Detection of Humanoid Robot Body parts

Lab Cuda Vision - Learning Vision Systems on Graphics Cards

Vidiyala Laharika, Madupu Sravya Reddy

Universität Bonn
s6lavidi@uni-bonn.de, Matrikelnummer: 3209317
s6srmadu@uni-bonn.de, Matrikelnummer: 3209858

Abstract. A seemingly trivial task such as identifying an object in an image was a hard problem to solve for many years. Research in fully Convolutional Neural Networks has witnessed great success in this field. This paper describes the implementation of an encoder-decoder approach followed from recently proposed segmentation models SegNet, U-net and V-net for humanoid robot body part detection followed by post-processing.

1 Introduction

The wide applications like security surveillance, scene understanding and robotics in computer vision have triggered vast research in recent times. Visual recognition systems performing image classification, localization and detection have backed a huge research base owing to their relevance in these applications. Significant developments in the area of neural networks have fueled the improved performance of object detection which is a sub-domain of visual recognition systems.

The Convolutional neural network(CNN) is a network having a series of convolutional layers where the neurons in one layer are connected to the next layer. One of the main applications of these networks is image recognition, object detection, document analysis, etc. They extract features from the input image by learning patterns during the forward pass. They have been very successful in the object detection tasks and proved to produce good results for object localization.

This project aims to detect the 4 body parts of a humanoid robot. The RoboCup is a robotics soccer competition where the robot must detect objects on the soccer field like opponents, goal posts, boundaries, soccer balls, etc. and play the game. Our implementation is developed as the visual perception for the robot used in this competition[1] and tuned a few parameters to achieve better results.

2 Literature

2.1 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 segmen-

tation architecture can be seen as an encoder network followed by a decoder network. Encoder network is a Convolutional Neural Network, usually pre-trained. 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.

2.2 U-Net

Developed for biomedical image segmentation, this network is based on the fully convolutional network. The contracting or the encoder part and the decoder or the expansive part of the network are symmetric hence the name U-net. The encoder consists of the typical convolutional network made of multiple convolutions followed by ReLU and max pooling operations. It has the addition of skip connections which allow a more refined and precise output.

2.3 SegNet

Developed for outdoor and indoor scene understanding, SegNet consists of a sequence of encoder layers and respective decoder layers. The encoder is made of convolutional layers followed by ReLU, max-pooling and sub-sampling. The decoder up-samples the image using the max-pooling indices. This helps in retaining high-frequency details in the segmented images.

2.4 V-Net

The volumetric convolutional neural network, also known as V-net is a 3D image segmentation network based on volumetric, fully convolutional neural network. This network is trained on MRI volumes and predicts the segmentation for whole volumes at once. To deal with the situations where there is an imbalance between background and foreground voxels an objective function is optimized based on Dice coefficient.

2.5 Our approach

We followed the encoder-decoder approach combining some of the key ideas from the above three approaches to identify the objects of 4 classes. The object classes are 4 body parts of the humanoid robots i.e. Head, Trunk, Foot and Hands. Our work establishes 3 lateral connections from encoder to decoder part to provide high-resolution spatial data (inspired from U-Net) providing very low learning rate of 10^{-6} for pre-trained layers and 10^{-3} for the remaining layers. we used Adam optimizer.

3 Proposed Method

3.1 Problem Formulation

The idea is to recognize the body parts of robots in the soccer field. Based on the environment of the Robocup, it is assumed that at a time at-most 3 robots will be on the soccer field. So the network should be able to learn the features of the body parts in the image based on the heat-map distributions. To achieve this pre-processing is done to prepare the dataset for training, using this model network is trained and a post-process method is used to detect the positions of objects in output from model. These processes are explained briefly below.

3.2 Pre-Processing

In the pre-processing phase, the images are rescaled to the VGA picture size of 640480 pixels (width * height). Then we applied several augmentation techniques like horizontal flip, vertical flip and color jitter to significantly improve the diversity of data available. The normalization of the dataset is performed with the same values used in the Resnet18 pre-trained network. All these transformations are applied to the images and also on the bounding box details wherever applicable. We then created the ground truth values with certain probability values around the center of the bounding box with a standard deviation of 8 pixels and zero otherwise. 4 heat-maps corresponding to 4 output channels are generated. Each heat-map uses Gaussian probability distribution and has size 160*140 i.e. width/4 * height/4 pixels for the VGA pictures used. Examples of heat-maps used to training along with images are shown in Fig.1 & Fig.2.

3.3 Network

Dataset prepared in the pre-processing step will be given as input to the model along with heat-map details as teacher values. The network starts to learn the features based on these images and heat-maps. Once the network learns the features, a dataset with a bounding boxes are used to test the model. The heat-maps generated by the model is given as input to the post-process to find the contour centers.

3.4 Post Processing

Once the features of the objects are detected, the network provides an output with a 4 channeled heat-map similar to the teacher value with a gaussian blob at the center of the identified object. The network is restricted to detect parts of at most 3 robots in a single image. 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. If the distance between them is tested for different thresholds of 4,5 or 6 pixels. we accept the output heat-map as a correct prediction if threshold is not crossed.

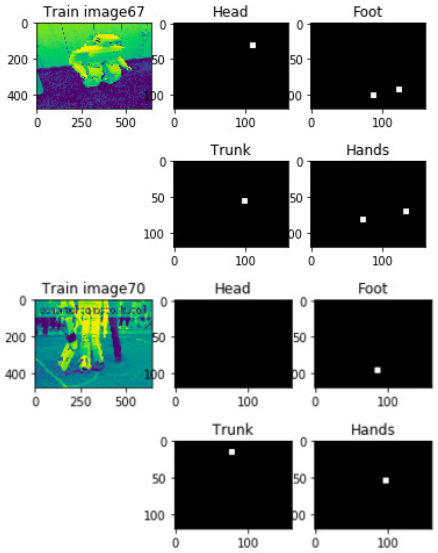

Fig. 1. Original Image(left), Heat-maps of corresponding images

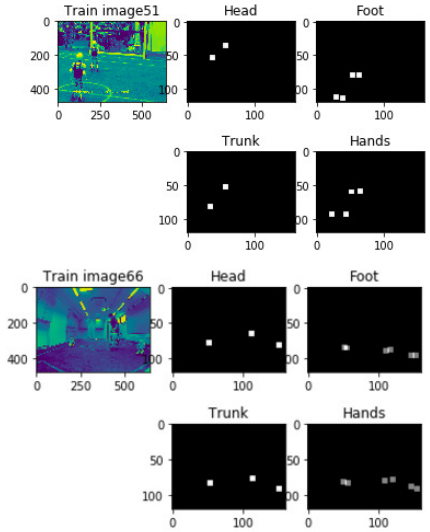

Fig. 2. Heat-maps for multiple robots

3.5 Metric Calculation

The performance of the network is evaluated by calculating the Recall, Accuracy and False detection rate.

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{1}$$

$$falsedetectionrate = \frac{false\ positives}{false\ positives + true\ positives} \tag{2}$$

$$Accuracy = \frac{true\ positives}{true\ positives + false\ positives + true\ positives} \tag{3}$$

True positive: The contour detected by the network is considered to be true positive if the euclidean distance between the center of the contour and center of the bounding box is within the range of 4,5 or 6 pixels.

False negative: When there is a bounding box detail in the true value but it is not detected by the network or detected center of the contour in the output is at a distance more than 4,5 or 6 pixels.

False positive: When there is no boundary box detail in the true value but is detected by the network.

4 Implementation and Evaluation

4.1 Network Architecture

We followed the encoder-decoder approach from recently proposed models like SegNet, V-Net and U-Net. A pre-trained ResNet-18 model is used to implement our encoder part, 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 four convolutional transpose layers with ReLU and Batch Normalization. Inspired by the U-Net model, 3 lateral connections are made from the encoder part to the decoder part to directly provide high-resolution details. The output of the decoder part has less spatial information than the full resolution image, so 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 object classes for the network to identify are head, trunk, hands and foot. The ground truths are created using the Gaussian blob around the center of the object bounding box. Similar to SweatyNet, we used mean squared error as loss function as center point detection is enough to identify the objects and not the full semantic segmentation loss.

We have tried to freeze the weights of the encoder part for the first 50 epochs as mentioned in the paper but did not achieve any satisfying results. So, We trained the whole network for 100 epochs to fit the data. The output channel of the network is 4 channeled, for 4 objects. The network architecture is shown below.

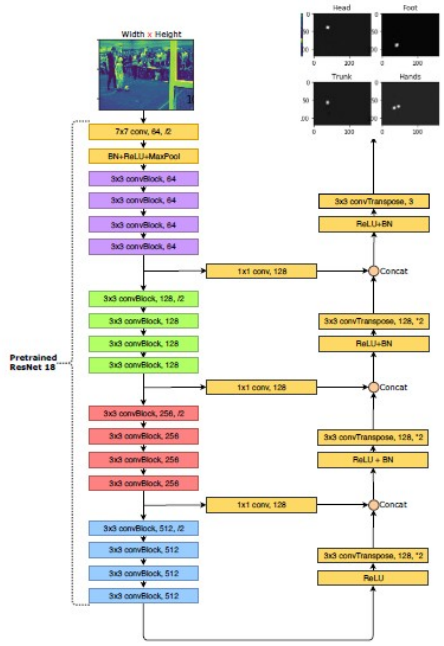

Fig. 3. The Network Architecture

4.2 Training

The training of the model is performed on 2128 images in total. The layers of the network are grouped together into blocks and the blocks of the resnet18 pre-trained network are trained with very a low learning rate of 10^{-6}.The other blocks are assigned with a learning rate of 10^{-3} . The training data is augmented, then shuffled with random horizontal flip, vertical flip and color jitter. This makes the network reasonably robust to position, brightness and color shifts between frames. Also they are rescaled and shuffled and passed as input to the network. The network converges after 60 epochs with a batch size of 5 but it is still trained to test against over-fitting. The loss is calculated between the heat-map and output using the mean squarred error(MSE) function. The network was also trained by freezing the weights of the pre-trained network and that did not yield any satisfactory results. The learning curve of the training phase is shown below in Fig 2.

4.3 Evaluation

A test dataset is prepared which returns images along with the approximately expected center of the output body parts. The output centers from the post process are evaluated against these centers. The recall and false detection rates are calculated based on the distance threshold of 4 pixels i.e. if the euclidean

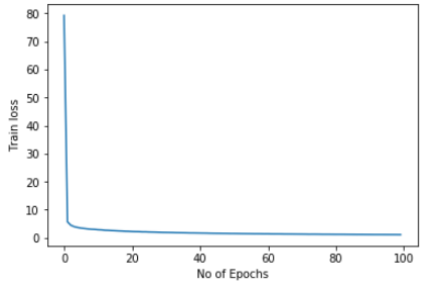

Fig. 4. Training Loss vs Epochs

distance between the true center and output center is upto 4 pixels is acceptable. As the dataset is limited to the precision of human eye. We have tested the model based on different distances. Output details for distances 4,5 & 6 pixels are mentioned in below tables. All the evaluations are performed on 816 unseen images. Outputs for different images for the four channels of the model are shown in Fig. 5. Two results on the left are best outputs from model and two outputs on the right hand side are not so satisfactory results from the model.

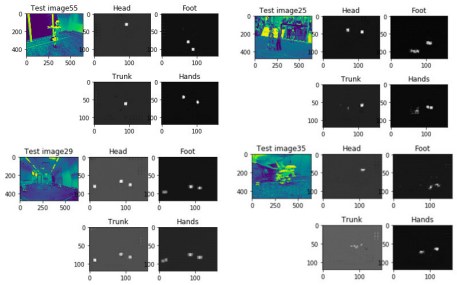

Fig. 5. Satisfactory(left) and unsatisfactory(right) results from the model

Object	RC	FDR	Total RC	Total FDR	Accuracy
Head	0.74	0.29	0.60	0.35	44%
Trunk	0.60	0.42			
Hands	0.59	0.34			
Foot	0.49	0.35			

Table 1. Data Records for distance of 4 pixels

Object	RC	FDR	Total RC	Total FDR	Accuracy
Head	0.96	0.08	0.79	0.15	68%
Trunk	0.81	0.21			
Hands	0.78	0.13			
Foot	0.61	0.18			

Table 2. Data Records for distance of 4 pixels

Object	RC	FDR	Total RC	Total FDR	Accuracy
Head	0.99	0.06	0.84	0.1	75%
Trunk	0.89	0.13			
Hands	0.84	0.07			
Foot	0.65	0.13			

Table 3. Data records for distance of 6 pixels

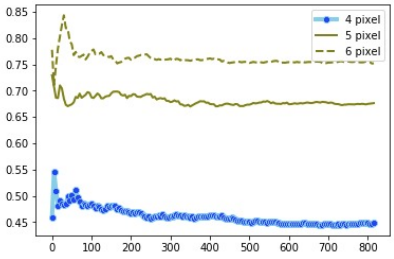

Fig. 6. Accuracy vs no.of testing samples

5 Conclusion & Future work

In conclusion, its shown from the experiment that the model can learn spacial data for object detection tasks. However there is still some room for improvements. Adding very low learning rate to the pre-trained layers has shown a significant improvement in the model learning. One of the challenges we faced for Foot detection are having the gaussian blobs overlapping with each other in the heat-map as the robot's foot are too close in the input image.

References

1. Grzegorz Ficht, Hafez Farazi, André Brandenburger, Diego Rodriguez, Dmytro Pavlichenko, Philipp Allgeuer, Mojtaba Hosseini, and Sven Behnke *NimbRo-OP2X: Adult-sized Open-source 3D Printed Humanoid Robot*
2. Badrinarayanan, Vijay et al. (2015). "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: CoRR abs/1511.00561. "rXiv: 1511.00561. URL: http://arxiv.org/abs/1511.00561.

3. Ronneberger, Olaf et al. (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: CoRR abs/1505.04597. arXiv: 1505.04597. url: http://arxiv.org/abs/1505.04597.
4. Diederik P. Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: CoRR abs/1412.6980 (2014). arXiv: 1412.6980. url: http://arxiv.org/abs/1412.6980.