Fully Convolutional Network for Clothing Parsing

Raymond Li
University of Toronto
raymo.li@mail.utoronto.ca

Abstract

In this report, I propose a fully convolutional neural network for parsing clothing on the fashion photograph in [16]. Based on the idea of a fully convolutional network (FCN) for semantic segmentation described in [14], I designed a similar approach using the smallest variant of VGG [15] as the backbone classification network. In the first task of binary pixel classification, I also employed the state-of-the-art instance segmentation network Mask R-CNN [4] pre-trained on MS COCO [9] for results comparison.

1 Introduction

Convolutional networks have been the driving forcing behind most recent advances in computer vision applications. With the help of deep learning, problems such as classification, object detection, semantic segmentation and instance segmentation have seen tremendous improvement in recent years. In addition, the popularity of fully convolutional network (FCN) has also increased as they are easily trained end-to-end and includes the desired properties of translation invariance across all layers of the feature maps. Some examples of the usage of such networks includes F-RCN [2] and RetinaNet [8] for object detection, InstanceFCN [1] for instance segmentation, as well as the work that my approach is based on [14].

The main difference between instance segmentation and semantic segmentation is that instance segmentation is usually on object detection where multiple proposals are usually produced along with a class label and mask for each instance of the object. On the other hand, semantic segmentation attempt to partition the images into semantically meaningful parts and classify each part into one of the predetermined classes. For the specific task of clothing paring on fashion photographs [16], the original paper proposed first using contour detection for obtaining super-pixels, then training a predictive graphic model (CRF) using logistic regression. However, with the recent advances of deep learning and convolutional networks, I opted to tackle this problem with a FCN for semantic segmentation trained end-to-end on a pixel-level.

There is also the option of using instance segmentation to tackle this problem. However, the original paper described this task as a semantic problem, such that we are looking detecting overlapping instances of objects in the image but simply classifying the pixels into one of the pre-defined classes. In addition, most instance segmentation networks have an involved multi-stage training pipeline that requires intensive computational power that I simply cannot afford. However, for the first task of performing binary segmentation between the person and background, I used a pre-trained model of state-of-the-art Mask R-CNN [4], and compared the results against my implementation of the much lighter FCN.

Although there have been more recent work of using FCN for semantic segmentation such as [17] and [13], the work that my approach was based on was one of the first attempts of utilizing FCN for this task. In addition, it can be easily implemented with the help of deep learning library PyTorch [10] due to the simplicity of its design and training pipeline. The entire training process was done on a laptop over-night as described in the later section.

2 Approach

2.1 Feature Extraction

The feature maps of convolution networks are great for feature extractions due to their ability to preserve spatial information across different layers. In particular, classification network are employed as feature extractor for task such as object detection [11][12], and segmentation [4][14]. In particular, VGG [15] and ResNet[5] have been the favourites for transfer learning due to their accurate predictions and fully convolutional designs.

Similar to the approach in [14], I used VGG-11, the smallest of its variants, pre-trained on ImageNet [3] as implemented in the TorchVision library, to be the feature extractor for my network. To preserve the translation invariance of the feature map, I removed all layers after the maxpool4 to be consistent with our fully-convolutional design.

2.2 Network Architecture

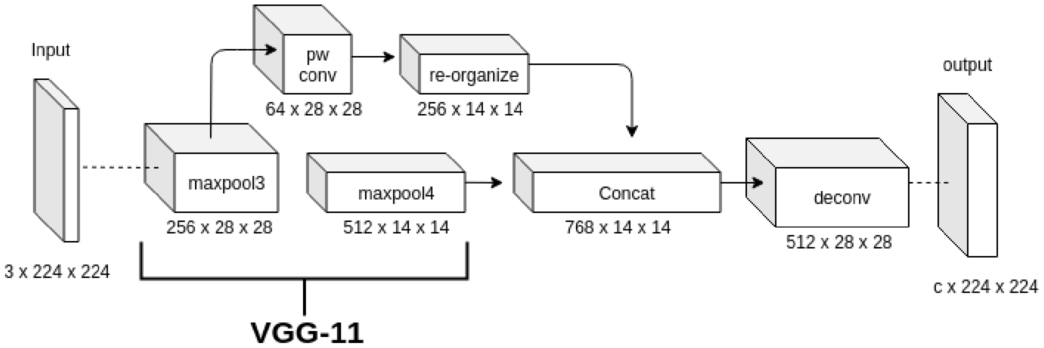

Figure 1: The FCN design for semantic segmentation, where c is the number of classes.

Since the backbone classification network is pre-trained on ImageNet, we re-size the input to 224×224 in addition to normalizing it with ImageNet mean and standard deviation. Following the approach in YOLOv2 [11], a passthrough layer is added between maxpool3 and maxpool4 of VGG-11 in order to benefit from finer grained features as illustrated in Figure 1. The passthrough layer concatenates the higher resolution features with the low resolution features by stacking adjacent features into different channels instead of spatial locations. However, in an attempt to reduce the computational costs as well as to prevent overfitting, I employed point-wise convolution as described in MobileNet [6] to bring down the dimensions of the feature maps after maxpool3.

Our FCN implementation is also inspired by the design in the original paper, where de-convolution layers are used to bi-linearly upsample the coarse $768 \times 14 \times 14$ feature maps to pixel-dense outputs. Similarly, the network outputs a binary mask with the same spatial size as the input for each of the pre-defined classes. For the first task of segmentation between background and person, only a single channel is used for binary pixel classification where 1 indicated person and 0 for background. To add non-linearity between layers, the ReLU activation function is placed after all convolution/deconvolution layers.

2.3 Training

The network is trained with ADAM optimizer [7], with an initial learning rate of 10^{-4} and weight decay of 10^{-5}. In addition, a scheduler was used to reduce the learning rate by a scale of 0.5 every 50 epochs. For the dataset of 600 photographs, we used the first 400 images (ordered by names) for training and the rest for testing. The only augmentation technique used in training was horizontal flipping the image with a probability of 0.5. The entire network is trained end-to-end with a simple binary cross-entropy loss $-y \log \sigma(x) - (1-y) \log \sigma(1-x)$ such that the target is down-sampled to have the same spatial dimension as the output. The model was trained on a laptop's GPU (NVIDIA Quadro M1200) with the process taking roughly 6 hours with a batch size of 8 images.

3 Experiments

We report the results for both tasks after 300 epochs with no-tuning of hyperparameters due to computational constraints.

3.1 Task 1: Binary Pixel Classification

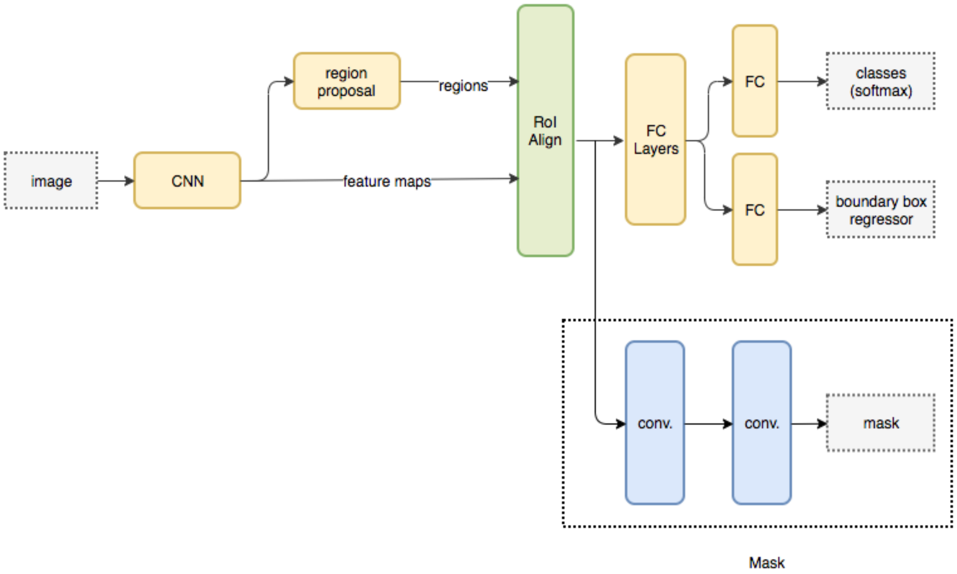

Figure 2: Summary of the Mask R-CNN design.

As mentioned in the previous section, a pre-trained Mask R-CNN model on MS COCO [9] is used to compare the results with our FCN. The original Github project was implemented by wannabeOG, and a forked repo (`https://github.com/raymondzmc/Mask-RCNN`) was created to save the network output for comparison against our method. During network inference, the class with the highest confidence with class ”person” is saved as a binary mask with the same size as the original input. The implementation of Mask R-CNN can be summarized as a multistage pipeline where regions of interests generated by a region proposal network is fed into a detector where object mask is outputted along with a refined bounding box and classification. The architecture of the network is illustrated in Figure 2, however, explicit details of the network and training procedure is out of the scope for this report.

Table 1: Accuracy comparison for task one

Accuracy	Mask R-CNN	FCN
person	95.75	96.51
background	86.05	88.59
pixel accuracy	96.64	97.25

The accuracy for each class is computed as the F1-score of $tp/(tp + fp + fn)$. Since our FCN takes in 224 × 224 sized input as required by the pre-trained classification network, the image is first down-sampled prior to feeding into the FCN and then up-sampled to the size of the original image, where both are performed with bilinear interpolation. Table 1 shows the accuracy between results of Mask R-CNN and FCN. As expected, our network performs better since it’s being trained on images from the same distribution. As seen from the visualizations in Figure 3, the results from our FCN looked aesthetically better than the results of Mask R-CNN. However, our FCN produced many enclaves of false-negative pixels within the regions of the true-positives. This is due to our network being trained on a pixel-level in contrast to using a connected region of interest.

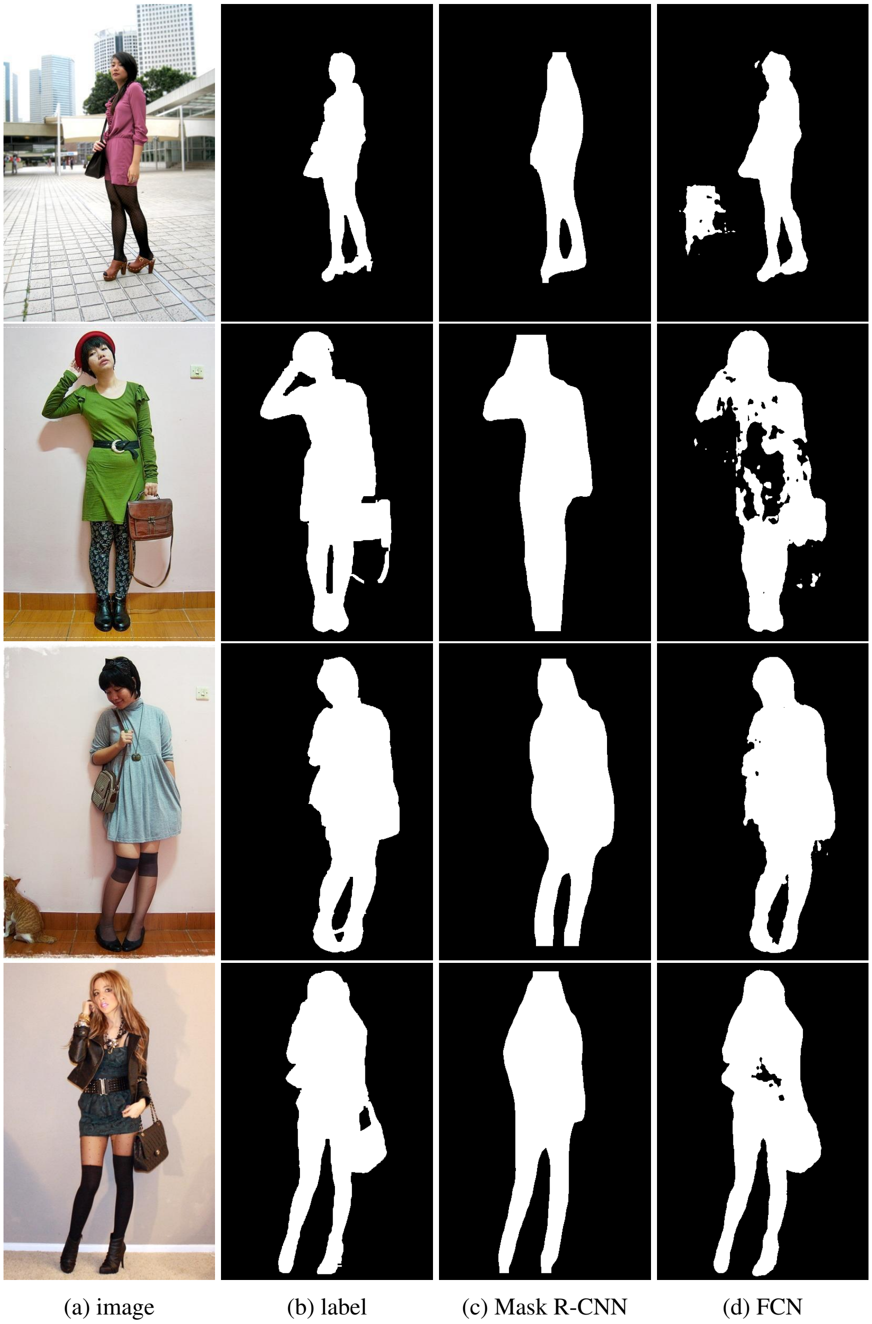
(a) image (b) label (c) Mask R-CNN (d) FCN

Figure 3: Visualization of output for task 1

3.2 Task 2: Multi-Class Pixel Classification

For the task of labeling 6 clothing types (including skin and hair) and background. The same scoring criteria is used to evaluate the accuracy of each class. From Table 2, we see that classes that appeared more frequent (background, hair, and skin) performed a lot better than less-occurring classes (tshirt, dress, shoes), as the network will do a better job of learning the features for a specific class when it appears more frequently in the training data. Although we used a much smaller dataset with less

Table 2: Accuracy for multi-class classification

Class	Accuracy
background	91.95
skin	64.92
hair	62.36
tshirt	22.07
shoes	39.36
pants	49.05
dress	34.61
pixel accuracy	90.40

predefined classes, this is still competitive with the 89% pixel accuracy with the method described in the original paper. We visualize the results for this task in Figure 4.

4 Conclusion

In this report, a novel implementation of [14] is proposed as a solution to the problem of parsing fashion photographs as described in [16]. Due to the computational constraints of training and evaluating deep convolution networks, many techniques to improve performance was not employed in this project. In particular, some additional techniques including the fine-tuning of the network hyperparameters, weighting the loss for class balancing, as well as pre-training the network on a large and generic dataset such as PASCAL VOC[] were left unexplored in this work. Last but not least, the code for this project is released on Github (`https://github.com/raymondzmc/csc420_project`), with the exception of the dataset as it belongs to the author of the original paper.

References

[1] *Dai Jifeng, He Kaiming, Li Yi, Ren Shaoqing, Sun Jian*. Instance-sensitive Fully Convolutional Networks // CoRR. 2016. abs/1603.08678.

[2] *Dai Jifeng, Li Yi, He Kaiming, Sun Jian*. R-FCN: Object Detection via Region-based Fully Convolutional Networks // NIPS. 2016.

[3] *Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L.* ImageNet: A Large-Scale Hierarchical Image Database // CVPR09. 2009.

[4] *He Kaiming, Gkioxari Georgia, Dollár Piotr, Girshick Ross*. Mask R-CNN // Proceedings of the International Conference on Computer Vision (ICCV). 2017.

[5] *He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian*. Deep Residual Learning for Image Recognition // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). June 2016.

[6] *Howard Andrew G., Zhu Menglong, Chen Bo, Kalenichenko Dmitry, Wang Weijun, Weyand Tobias, Andreetto Marco, Adam Hartwig*. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications // CoRR. 2017. abs/1704.04861.

[7] *Kingma Diederik P., Ba Jimmy*. Adam: A Method for Stochastic Optimization // CoRR. 2015. abs/1412.6980.

[8] *Lin T., Goyal P., Girshick R., He K., Dollr P.* Focal Loss for Dense Object Detection // 2017 IEEE International Conference on Computer Vision (ICCV). Oct 2017. 2999–3007.

[9] *Lin Tsung-Yi, Maire Michael, Belongie Serge J., Bourdev Lubomir D., Girshick Ross B., Hays James, Perona Pietro, Ramanan Deva, Dollár Piotr, Zitnick C. Lawrence*. Microsoft COCO: Common Objects in Context // CoRR. 2014. abs/1405.0312.

[10] *Paszke Adam, Gross Sam, Chintala Soumith, Chanan Gregory, Yang Edward, DeVito Zachary, Lin Zeming, Desmaison Alban, Antiga Luca, Lerer Adam*. Automatic differentiation in PyTorch // NIPS-W. 2017.

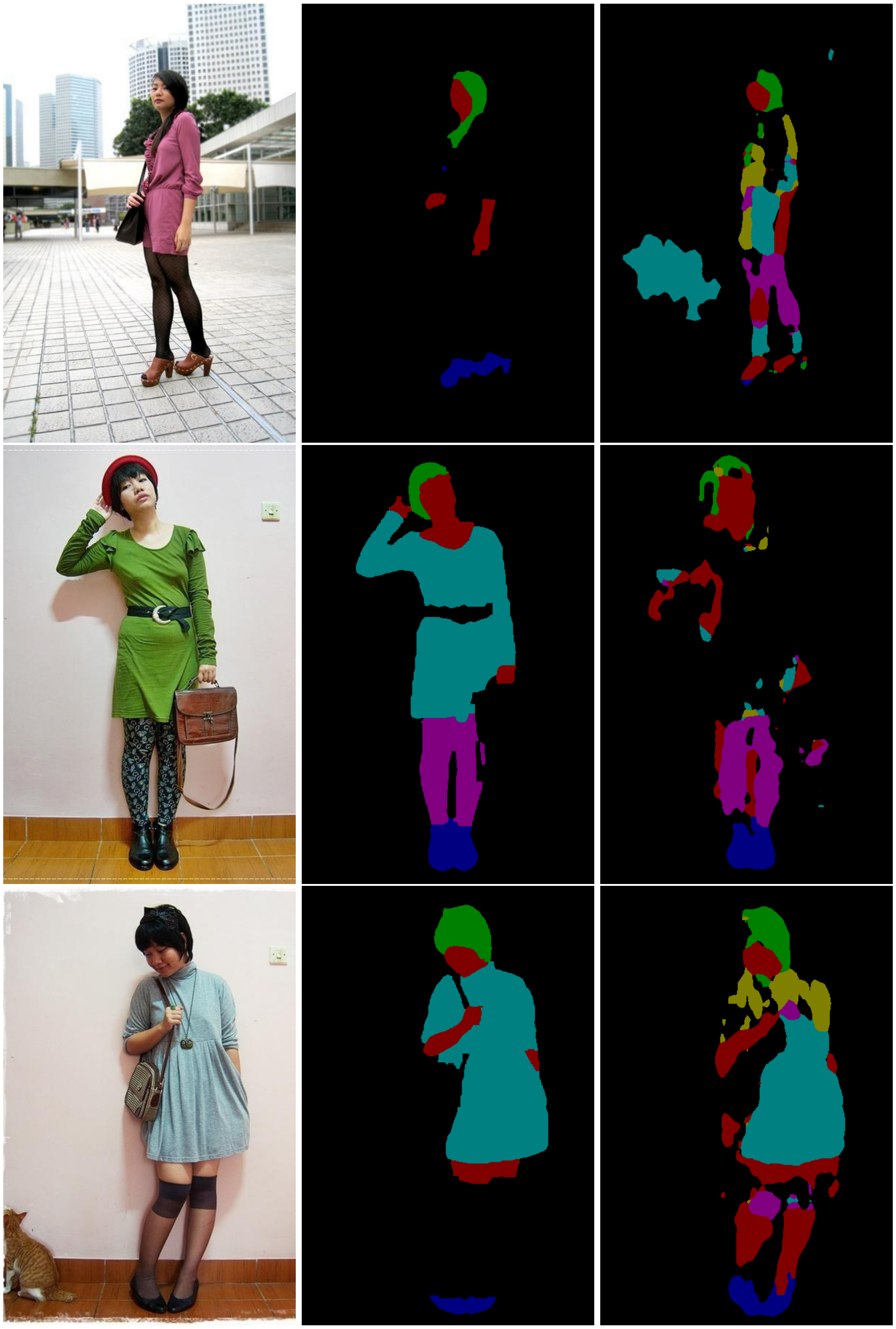

Figure 4: Visualization of output for task 2

[11] *Redmon Joseph, Farhadi Ali.* YOLO9000: Better, Faster, Stronger // CoRR. 2016. abs/1612.08242.

[12] *Ren Shaoqing, He Kaiming, Girshick Ross, Sun Jian.* Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // Advances in Neural Information Processing Systems (NIPS). 2015.

[13] *Shaban Amirreza, Bansal Shray, Liu Zhen, Essa Irfan, Boots Byron*. One-Shot Learning for Semantic Segmentation // CoRR. 2017. abs/1709.03410.

[14] *Shelhamer Evan, Long Jonathan, Darrell Trevor*. Fully Convolutional Networks for Semantic Segmentation // IEEE Trans. Pattern Anal. Mach. Intell. 2017. 39, 4. 640–651.

[15] *Simonyan Karen, Zisserman Andrew*. Very deep convolutional networks for large-scale image recognition // arXiv preprint arXiv:1409.1556. 2014.

[16] *Tangseng Pongsate, Wu Zhipeng, Yamaguchi Kota*. Looking at Outfit to Parse Clothing // CoRR. 2017. abs/1703.01386.

[17] *Zheng Shuai, Jayasumana Sadeep, Romera-Paredes Bernardino, Vineet Vibhav, Su Zhizhong, Du Dalong, Huang Chang, Torr Philip H. S.* Conditional Random Fields As Recurrent Neural Networks // Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). Washington, DC, USA: IEEE Computer Society, 2015. 1529–1537. (ICCV '15).