Soccer Robot perception

Lab Vision Systems
Rheinische Friedrich-Wilhelms-Universität Bonn
s6sskhan@uni-bonn.de, 3179769
s6temorb@uni-bonn.de, 3200842

Khan Saba, Morbagal Harish Tejas

April 10, 2020

Abstract

Object Detection and Image Segmentation has seen significant improvements in recent years through the use of Fully Convolutional Neural Network. This paper describes the implementation of a unified perception convolutional neural network based on encoder-decoder approach which can detect soccer-related objects such as robot, goalposts, soccer ball and simultaneously performs a multi-class semantic segmentation of the soccer field.

1 Introduction

The recent advancement in artificial intelligence and machine learning has contributed to the growth of computer vision and image recognition significantly. However, building a model which performs object detection and image segmentation simultaneously is a challenging task. We have implemented encoder-decoder based visual perception system that is robust against low lighting conditions and other such conditions that affect the performance of the model. The implemented unified model performs object detection and pixel-wise classification quite reliably.

2 Literature Survey and Related works

2.1 Fully Convolutional Network for Object Detection

The ability to learn the feature extraction step present in deep learning-based algorithms comes from the extensive use of convolutional neural networks (ConvNet or CNN). In this context, convolution is a specialized type of linear operation and can be seen as the simple application of a filter to a determined input. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in the input by tweaking the parameters of the convolution. The network can adjust itself to reduce the error and therefore learn the best parameters to extract relevant information on the database.

2.2 Fully Convolutional Network for Semantic Segmentation

Unlike the image classification or object detection tasks, semantic segmentation classifies each pixel of the image into its respective class. Any semantic segmentation architecture can be seen as an encoder network followed by a decoder net- work. Encoder network is a Convolutional Neural Network, usually pretrained. The decoder has to semantically project the distinctive features learned by the encoder onto the pixel space to get a dense classification. This method not only classifies each pixel but has to project the classified features learned at different stages of encoder onto the pixel space. A fully convolutional semantic segmentation network takes an image of any size and generates an output of the corresponding spatial dimensions.

2.3 U-Net

U-Net is a generic deep-learning solution for frequently occurring quantification tasks like cell detection and shape measurements in biomedical image data.U-Net architecture consists of three sections: The contraction, The bottleneck, and the expansion section. The contraction section is made of many contraction blocks. Each block takes an input,applies two 3X3 convolution layers followed by a 2X2 max pooling. The number of kernels or feature maps after each block doubles so that architecture can learn the complex structures effectively. The bottom-most layer mediates between the contraction layer and the expansion layer. It uses two 3X3 CNN layers followed by 2X2 up convolution layer. U-Net uses a rather novel loss weighting scheme for each pixel such that there is a higher weight at the border of segmented objects. This loss weighting scheme helped the U-Net model segment cells in biomedical images in a discontinuous fashion such that individual cells may be easily identified within the binary segmentation map.

2.4 SegNet

SegNet, is designed to be an efficient architecture for pixel-wise semantic segmentation. It is primarily motivated by road scene understanding applications which require the ability to model appearance (road, building), shape (cars,pedestrians) and understand the spatial-relationship (context) between different classes such as road and side-walk. In typical road scenes, the majority of the pixels belong to large classes such as road, building and hence the network must produce smooth segmentation. SegNet has an encoder network and a corresponding decoder network, followed by a final pixelwise classification layer.

2.5 Our approach

A unified deep-learning network is implemented based on encoder-decoder approach to perform object detection and pixel-wise classification on a dataset of 5,263 and 1,108 images respectively. The proposed model is trained alternately with different data loaders and loss functions to perform detection and segmentation tasks using the unified model.

3 Proposed Method

3.1 Problem Formulation

The idea is to recognize location of the ball, robots, goalposts and perform multi-class segmentation of soccer field. The network should be able to learn the features of the different objects in the image based on the heat-map distributions for object detection and features of lines and field based on image and target for segmentation. To achieve this, data is pre-processed for training, the proposed network is trained and a post-process method is used to detect the positions of objects in the output from model within a proximity of 4 pixels. The same trained network is used for pixel-wise segmentation. These processes are explained briefly below.

3.2 Pre-Processing

In the pre-processing phase, the images are rescaled to the picture size of 307200 pixels (640*480). For image detection, we applied some data augmentation methods like horizontal flip and slight color jitter to introduce variability in the data. Further, the normalization of the dataset is performed with the same values used in the Resnet18 pre-trained network. Furthermore, three heat-maps corresponding to three output channels are generated for robot, goalposts and ball respectively. Each heatmap uses Gaussian blob to represent the location of objects and has size 160*120 i.e. width/4 * height/4 pixels for the VGA pictures used.

For segmentation, the training images are resized to the picture size of 307200 pixels i.e (640*480) and the test images are resized to (160 * 120) to match the output from the model.

Heat-maps for test data under bright and low lighting conditions are as shown in Fig.1

3.3 Network

For detection, randomly selected 70 percent of dataset is pre-processed and given as input to the model along with heat-map details as teacher values. Similarly for segmentation, randomly selected 70 percent dataset containing the image of the soccer field along with it's segmentation mask is given to the model. A pre-trained ResNet-18 is chosen as the encoder and a shorter decoder is used to make it computationally effective. To utilize location-dependent features, the newly proposed location-dependent convolutional layer is used as the last layer. Different losses were used for different network heads. For detection head, the mean squared error and for segmentation head, cross-entropy loss is used. The network starts to learn the features for both detection and segmentation simultaneously. Once the network learns the features, the bounding boxes details of the test dataset are used to test the model for detection. The heat-maps generated by the model is given as input to the post-process to find the contour centers. To test the segmentation head, we visualize the output of the model for several images. We also calculate the pixel accuracy and IOU for the test data.

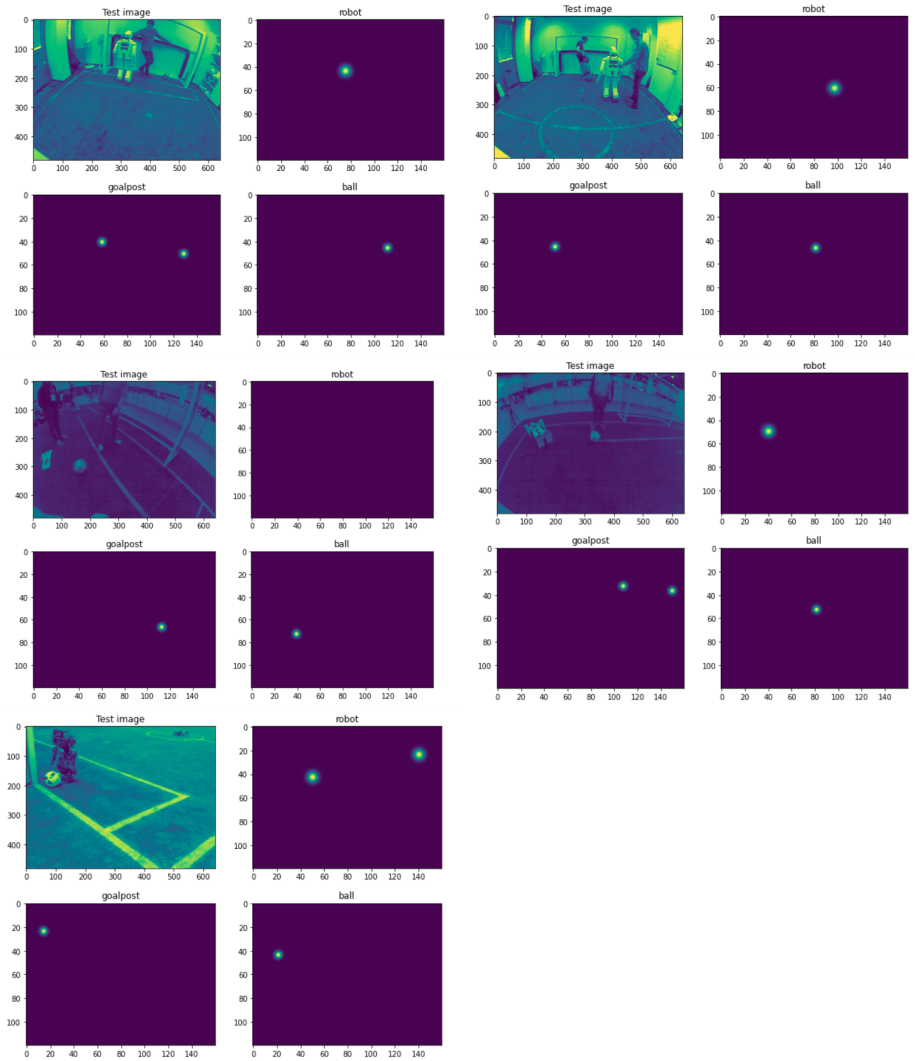

Figure 1: Test Heatmaps: Input image on top-left corner of each box with corresponding heatmaps

3.4 Post Processing

Once the features of the objects are detected, the network provides a 3 channeled output, where 3 output heads correspond to heatmaps similar to the teacher values with a gaussian blob at the center of the identified object and the same 3 output heads correspond to segmentation head to determine pixel-wise segmentation of field, lines and background. For detection, a post-processing approach is followed to identify these blobs in the output using OpenCV contour detection and the center of the contour is then compared to the center of the bounding box of the input images. The distance between them is tested for a threshold of 4 pixels. We accept the output heat-map as a correct prediction if threshold criterion is met. In case of segmentation, for each pixel in the image, we get the probabilities whether it belongs to field, lines or the background. We assign each pixel to the class with maximum probability and later calculate per pixel accuracy and IOU for segmentation.

3.5 Metric Calculation

3.5.1 Detection

The performance of the Detection is evaluated by calculating the accuracy, recall, precision, F1 score and False detection rate.

$$accuracy = \frac{positives}{TP + TN + FP + FN} \tag{1}$$

Accuracy is defined as the sum of true positives(TP) and true negatives(TN) divided by the sum of true positives(TP), true negatives(TN), false positives(FP), false negatives(FN).

$$recall = \frac{TP}{TP + FN} \tag{2}$$

Recall has been defined to be the number of true positives divided by the sum of true positives and false negatives.

$$precision = \frac{TP}{TP + FP} \tag{3}$$

Precision is defined as the number of true positives divided by the sum of true positives and false positives.

$$F1score = \frac{2 * precision * recall}{precision + recall} \tag{4}$$

F1 Score is the weighted average of Precision and Recall.

$$Falsedetectionrate = \frac{FP}{FP + TP} \tag{5}$$

False detection rate(FDR) is defined as the number of false positives divided by the sum of false positives and true positives

3.5.2 Segmentation

The performance of segmentation is evaluated by calculating pixel accuracy and Intersection over union(IOU) Pixel Accuracy is the percentage of pixels classified correctly. When considering the per-class pixel accuracy we're essentially evaluating a binary mask; a true positive represents a pixel that is correctly predicted to belong to the given class (according to the target mask) whereas a true negative represents a pixel that is correctly identified as not belonging to the given class.

$$PixelAccuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

The Intersection over Union (IoU) metric, also referred to as the Jaccard index, is essentially a method to quantify the percent overlap between the target mask and our prediction output

$$PixelAccuracy = \frac{target \cup prediction}{target \cap perdiction} \tag{7}$$

4 Implementation and Evaluation

4.1 Network Architecture

A pre-trained ResNet-18 model is used to implement our encoder part, and the final fully connected layer and the GAP layer are removed to make a connection directly to the decoder. To achieve real-time computation and to minimize the number of parameters in the network, the decoder part has designed to be short and it has three convolutional transpose layers with ReLU and Batch Normalization and an additional location-dependent Convolution layer to use location-dependent features. We have implemented a post-processing part where contours are detected from the outputs of the network and centers are calculated, comparing them with the original centers. The detection head, consisting of three output channels is used to identify robot, ball and goalpost and to identify field and lines. We use mean squared error for training for detection and cross-entropy for segmentation. The network architecture is shown in figure 2.

4.2 Training

The training of the model is performed simultaneously on images for both detection and segmentation and the losses are normalized during training. The layers of the network are grouped together into blocks and the blocks of the resnet18 pre-trained network are trained using Adam optimizer with a low learning rate of 0.00001. The other blocks are assigned with a learning rate of 0.01. The training data is augmented, and then shuffled. This makes the network reasonably robust to position, brightness and color shifts between frames. Also, they are re-scaled and randomly divided into training and testing data, and then passed as input to the network. The network converges after 50 epochs with a batch size of 5. For detection, the loss is calculated between the heat-map and output using the mean squared error(MSE) loss and for segmentation cross entropy loss is used between the output and the target segmentation mask. The learning curve is as shown in figure 3.

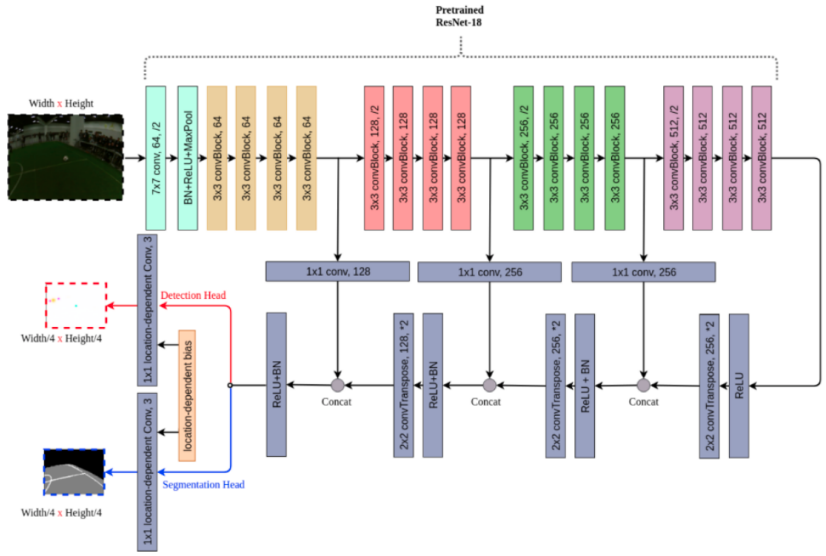

Figure 2: Network Architecture.[RoboCup 2019 AdultSize Winner NimbRo,Diego Rodriguez, Hafez Farazi, Grzegorz Ficht]

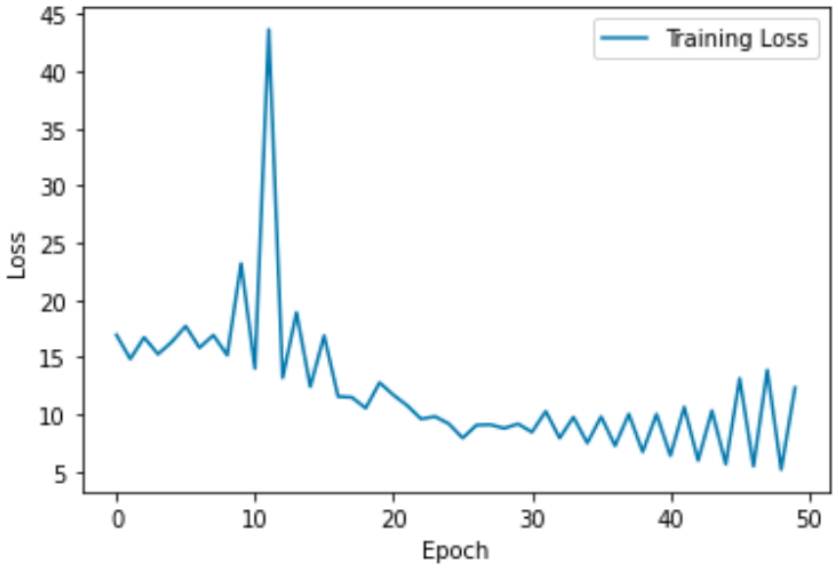

Figure 3: Learning Curve.

Type	F1 score	Accuracy	Recall	Precision	FDR
Ball	0.54	0.29	0.54	0.54	0.95
Robot	0.56	0.88	0.56	0.56	0.93
Goalpost	0.73	0.77	0.73	0.73	0.76
Total	0.61	0.65	0.61	0.61	0.88

Table 1: Results of the detection head.

Accuracy	IOU
0.74	0.46

Table 2: Overall Accuracy and IOU of Segmentation

4.3 Evaluation

For the evaluation for detection for robot, ball and goalposts, test dataset is prepared and given to the model which returns images along with the approximately expected center of the output objects. The output centers from the post process are evaluated against these centers. Then, true positive, true negative, false positive and false negative values are calculated for each of detection object based on the distance threshold of 4 pixels i.e. if the euclidean distance between the true center and output center is up to four pixels it is acceptable. Further, recall, false detection rates, accuracy, precision and F1 Score values are calculated based on above calculated values. Table 1. shows the evaluation result for the detection part. For Segmentation, test dataset is given to the trained model to get the segmentation results of the test images, then pixel-to-pixel comparison of target and output image is recorded for accuracy. The segmentation result is visualized in Figure 4 and the evaluation results are shown in the Table 2.

5 Conclusion

In conclusion, it is shown from the experiment that the model can learn spatial data for object detection and image segmentation tasks using the proposed encoder-decoder model. There is much scope of improvement in the model for better performance as training the unified model was challenging. However, adding a location-dependent bias layer to the model has shown a significant improvement in the model learning.

References

[1] *RoboCup 2019 AdultSize Winner NimbRo* Diego Rodriguez, Hafez Farazi, Grzegorz Ficht, Dmytro Pavlichenko, André Brandenburger, Mojtaba Hosseini, Oleg Kosenko, Michael Schreiber, Marcel Missura, and Sven Behnke. Autonomous Intelligent Systems, Computer Science, Univ. of Bonn, Germany.

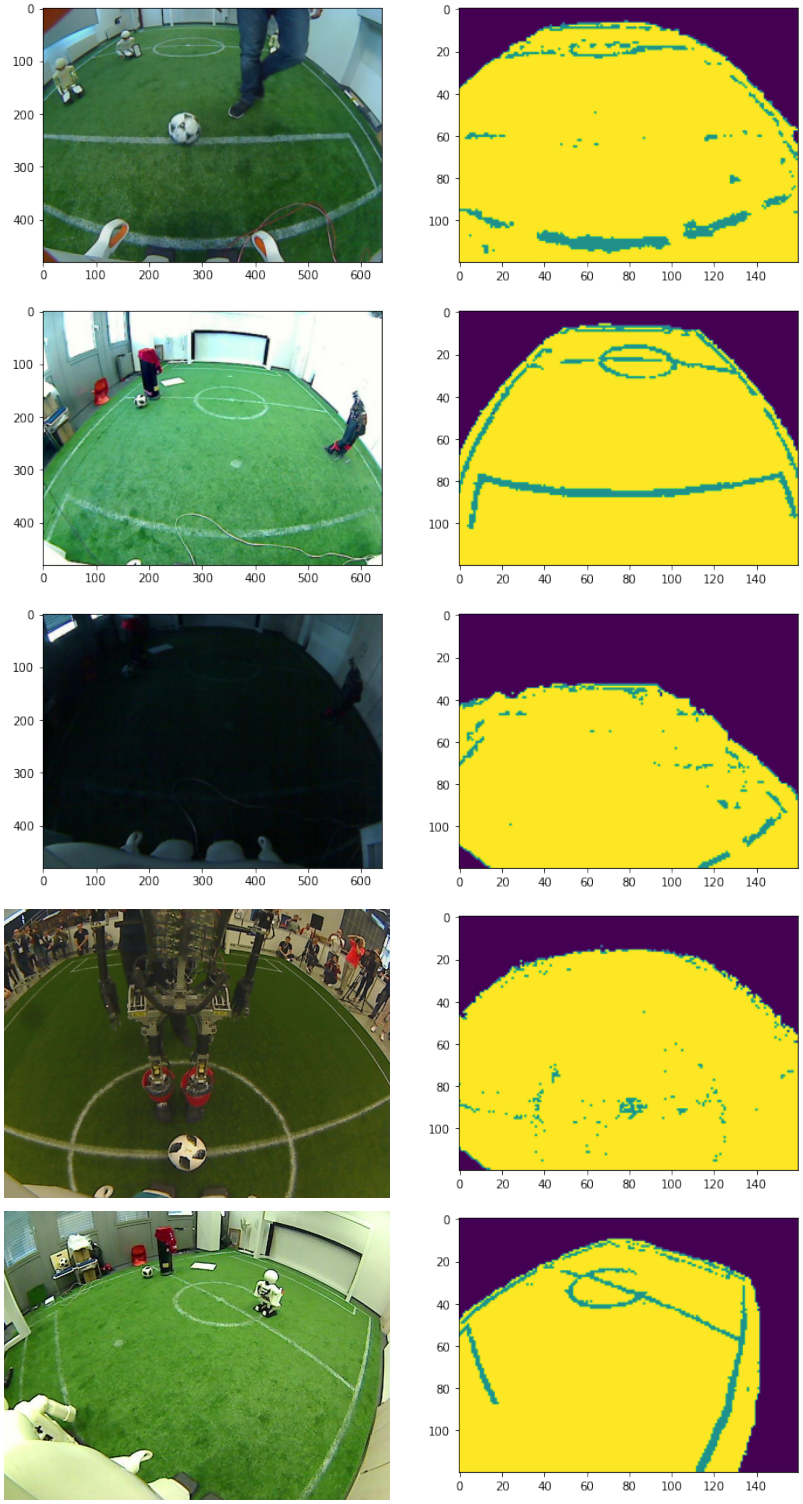

Figure 4: Segmentation Results : Input image on left side and segmentation result on right side

[2] *U-Net: Convolutional Networks for Biomedical Image Segmentation* Olaf Ronneberger, Philipp Fischer, and Thomas Brox Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany

[3] *SegNet: A Deep ConvolutionalEncoder-Decoder Architecture for Image Segmentation* Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla, Senior Member, IEEE

[4] *Location Dependency in Video Prediction* Niloofar Azizi, Hafez Farazi, and Sven Behnke Bonn University, Computer Science Department `https://github.com/AIS-Bonn/LocDepVideoPrediction/blob/master/Conv_PGP_LB.ipynb`