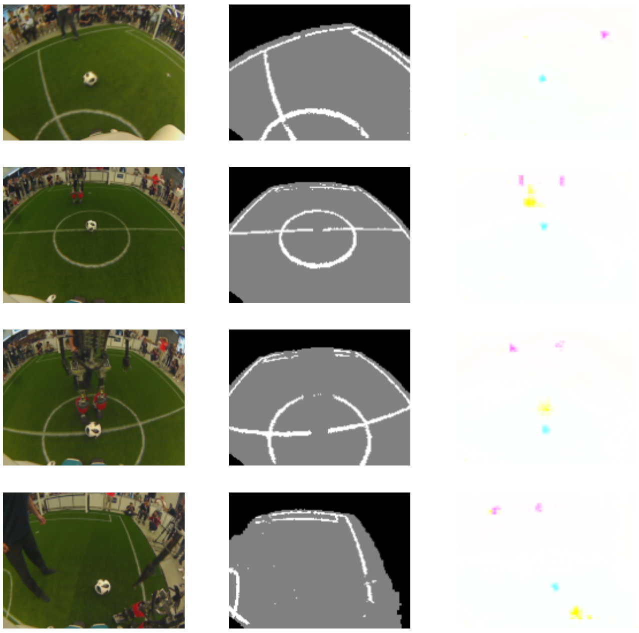


Fig. 6: Output of the predicted segmentation map and object detection blobs. First column represents the original image, second column shows the predicted segmentation map and last column depicts the detection blobs.

8 Acknowledgements

We would like to thank Hafez Farazi for his support and valuable feedback during the course of the project.

References

[1] Teresa Araújo et al. "UOLO-automatic object detection and segmentation in biomedical images". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 165–173.

[2] Vijay Badrinarayanan et al. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[3] Garrick Brazil et al. "Illuminating pedestrians via simultaneous detection & segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4950–4959.

[4] Jiale Cao et al. "Triply supervised decoder networks for joint detection and segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7392–7401.

[5] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *arXiv preprint arXiv:1704.00028* (2017).

[6] Bharath Hariharan et al. "Simultaneous detection and segmentation". In: *European Conference on Computer Vision*. Springer. 2014, pp. 297–312.

[7] Diego Rodriguez et al. "RoboCup 2019 AdultSize winner NimbRo: Deep learning perception, in-walk kick, push recovery, and team play capabilities". In: *Robot World Cup*. Springer. 2019, pp. 631–645.

[8] Olaf Ronneberger et al. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[9] Fabian Schnekenburger et al. "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.

[10] Yingying Yu et al. "A Two-Stream CNN With Simultaneous Detection and Segmentation for Robotic Grasping". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2020).

[11] Chong Zhang et al. "Toward Efficient Simultaneous Detection and Segmentation". In: *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE. 2018, pp. 1–5.