Detection of humanoid body parts
Lab Vision Systems

Bhandary K, Swaroop; Pandey, Deepansh

Hochschule Bonn-Rhein-Sieg, Universität Bonn
swaroop.bhandaryk@smail.inf.h-brs.de, s6depand@uni-bonn.de,
Matrikelnummer: 9031891, 3057908

Abstract. This papers outlines an approach for detecting humanoid robot parts mainly in the context of Robocup soccer. The main challenge is to build a model which is not just highly accurate but is also light weight and extremely fast. In this work, we evaluate a deep learning model proposed by [1] for segmenting body parts such as head, hands, trunk and legs of a humanoid robot and post processing this image using traditional computer vision methods to determine the center points. The training was performed on a manually annotated dataset consisting of 1867 samples. The best performing model is chosen based on recall and false detection rate metrics. The model was able to achieve an recall and false detection rate of 15 and 6 respectively on test set.

1 Introduction

Convolutional Neural Networks (CNN) are the state of the art for almost all computer vision tasks such as image recognition, object detection, image segmentation to name a few[11][7]. The performance improvement of CNN models when compared to traditional neural network comes from the network exploiting such spatial relationship between the pixels in the image. The CNN models have been inspired by the organization of the visual cortex in the brain [8]. The first convolutional neural network model LeNet [9] was introduced by Yann Lecun et al. for the purpose of recognizing postal codes. However, the convolutional neural networks gained popularity after Alex Krizhevsky et al. used a deep convolutional neural network for classification of 1.2 Million high resolution images of ImageNet dataset [7]. A major part of this success came from training larger and deeper network on large amounts of labeled data. The features learned by the CNN models have been shown to be useful for other tasks such as object detection and image segmentation. Hence, the models for other computer vision tasks are models with modifications on top of an existing image classification architecture such ResNet[4], VGG[16], Densenet[5].

The task of object detection requires predicting the position of the object in the images, the bounding box encompassing the entire object and a object class. This task is significantly complicated when compared to the task of image classification where the network needs to predict a label for the entire image. In this work, we are mainly interested in detecting the center point of the object and object class. The current approaches in object detection can be mainly divided into two categories: single shot detectors and multi stage networks. Single shot detectors include approaches such as You Only Look Once(YOLO)[12] and Single Shot Multi-box Detector(SSD)[10] whereas multi stage approaches include R-CNN[3], Fast R-CNN[2], Faster R-CNN[13]. In [1], the authors first segment the humanoid body parts in the image and then use traditional computer vision methodologies to determine the object centers.

1.1 Contributions

The goal of this work is to build and evaluate a resource efficient deep learning model to detect the center points of various body parts of humanoid robots in the context of Robocup soccer. Hence, the contributions of this work can be divided as follows:

- Dataset creation: A dataset consisting of 1867 images have been created with manually annotated center points of various humanoid body parts.
- Evaluate a resource efficient model: The model described in [1] has been evaluated using metrics such as false detection rate and recall.

2 Related Work

This section gives a brief description of the various deep learning models that are mainly used for task of object detection.

2.1 Single shot methods

In single shot methods, only a single pass of the image is sufficient for detecting the objects present in the image. In the YOLO approach, the input image is divided into $s \times s$ grid cells. The grid containing the center point is responsible for the predictions of the object. Each grid cell is associated with b bounding boxes. The predictions of the network are encoded as $s \times s \times b \times (c + 5)$, where c is the number of classes in the dataset. The 5 values are the center x, center y, width, height, confidence. The network is trained end to end with a multi part loss function which is the squared error of the prediction for only the grids containing the object and squared error of confidence for all other grids.

In the SSD approach, multiple features maps with different resolutions are used for detecting the objects in the image. Default boxes and aspect ratios are first fixed. The network predicts the confidence score and offset for default bounding box coordinates using a kernel of size 3×3. Hence, for a feature map of size $m \times n$, the network will predict $m \times n \times k \times (c+4)$ output, where k is the number of default boxes and c is the number of classes. Hard negative mining is used to deal with imbalance between the positive and negative examples. The network is then trained end to end with the loss being the weighted sum of localization loss (Smooth L1) and confidence loss (Softmax)

2.2 Multi stage methods

The multi stage methods include the methods such as Region with CNN features (R-CNN), Fast R-CNN and Faster R-CNN. The R-CNN approach uses selective search algorithm to first determine the region of interest in the given image. In R-CNN 1867 such proposals are extracted. These proposals are then wrapped to a fixed square size and then passed through a CNN. An SVM then classifies these features and bounding box regressor is used to regress for the offsets of the bounding box. This method is very slow as 1867 passes are required to detect objects present in each image. In the subsequent papers the authors have improved the speed of training and prediction by attacking specific parts of the network. In the Fast R-CNN, the authors have extracted the region of interest on the convolution features itself. Hence, only a single pass of the network was sufficient to detect all the objects present in the image. In Faster R-CNN, the selective search is replaced with a region proposal network.

2.3 Segmentation + traditional computer vision

In [1], the authors have used a modified U-net[15] architecture to segment various objects such as ball, robot and goal posts. The author annotated the center point of the these objects and then during training use a gaussian blob of a fixed radius around the center point. The number of channels in the final layer is equal to the number of the objects to segment. The output of the network is then processed using traditional computer vision methods to find contours and determine the center points of the detected contours.

3 Network Architecture

The network architecture is as shown in figure 1. It consists of two main sections: encoder and decoder. The encoder network extracts features from the given input. Network models such as VGG, ResNet, DenseNet pretrained on ImageNet datasets can be leveraged for this purpose. Using a pretrained network will reduce the chances of the model overfitting to the dataset. ResNet model has been used in the encoder due to reduced complexity when compared to VGG model and also since the ResNet models facilitate better gradient flow with the skip connection. Resnet18 has been used as the encoder as it is readily available with pretrained weights in the Pytorch zoo. The last fully connected and Global Average Pooling layers have been removed to keep the network fully convolutional.

The decoder consists of four upsampling blocks. Each umsampling block consists of an ReLU activation layer, followed by batch normalization and finally transpose convolution. The stride of each upsampling block is set to 2. Hence the output of each upsampling block will increase spatial dimensions by a factor of 2. The decoder is designed to shorter than the encoder part. This has been done to reduce the model complexity and to achieve real time computation as described in [1]. Hence, the output of the network will spatially downsampled by a factor

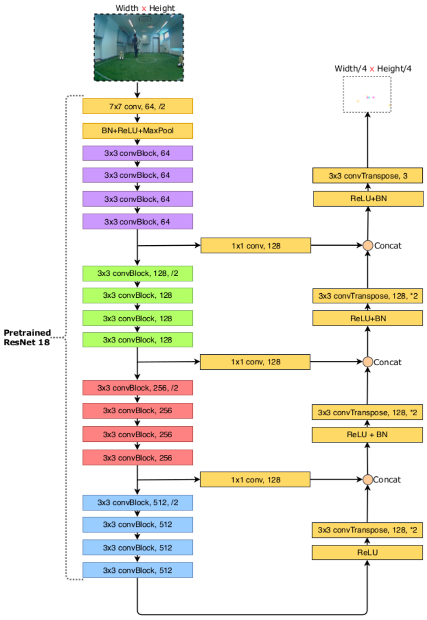

Fig. 1. Humanoid part detection network. Taken from [1]

of 4. Lateral connections have been added from the downsampling blocks to the corresponding upsampling blocks. This is because a lot of spatial information that is required to effectively upsample the image is lost in the encoder. Hence, including such lateral connection will enable to network to better recover spatial information.

4 Training

For the purpose of training the model a data set of 1867 images has been generated from various videos of robocup soccer competitions. The dataset includes robots such as Copedo, Dynaped, Igus and Nimbro-OPX2 [14]. The data set is annotated for 4 different output classes using labelimg [17] annotation tool. The output classes are robot different body parts such as head, trunk, hand and foot.

The dataset is divided into 70% for training data, 20% for validation data and 10% for test data. The batch size of 3 is used for training and validation set while batch size of 1 is used for test set. The network is first trained by keeping the encoder weights frozen for 150 epochs. The encoder weights are then unfreezed and the training is continued for 100 more epochs. The Adam optimizer[6] with a learning rate of $1e^{-4}$ and L2 weight decay of $1e^{-5}$ was used for training the model. Mean squared error is the loss function used.

5 Extracting center points from segmented images

The network segments various body parts of the humanoid robot from the input image. The center point of the various body parts are then obtained by first thresholding the network output and determining the contour present in the image. The centroid c of the contour can be defined as the average of all the points inside the contour.

$$c = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

This can also be done using image moments which help determine various properties such as radius, area, centroid.

$$C_x = \frac{M_{10}}{M_{100}} \qquad C_y = \frac{M_{01}}{M_{00}} \tag{2}$$

where C_x is the x coordinate and C_y is the y coordinate.

6 Results

The experiments are focused on validating the effectiveness of the model on the manually annotated dataset. In this sections the metrics used to evaluate the model and the results of the evaluation has been discussed.

6.1 Metrics

Recall is defined as True Positives divided true positives (TP) divided by the sum of true positive and false negative (FN). It defines the fraction of correct prediction over the total correct predictions.

$$RC = \frac{TP}{TP + FN} \tag{3}$$

False Detection Rate is defined as the ratio of false positives (FP) divided by the sum of false positives and true positive (TP)

$$FDR = \frac{FP}{TP + FP} \tag{4}$$

A detection is determined as true positive if the predicted value is within 4 pixel distance from the true value. The metric curves are shown in figure 2 and 3

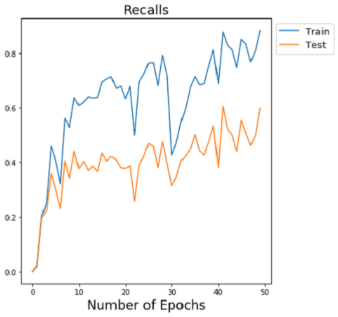

Fig. 2. Recall

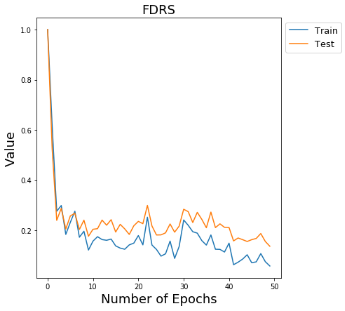

Fig. 3. False Detection Rate

6.2 Output

The figure 4 and figure 5 show the visualization for the centers of various body parts of humanoid robots. The color green show center of foot, blue for hand, yellow for trunk and red for head. The upper half of these figures depicts the predicted image plots while the lower half shows the plot for actual images. The figure 4 of bad predictions on the upper half shows the plots with no predicted center for body center while on the upper half of figure 5 of good predictions depicts the exact center points as annotated in the actual images.

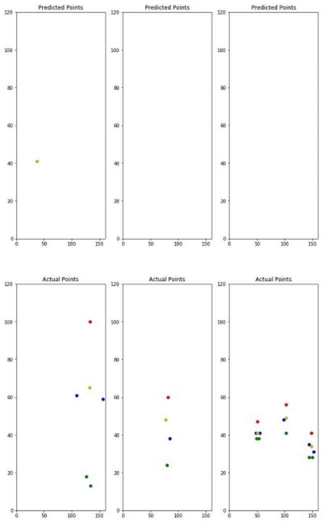

Fig. 4. Bad Predictions

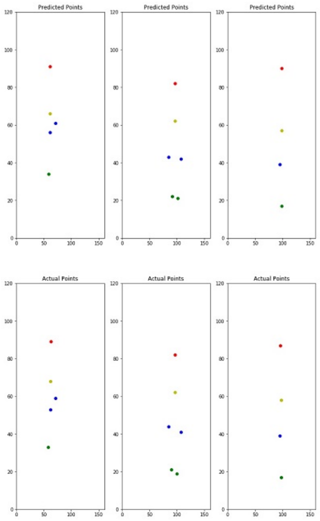

Fig. 5. Good Predictions

The figure 6 and figure 7 shows the original images for the plots shown in figure 4 and figure 5 respectively.

Fig. 6. Bad Predictions Images

Fig. 7. Good Predictions Images

7 Conclusion

A dataset of 1867 images was created from various Robocup soccer competition videos. A encoder-decoder framework with the encoder as pretrained network and a smaller sized decoder with tranposed convolutions is used to perform the task of segmenting the various body parts with traditional computer vision operations being performed on top of the network output to get the center points of the body parts. Training is done stage-wise by freezing and unfreezing the weights of the encoder network. The performance of the model is measured in terms of Recall and False Detection Rate. Table 1 shows the various metrics rounded to the nearest value. It was noticed the model was overfitting to the dataset. Hence, L2 regularization was used to combat the overfitting. We would like to conclude that model would provide a better performance with more training data.

	Train	Val	Test
Accuracy	84	58	54
FDR	14	14	15
Recall	88	60	6

Table 1. Metrics for humanoid part detection

Bibliography

[1] G. Ficht, H. Farazi, A. Brandenburger, D. Rodriguez, D. Pavlichenko, P. Allgeuer, M. Hosseini, and S. Behnke. Nimbro-op2x: Adult-sized open-source 3d printed humanoid robot. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, pages 1–9, Nov 2018.

[2] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[5] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, July 2017.

[6] D. P Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[8] M. Laskar, L. G S. Giraldo, and O. Schwartz. Correspondence of deep neural networks and the brain for visual textures. *arXiv preprint arXiv:1806.02888*, 2018.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. Ssd: Single shot multibox detector. 2016.

[11] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. pages 779–788, 06 2016.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[14] Robots. University of bonn. URL `http://www.ais.uni-bonn.de/nimbro/Humanoid/robots.html`.

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[17] Darrenl Tzutalin. Labelimg. URL `https://github.com/tzutalin/labelImg`.