International Journal for Modern Trends in Science and Technology, 6(12): 01-07, 2020

ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST061201
Available online at: http://www.ijmtst.com/vol6issue11.html

Real-Time Person Segmentation – Based on Body Pix

Anish Mankotia[1] | Meenu Garg[2]

[1]B.Tech scholar, Department of IT Maharaja Agrasen Institute of Technology.
[2]Assistant Professor, Department of IT Maharaja Agrasen Institute of Technology.

To Cite this Article
Anish Mankotia and Meenu Garg, "Real-Time Person Segmentation – Based on Body Pix", *International Journal for Modern Trends in Science and Technology*, 6(12): 01-07, 2020.

Article Info
Received on 06-November-2020, Revised on 18-November-2020, Accepted on 25-November-2020, Published on 29-November-2020.

ABSTRACT

In this paper, we propose a novel semantic segmentation-based on the body pix module of the Tensor flow.js which can keep up with the accuracy of the state-of-the art approaches while running in real time. The solution follows the convolutional neural networks, each step in the workflow being enhanced by additional information from semantic segmentation. Therefore, we introduce several improvements to computation, aggregation, and optimization by adapting existing techniques to integrate additional surface information given by each semantic class. Using the body pix model which is trained using CNN, the ResNET50, this network can work with more than 150 layers, removing the problem of vanishing gradients. Using this network our body pix module, creates a more accurate and defined segmentation, and also supports multi-person segmentation.

Index Terms—Semantic Segmentation, Convolutional Networks, Deep Learning

INTRODUCTION

CONVOLUTIONAL networks are inducing advances in recognition. These are not only improving for whole-image classification [1], but also making progress on tasks which are concerned with a defined and structured output. These constitute of advances in key-point prediction [7], [8], bounding box object detection [4] and local correspondence [8], [9].

The natural steps required in the progression from opaque to fine inference is to make a prediction at every pixel. Previously researched approaches have used convnets for semantic segmentation [10], [11], [12], [13], in which each enclosed object is labeled with its corresponding pixel.

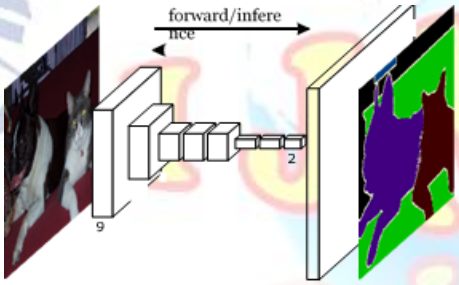
We show that fully convolutional networks (FCNs) trained end-to-end, pixels-to-pixels on the ResNET 50 network, exceed the previous best results without further machinery. To our knowledge, this is the first work to train FCNs end-to-end (1) for pixelwise prediction and (2) from supervised pre-training. This method can be used for augmented reality and photo editing effects, it is now widely being used for medical diagnostics as well.

The earlier technologies use the patchwise training[10], [16] but lacks the efficiency of full convolution training. Our approach does not make use of pre or post processing complications, using the Body Pix module which is trained using the ResNET architecture, it is trained to do pixel wise

body segmentation and is trained on 24 body parts as classes.

The model is able to do real-time segmentation of the video, differentiating all the different identifiable parts of the human body, Semantic segmentation faces an inherent conflict between semantics and locality: global information resolves *what* while local information resolves *where*. What can be done to navigate this spectrum from location to semantics? How can local decisions respect global structure? It is not immediately clear that deep networks for image classification yield representations sufficient for accurate, pixelwise recognition.

The following diagram shows the internal working of the convolutional neural network, which can efficiently be used for per-pixel task like segmentation. The process involves multiple layers, unlike the Alex net which only operated on 8 convolution layers, the ResNet 150 operates on more than 100 layers and is hence provide a better and more concise segmentation than its previous counterparts.



II. RELATED WORK

Our approach draws on earlier successes of deep net for image classification [1], and transfer learning [18],[19]. Transfer was first demonstrated on various visual recognition tasks then on detection, and on both instance and semantic segmentation in hybrid proposal- classifier [14], [15]. We now re-architect and fine-tune classification nets to direct, dense prediction of semantic segmentation. We chart the space of Fully convolutional networks and relate previous models both historical and recent.

To our knowledge, the idea of extending a convnet to arbitrary-sized inputs first appeared in Matan *et al.* [20], which was an extension the classic LeNet [21]to recognize strings of digits. Because their net was only limited to one-dimensional input strings not multi-dimensional, Matan *et al.* used Viterbi decoding to obtain their outputs. Wolf and Platt [22]used a sort of similar technique to expand convnet outputs to 2-dimensional maps of detection scores for the four corners of the postal address blocks. Both of these historical works use the convolution neural networks for inference and detection, however they do not support multi-person segmentation and multiclass segmentation. Ning *et al.* [10] define a convnet for coarse multiclass segmentation of *C. elegans* tissues with fully convolutional inference.

Fully convolutional computation has also been used in the present-era in the present era of many-layered nets. Image restoration by Eigen *et al.* [23]. Sliding window detection by Sermanet *et al.* [4], image restoration by Eigen *et al.* [23] do fully convolutional inference. Fully convolutional training is rare and difficult , but used in a very efficient manner by Tompson *et al.*[24] to learn a part detector and spatial model for pose estimatation analysis, although they do not clarify on or analyze this method.

Several recent works have applied convnets to dense prediction problems, including semantic segmentation by Ning *et al.* [10], Farabet *et al.* [12], and Pinheiro and Collobert [13]; boundary prediction for electron microscopy by Ciresan *et al.* [11]and for natural images by a hybrid convnet/nearest neighbor model by Ganin and Lempitsky [16]; and image restoration and depth estimation by Eigen et al. [23], [25]. Common elements of these approaches include,small models restricting capacity and receptive fields;

- patchwise training [10], [11], [12], [13], [16];
- refinement by superpixel projection, random field regularization, filtering, or local classification [11], [12], [16];
- "interlacing" to obtain dense output [4], [13], [16];
- multi-scale pyramid processing [12], [13], [16];
- saturating tanh nonlinearities [12], [13], [23]; and
- ensembles [11], [16], whereas our method does without this machinery. We use the ResNET50 architecture, which works on multiple layers of CNN and produce multi-person segmentation.

Unlike these existing methods, we adapt and extend deep classification architectures, using image classification as supervised pre-training, and fine-tune fully convolution- ally to learn simply and efficiently from whole image inputs and whole image ground truths.

Hariharan *et al.* [14] and Gupta *et al.* [15] likewise adapt deep classification nets to semantic segmentation, but do so in hybrid proposal-classifier models. These approaches fine-tune an R-CNN system [5] by sampling bounding boxes and/or region proposals for detection,

semantic segmenta- tion, and instance segmentation. Neither method is learned end-to-end. They achieve the previous best segmentation results on PASCAL VOC and NYUDv2 respectively.

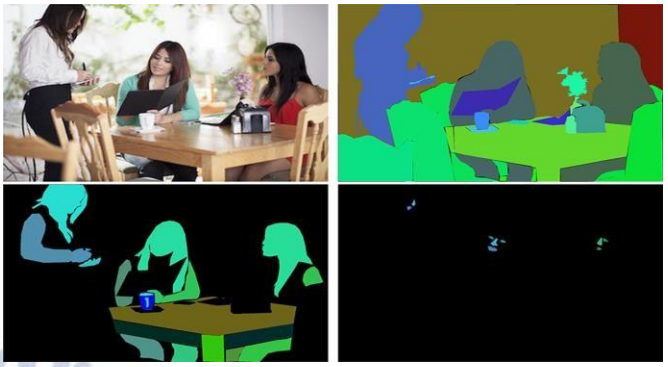
We fuse features across layers to define a nonlinear local-to-global representation that we tune end-to-end and layer to layer . The Laplacian pyramid [26] is a classic multi-scale representation made of fixed smoothing and differencing. The jet of Koenderink and van Doorn [27] is a rich, local feature defined by compositions of partial derivatives. In the context of deep networks, Sermanet *et al.* [28] fuse intermediate layers but doesn't take into account the resolution in doing so. In a contemporary work done by Hariharan *et al.* [29] and Mostajabi *et al.* [30] also fuse multiple layers but do not learn end-to-end and rely on fixed bottom-up grouping.

In addition, new works have improved the FCNs pre- sented here to further advance the state-of-the-art in se- mantic segmentation. The DeepLab models [39] raise output resolution by dilated convolution and dense CRF inference. The joint CRFasRNN [40] model is an end-to-end integra- tion of the CRF for further improvement. ParseNet [41] normalizes features for fusion and captures context with global pooling. The "deconvolutional network" approach of [42]restores resolution by proposals, stacks of learned deconvolution, and unpooling. U-Net [43] combines skip layers and learned deconvolution for pixel labeling of mi- croscopy images. The dilation architecture of [44] makes thorough use of dilated convolution for pixel-precise output without a random field or skip layers.

These traditional methods do not provide the definition and the accuracy by which we perform segmentation . Our network is constructed by repeating a building block that aggregates a set of transformations with the same topology. Our simple design results in a homogeneous, multi-branch architecture that has only a few hyper-parameters to set. This strategy exposes a new dimension, which we call "cardinality" (the size of the set of transformations), as an essential factor in addition to the dimensions of depth and width.

III. METHODOLOGY

As discussed in the abstract, the basic motive is to build a model which can perform real time- body segmentation of the different body parts of the human body, the output can look something like this:



The above image shows the segmented parts of the human body from the background in a still image. The result is somewhat similar in our case, the only difference is that the segmentation is done in real time on moving human beings. Our approach consists of using the Body Pix module of Tensor Flow JS, along with the react web-cam. The react webcam helps in accessing the camera of the particular machine that we are running the model on. The Body Pix model is trained on the ResNET50 architecture. The advantage of this particular architecture from its predecessors is that it has the ability to work with multiple layers.

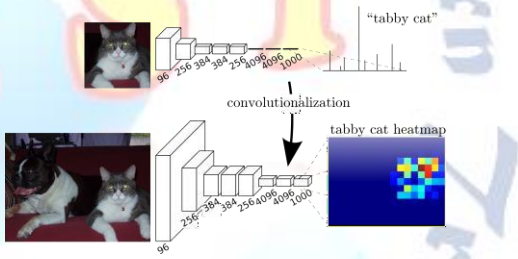


Fig. 2. Transforming fully connected layers into convolution layers enables a classification net to output a spatial map. Adding differentiable interpolation layers and a spatial loss produces an efficient machine for end-to-end pixelwise learning

The BodyPix module contains a function called the "drawMask", which is used to provide the outlining of the image from the background, and highlighting body parts with different colors. Further we will describe in detail the internal working of the ResNET50 architecture over which the BodyPix model is maid.

Adapting classifiers for prediction of pixels

Typical recognition nets, including LeNet [21], AlexNet [1], and its deeper successors [2], [3], ostensibly take fixedsized inputs and produce non-spatial outputs. The fully connected layers of these nets have fixed dimensions and throw away spatial coordinates. However, fully connected

layers can also be viewed as convolutions with kernels that cover their entire input regions.
In our approach the ResNet is trained on more than a million images from the ImageNET. The network can classify images of upto 1000 object classification categories for this reason this network has learned rich feature representations of a wide range of images.

One reason for which CNN is used for our purpose is because of the spatial output maps of these models. With ground truth available at every output cell, both the forward and backward passes are straightforward, and both take advantage of the inherent computational efficiency (and aggressive optimization) of convolution. The corresponding backward times for the AlexNet example are 2.4 ms for a single image and 37 ms for a fully convolutional 10 × 10 output map, resulting in a speedup similar to that of the forward pass.

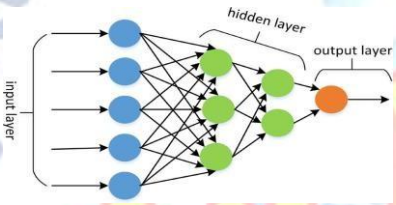
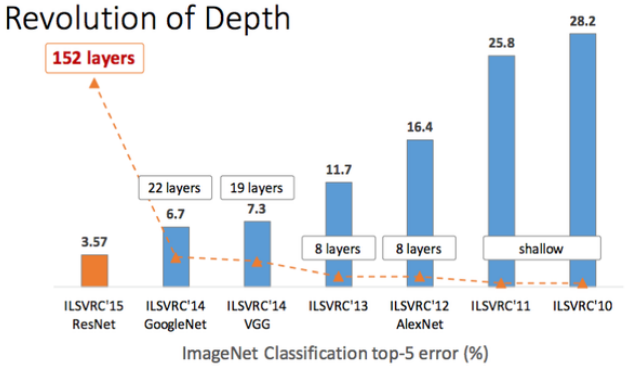
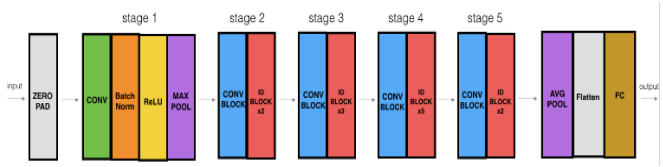


Fig.3 An overview of the the CNN network.

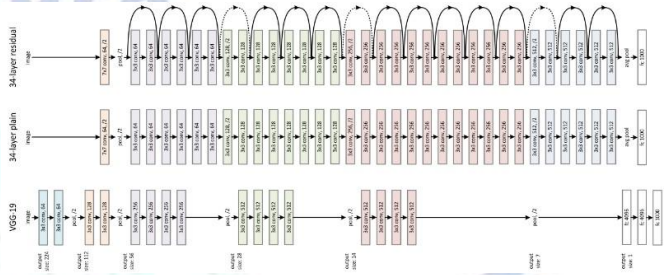
ResNET50 architecture

The ResNET50 architecture provides a efficient as well as a fast method for segmentation as It works on more than 150 layers and can support multi-person segmentation with more defined results.
AlexNet, the winner of ImageNet 2012 and the model that apparently kick started the focus on deep learning had only 8 convolutional layers, the VGG network had 19 and Inception or GoogleNet had 22 layers and ResNet 152 had 152 layers.





The ResNet-50 model consists of 5 stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.



Components of a network include 3x3 filters, CNN down-sampling layers with stride 2, global average pooling layer and a 1000-way fully connected layer with softmax in the end. ResNET is advanced as it uses a skip connection on the original output of the convolution block, hence helps in solving the problem of the vanishing gradient, which very prevalent in the previous architectures like the AlexNet. In addition to this, it uses the identity function which helps higher layer to perform as efficiently as as the lower layer and not worse.

Body-Pix Implementation

This model can be used to segment an image into pixels that are and are not part of a person, and into pixels that belong to each of twenty-four body parts. It works for multiple people in an input image or video.
The following command can be used to load the model:-
npm install @tensorflow-models/body-pix
Body-Pix comes with a few different versions of the model, with different performance characteristics trading off model size and prediction time with accuracy.
In order to load the model:-
const net = await bodyPix.load();
By default, BodyPix loads a MobileNetV1 architecture with a 0.75 multiplier. This is recommended for computers with mid-range/lower-end GPUs. A model with a 0.50 multiplier is recommended for mobile. The ResNet architecture is recommended for computers with

even more powerful GPUs as it is more efficient and works with a large number of layers.

The following code is run in order to load the Body-Pox model with the ResNET architecture:

```
const net = await bodyPix.load({
architecture: 'ResNet50',
outputStride: 32,
quantBytes: 2
});
```

a) Config params in bodyPix.load()

• • architecture - Can be either MobileNetV1 or ResNet50. It determines which BodyPix architecture to load.

• • outputStride - Can be one of 8, 16, 32 (Stride 16, 32 are supported for the ResNet architecture and stride 8, and 16 are supported for the MobileNetV1 architecture). It specifies the output stride of the BodyPix model. The smaller the value, the larger the output resolution, and more accurate the model at the cost of speed. *A larger value results in a smaller model and faster prediction time but lower accuracy*.

• • multiplier - Can be one of 1.0, 0.75, or 0.50 (The value is used *only* by the MobileNetV1 architecture and not by the ResNet architecture). It is the float multiplier for the depth (number of channels) for all convolution ops. The larger the value, the larger the size of the layers, and more accurate the model at the cost of speed. A smaller value results in a smaller model *and faster prediction time but lower accuracy***.**

quantBytes - This argument controls the bytes used for weight quantization. The available options are:

• • 4. 4 bytes per float (no quantization). Leads to highest accuracy and original model size.

• • 2. 2 bytes per float. Leads to slightly lower accuracy and 2x model size reduction.

• • 1. 1 byte per float. Leads to lower accuracy and 4x model size reduction.

The following table contains the corresponding BodyPix 2.0 model checkpoint sizes when using different quantization bytes:

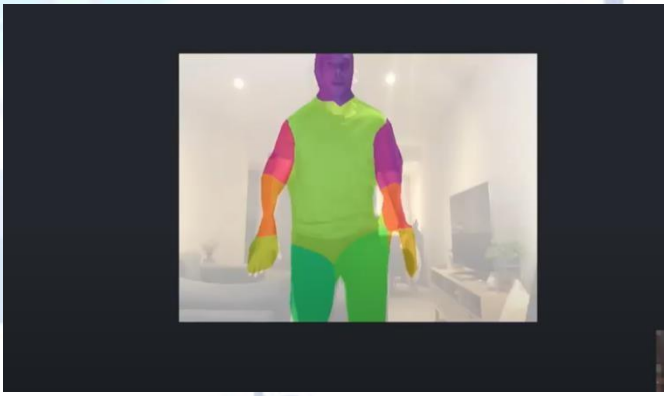
Architect ure	**quantBytes=4**	**quantBytes=2**	**quantBytes=1**
ResNet50	~90MB	~45MB	~22MB
MobileNet V1(1.00)	~13MB	~6MB	~3MB
MobileNet V1(0.75)	~5MB	~2MB	~1MB
MobileNet V1(0.50)	~2MB	~1MB	~0.6MB

In the above table we can see that the ResNET50 provides the best accuracy and prediction compared to the other architectures.

IV. RESULTS

In this paper we used the body-pix module, which is based on the ResNET50 framework in order to do real-time body segmentation of the human from the background, we provided theoretical as well as practical explanation as to why this new approach is better than the previous architectures designed for the same purpose.

Apart from this we added the real-time factor using the react-webcam, using this library we were successfully able to access out system's camera, providing real-time segmentation of the person. The accuracy and definition of our model beats, the definition of the previously used architectures such as the AlexNet, MobileNet and FCN network, although a bit slower and a bit more GPU intensive, the accuracy and definition is unparalleled by far. Machine power is not much of a threshold in our day-by-day technologically advancing world. This technology of semantic segmentation is now leading to rapid improvements in medical diagnostics, autonomous vehicles and materials.



The above image shows a still from a real-time segmentation of the user, from the front camera of the system. The model is capable of distinguishing upto 24 parts of the human body, and it is not just applicable on one person as shown in the image, there can be multiple people in the video. If the image is captured without any disruptions, the model successfully gives a sematic segmentation of all the people visible in front of the camera, in real-time.

V.CONCLUSION

Semantic image segmentation is a key application in image processing and computer vision domain. Besides briefly reviewing on traditional semantic image segmentation, this paper comprehensively provides an approach which provides better definition and accuracy, taking the help of some react libraries in order to demonstrate the functionalities. This paper basically states how the recent progress in semantic image segmentation, especially based on ResNET50 has provided better results than the previously used architectures. Till now, more and more methods are emerging to make semantic image segmentation more accurate or faster or both on accuracy and speed. We hope this review on recent progress of semantic image segmentation can make some help to researchers related to this area.

REFERENCE

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classifi- cation with deep convolutional neural networks." in *NIPS*, 2012. 1,2,3,5

[2] K. Simonyan and A. Zisserman, "Very deep convolutional net- works for large-scale image recognition," in *ICLR*, 2015.1,2,3, 5

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.1,2,3,5

[4] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Le- Cun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *ICLR*, 2014.1,2,4

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmen- tation," *PAMI*, 2015.1,2,8

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014.1

[7] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *ECCV*, 2014, pp. 834–849.1

[8] J. Long, N. Zhang, and T. Darrell, "Do convnets learn correspon- dence?" in *NIPS*, 2014.1

[9] P. Fischer, A. Dosovitskiy, and T. Brox, "Descriptor matching with convolutional neural networks: a comparison to SIFT," *arXiv preprint arXiv:1405.5769*, 2014.1

[10] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano, "Toward automatic phenotyping of developing embryos from videos," *Image Processing, IEEE Transactions on*, vol. 14, no. 9, pp. 1360–1371, 2005.1,2,4,7

[11] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images." in *NIPS*, 2012, pp. 2852–2860.1,2,4,7

[12] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *PAMI*, 2013.1,2,4,7, 8

[13] P. H. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *ICML*, 2014.1,2,4,7,8

[14] B. Hariharan, P. Arbel a´ez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *ECCV*, 2014.1,2,4,5,7,8,9

[15] S. Gupta, R. Girshick, P. Arbelaez, and J. Malik, "Learning rich fea- tures from RGB-D images for object detection and segmentation," in *ECCV*, 2014.1,2,8

[16] Y. Ganin and V. Lempitsky, "N 4-fields: Neural network nearest neighbor fields for image transforms," in *ACCV*, 2014.1,2,7

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional net- works for semantic segmentation," *CVPR*, 2015.1,2

[18] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *ICML*, 2014.2

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding con- volutional networks," in *ECCV*, 2014, pp. 818–833.2,4

[20] O. Matan, C. J. Burges, Y. LeCun, and J. S. Denker, "Multi-digit recognition using a space displacement neural network," in *NIPS*, 1991, pp. 488–495.2

[21] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to hand- written zip code recognition," in *Neural Computation*, 1989.2,3

[22] R. Wolf and J. C. Platt, "Postal address block location using a convolutional locator network," in *NIPS*, 1994, pp. 745–745.2

[23] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *ICCV*, 2013, pp. 633–640.2

[24] J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *NIPS*, 2014.2

[25] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014. 2

[26] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *Communications, IEEE Transactions on*, vol. 31, no. 4, pp. 532–540, 1983.2,6

[27] J. J. Koenderink and A. J. van Doorn, "Representation of local geometry in the visual system," *Biological cybernetics*, vol. 55, no. 6, pp. 367–375, 1987.2,6

[28] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedes- trian detection with unsupervised multi-stage feature learning," in *CVPR*, 2013.2

[29] B. Hariharan, P. Arbel a´ez, R. Girshick, and J. Malik, "Hyper- columns for object segmentation and fine-grained localization," in *CVPR*, 2015.2

[30] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedfor- ward semantic segmentation with zoom-out features," in *CVPR*, 2015.2

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.2

[32] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015.2

[33] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *PAMI*, 2015.2

[34] P. Fischer, A. Dosovitskiy, E. Ilg, P. Husser, C. Hazırba, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Learning optical flow with convolutional networks," in *ICCV*, 2015.2

[35] D. Pathak, P. Kr a¨henbu¨ hl, and T. Darrell, “Constrained convolu- tional neural networks for weakly supervised segmentation,” in *ICCV*, 2015.2
[36] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille, “Weakly- and semi-supervised learning of a DCNN for semantic image segmentation,” in *ICCV*, 2015.2
[37] J. Dai, K. He, and J. Sun, “Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation,” in *ICCV*, 2015.2