ModaNet: Multi-Label Classification on Street Fashion Dataset

Project report submitted for

VI Semester Minor Project-2

In

Department of CSE

By

Krutika Bapat (16100034)
Kushashwa Ravi Shrimali (16100035)
Saurabh Kumar Singh (16100051)

Department of CSE
Dr. Shyama Prasad Mukherjee
International Institute of Information Technology, Naya Raipur
(A Joint Initiative of Govt. of Chhattisgarh and NTPC)
Email: iiitnr@iiitnr.ac.in
Web: www.iiitnr.ac.in

CERTIFICATE

This is to certify that the project titled "ModaNet: Multi-Label Classification on Street Fashion Dataset" by "Krutika Bapat, Kushashwa Ravi Shrimali, Saurabh Kumar Singh" has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

(Signature of Guide)

_____

Dr Vivek Tiwari
Assistant Professor
Department of CSE
Dr. SPM IIIT-NR

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

PLAGIARISM REPORT

ModaNet: Multi-Label Classification on Street Fashion Dataset

ORIGINALITY REPORT

%9

SIMILARITY INDEX

EXCLUDE QUOTES	OFF
EXCLUDE BIBLIOGRAPHY	OFF

APPROVAL SHEET

This project report entitled "ModaNet: Multi-Label Classification on Street Fashion Dataset" by "Krutika Bapat, Kushashwa Ravi Shrimali, Saurabh Kumar Singh" is approved for VI Semester Minor Project.

_____

Dr. Muneendra Ojha (Chair)

_____

Dr. Vivek Tiwari

_____

Dr. Satyanarayana Vollala

Date: 13th May 2019

Place: IIIT Naya Raipur

ModaNet: Multi-Label Classification on Street Fashion Dataset

Abstract

ModaNet is a street fashion images dataset consisting of annotations related to RGB images. ModaNet provides multiple polygon annotations for each image. This dataset is described in a technical paper with the title ModaNet: A Large-Scale Street Fashion Dataset with Polygon Annotations. Each polygon is associated with a label from 13 meta fashion categories. The annotations are based on images in the Paper Doll image set, which has only a few hundred images annotated by the superpixel-based tool. The contribution of ModaNet is to provide new and extra polygon annotations for the images.

Section 1: Introduction

Understanding clothes from a single image has strong commercial and cultural impacts on modern societies. However, this task remains a challenging computer vision problem due to wide variations in the appearance, style, brand and layering of clothing items. We use a new database called ModaNet, a large-scale collection of images based on Paperdoll dataset. The dataset provides 55,176 street images, fully annotated with polygons on top of the 1 million weakly annotated street images in Paperdoll. ModaNet aims to provide a technical benchmark to fairly evaluate the progress of applying the latest computer vision techniques that rely on large data for fashion understanding. The rich annotation of the dataset allows to measure the performance of state-of-the-art algorithms for object detection, semantic segmentation and polygon prediction on street fashion images in detail.

Fashion understanding and analysis have been a popular topic amongst the Computer Vision community beginning from the popular revised version of MNIST to Fashion MNIST. Popular stars and personalities tend to wear branded and trending clothings to attract attention of the public and fashion brands. But it still remains a challenge to identify what type of wear it is and what brand it is. There is no commercial tool to allow a person to analyze and understand a wear of a star or a model. The aim is to use information available from the dataset to successfully classify each fashion item amongst the given 13 classes of items (see Fig. 1). The workflow is, given a street fashion photo, the algorithm identifies each item and segments it from the image which is then assigned a class from the given 13 classes in the dataset. This attracts an application in commercial markets as well. Along with identification and classification of each fashion item, localization and segmentation of an item is also an important task. This can be

further extended to personalized makeover of a model and allowing to explore more styles based on previous successful models and fashion designers. Localization is an important objective in the sense that a person might just be interested in a specific fashion item and not the whole image. Thinking of a scenario where a person is only interested in the type of shoes a model is wearing, and it makes no sense to pass whole image to any visual search engine. Thus, it will be a good idea to localize that item from the image, segment it and only pass it to the search engine. This enables quicker and accurate results.

The reason ModaNet works well in the case of our objective is that it provides different poses of each model with the same fashion wears, enabling real time use. While there are other datasets available, but with less number of images and only single pose for each model with a wear, thus making very difficult to be used in case of real time applications. ModaNet being 10X larger than other fashion datasets, along with different poses and with enough diversity in the images, is thus the best choice for our objectives.

Label	Description	Fine-Grained-categories
1.	bag	bag
2.	belt	belt
3.	boots	boots
4.	bootwear	footwear
5.	outer	coat/jacket/suit/blazers/cardigan/sweater/Jumpsuits/Rompers/vest
6.	dress	dress/t-shirt dress
7.	sunglasses	sunglasses
8.	pants	pants/jeans/leggings
9.	top	top/blouse/t-shirt/shirt
10.	shorts	shorts
11.	skirt	skirt
12.	headwear	headwear
13.	Scarf & tie	Scarf & tie

Fig. 1: ModaNet Statistics: Grouped 13 meta-categories

Section 2: Literature Review

Section 2.1: Detection

There are various applications based on existing datasets that try to understanding fashion from different perspectives. One important task is to detect fashion items from images. Recently, various approaches based on deep neural networks have been proposed for generic object detection and achieved promising results, among which, some representative works are RCNN [12], Fast RCNN [11], Faster RCNN [32], SSD [28], R-FCN [7], YOLO [31], etc. To improve the performance of these detectors, several modifications have been proposed. Shrivastava et al. [36] propose to use online hard negative mining to adaptively select diverse, high-loss samples for training. Lin et al. [24] propose the focal loss as an alternative way to do hard negative mining to alleviate the effect of overwhelming negative samples. Bodla et al. [2] introduce a soft-NMS to replace the traditional non-maximal suppression (NMS) used in object detection to discount the confidence score of predicted boxes rather than completely discarding them. By changing the backbone networks used in the detector, feature pyramid network [23], also called RetinaNet, has improved the detection accuracy, where feature maps from different convolutional layers are fused to provide more discriminative power. Deformable convolutional networks [8] also greatly improves detection performance for non-rigid objects by allowing convolution to operate on irregular regions instead of a grid. Regarding fashion item detection specifically, Hara et al. [13] incorporate contextual information from body poses to guide detection and extract features from an off-the-shelf deep neural network. Liu et al. [30] present preliminary results using Fast RCNN trained on the DeepFashion dataset, but the model can only detect upper-body, lower-body and full-body objects due to the lack of fine-grained annotations.

Section 2.2: Segmentation

The main focus of semantic image segmentation is to assign an object label to each pixel in an image. Recently, researchers have also begun tackling new problems such as instance segmentation [14], which aims to assign a unique identifier and a semantic meaningful label to each segmented object in the image. Semantic segmentation has traditionally been approached using probabilistic models known as Conditional Random Fields (CRFs) [16], which explicitly model the correlations among the pixels being predicted. However, in the recent years, learning a better feature representation [19, 35] has shown to play an important role in pushing the state-of-the-art performance of semantic image segmentation. The rise of deep Convolutional Neural Networks has significant improved the way of learning feature representation. In particularly, the fully convolutional networks (FCNs) have shown significantly performance boost in semantic image segmentation [34]. There are two directions of improving semantic

image segmentation. One is to improve the architectures or the bottleneck module of neural networks. The representative work is Chen et al. [4], which has further developed the FCNs using atrous layers. It has also used densely-connected CRFs as a post-processing step. Yu et al. [44] has improved semantic image segmentation by introducing dilated convolution, which increases the resolution of output feature maps without reducing the receptive field of individual neurons. Zhao [45] has proposed a pyramid scene parsing network that explores the prior global representation to produce good results on scene parsing task. Chen [5] has developed modules that employ the atrous convolution in cascade or in parallel to capture multi-scale context by adopting multiple atrous rates, which has shown state-of-the-art performance in PASCAL VOC benchmark. The other direction is to incorporate CRFs into an end-to-end trainable framework for semantic image segmentation, with the hope that joint training would help improve performance further. Zheng et al. [46] formulate an end-to-end trainable framework using the fully convolutional neural networks and densely-connected CRFs, while Liu et al. [29], Schwing et al. [33] and Lin et al. [22] have explored similar ideas along this line. These approaches are developed for generic object categories, and some are developed for scene parsing. Different from generic objects or scenes, there are more self-occlusion in fashion images. One goal of the ModaNet is to facilitate future semantic image segmentation that would perform well in fashion.

Several works have also explored semantic image segmentation in the context of fashion. Yamaguchi et al. [41] has pioneered the work in clothing parsing, and later further improved the fashion parsing performance by using retrieval-based approach [40]. Dong et al. [10] use Parselets as the building blocks of the parsing model. Liu et al. [27] present a solution that harnesses the context in fashion videos to boost the fashion parsing performance. Liang et al. [21] developed a convolutional neural network approach for human parsing. These works can be categorized into human parsing and clothing parsing. Clothing parsing attempts to identify the fine-grained categories of clothing items such as t-shirt and blouse, whereas human parsing tries to identify the body parts and broad clothing categories. The ModaNet dataset focuses on the clothing fashion parsing rather than human parsing.

Section 3: Methodology

Section 3.1: Problem Statement

Based on the existing enhancements and work in the field of Fashion Labelling, there are below listed challenges:

1. The existing segmentation models for Fashion Data set don’t work for multiple persons in an image.

2. There is no comparison of existing Segmentation Techniques using ModaNet.
3. Existence of noise is obvious for any commercial tool to be used in a real time application. Current tools don't work well with images having noise.

We show and discuss evidences of this in Section 2.4 of Result. The ModaNet data set is quite diverse as it is evident in Fig. 3. The problem of multi label classification is evidently useful for both commercial and educational purposes. We list down the reasons of why this application is so important.

- Understanding clothes from a single image would have huge commercial and cultural impacts on modern societies. However, this task remains a challenging computer vision problem due to wide variations in the appearance, style, brand, and layering of clothing items.
- This dataset provides 55,176 street images, fully annotated with polygons on top of the 1 million weakly annotated street images in Paperdoll.
- The main focus of semantic image segmentation is to assign an object label to each pixel in an image. Recently, researchers have also begun tackling new problems such as instance segmentation, which aims to assign a unique identifier and a semantic meaningful label to each segmented object in the image.

The applications of multi label classification for commercial tools and platforms is quite evident in the case of visual search engines. We use localization techniques to make the search faster, where if the person is interested in only a specific item then only the segmented image of the localized item should be passed to the visual search engine which makes the whole process faster and accurate. In this way, the search engine has to just work with a part of an image. The applications of ModaNet lie in the hands of researchers to test and analyze their proposed algorithms for Object Detection, Segmentation, Localization, Classification and many more. The problem is clearly identified in Fig. 4. While the glimpses of data are given in Fig. 2 and Fig. 3.

Object Detection on ModaNet

The dataset enables fashion item detection, where each fashion item is localized and assigned with a category label, which can be further used for visual search and product recommendation. The first step is to generate ground truth based on polygon annotations. Once this is done, there are N number of detectors available which can be used and compared based on their precision-recall. Some good performing detectores include RCNN,

SSD and YOLO. YOLO and SSD are both real time detectors and can be used in mobile applications. This is because of their similar architectures.

Meta	Raw	#Train	#Val	Avg Inst. size
bag	bag	36, 699	2, 155	4.88%
belt	belt	13, 743	771	0.46%
boots	boots	7, 068	691	2.40%
footwear	footwear	39, 364	1, 617	0.96%
outer	coat, jacket, suit, blazers	23, 743	1, 358	7.48%
dress	dress, t-shirt dress	14, 460	804	10.49%
sunglasses	sunglasses	8, 780	524	0.31%
pants	pants, jeans, leggings	23, 075	1, 172	5.65%
top	top, blouse, t-shirt, shirt	34, 745	1, 862	4.83%
shorts	shorts	5, 775	429	2.86%
skirt	skirt	10, 860	555	6.40%
headwear	headwear	5, 405	491	1.25%
scarf&tie	scarf, tie	3, 990	378	2.55%

Fig. 2: ModaNet Detailed Statistics: Number of Training images and Validation images along with meta-categories distribution.

Fig. 3: Examples of original images in the ModaNet dataset along with pixel-wise segmentation masks and bounding box annotations. Top row displays original images of street fashion dresses, with diverse poses. The middle row displays segmentation masks on the original images, while the bottom row displays bounding box annotations on each original image.

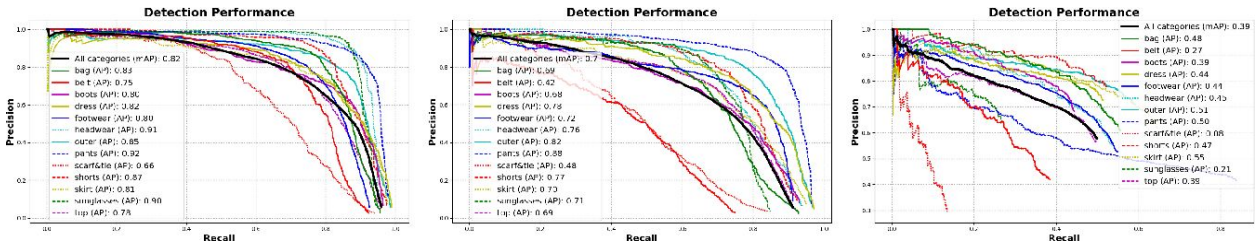

Fig. 4: Performance comparison of Object Detection on ModaNet using (left to right) Faster RCNN, SSD and YOLO.

Semantic Segmentation on ModaNet

ModaNet dataset provides pixel-wise annotation to enable semantic segmentation research for fashion. Again, the first step is to generate ground truths based on the pixel-wise annotations. Semantic Segmentation also enables to identify colors and this might come handy as shown in Fig. 5.

Recent approaches such as PolygonRNN and Polygon-RNN++ have addressed the problem of polygon prediction directly using neural networks. We set up the baseline performance on ModaNet using the pre-trained model of PolygonRNN++. This model is an encoder-decoder network. The encoder produces image features that are used to predict the first vertex, and then the first vertex and the image features are fed to the recurrent decoder. The recurrent neural network exploits the visual attention at each time step to produce polygon vertices. A learned evaluator is employed to select the best polygon from a set of candidates proposed by the decoder. In the final stage, a graphbased neural network re-adjusts the polygons and augments them with additional vertices at a higher resolution. The base model in the encoder is a modified ResNet-50, which has reduced stride and dilation factors.

In our experiment, we adapt the public available Polygon-RNN++ model that is pre-trained on the Cityscape dataset to produce the baseline performance. We form the inputs of Polygon-RNN++ using the cropped images based on the FasterRCNN detection results. The polygon predictions of polygon-RNN++ are evaluated on the validation set of ModaNet dataset. We convert the polygon predictions to the mask-like predictions. A perfect polygon prediction should give the same semantic image segmentation mask created by human annotators. Using the masks for semantic segmentation as groundtruth, we evaluate how well the results from the pre-trained polygon-RNN++ model align well with the segmentation masks. The Polygon-RNN++ model achieves mean IOU 30.7%, mean precision 83.4%, mean recall 32.5%

and mean F-1 score 45.0%. We hope such preliminary baseline results on the ModaNet dataset would motivate future research on polygon prediction.

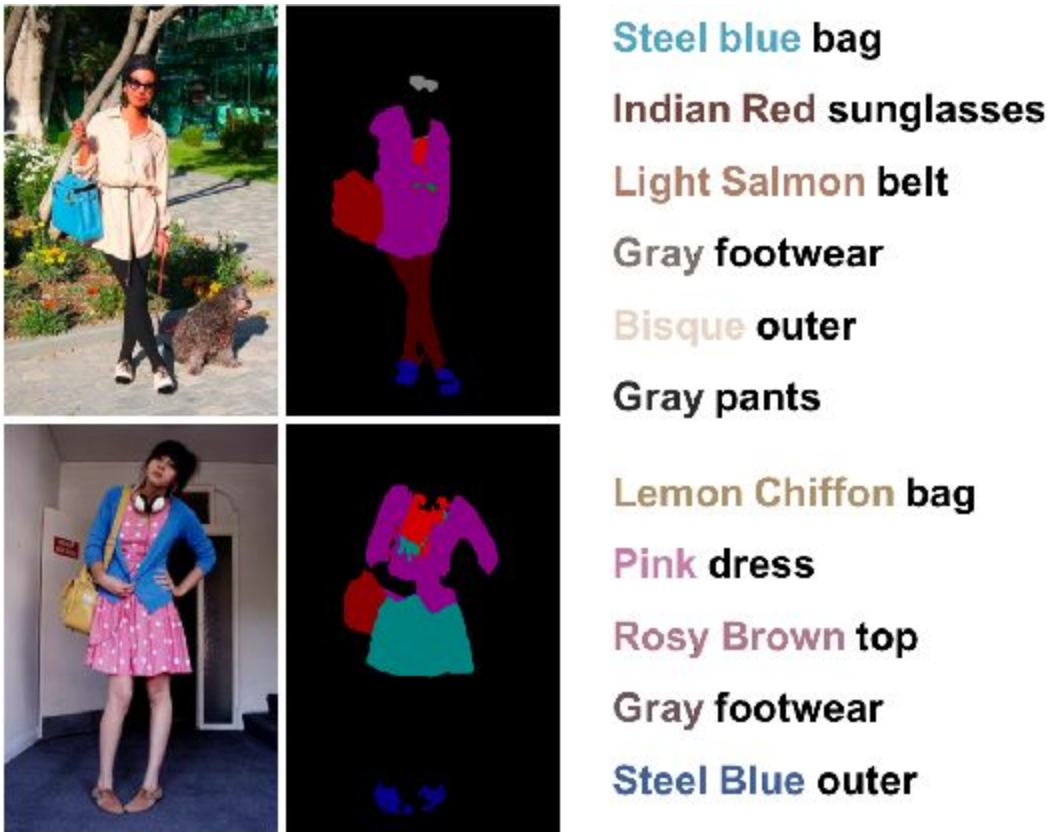

Fig. 5: Color Attribute Prediction and Segmentation combined on ModaNet.

Section 2.2: Proposed work

We analyze the problems one by one and give solutions to the same below.

1. Problem with multiple persons in the image. Solution:
    a. Use deep object detection technique (YOLOv3) to recognize each person in the image.
    b. Pass each person to the model, use parallel-processing.
    c. Over-write result on the original image.
2. Comparison with Classifical Segmentation Techniques like Watershed with Deep Learning Techniques. Solution:
    a. Introduce SegNet architecture on the ModaNet Dataset for higher accuracy.
    b. Automatic Background Removal/Foreground Extraction using OpenCV

We propose a network on the top of SegNet which has an encoder network and a corresponding decoder network, followed by a final pixel wise classification layer. This architecture is illustrated in Fig. 6. The encoder network consists of 13 convolutional layers which correspond to the first 13 convolutional layers in the VGG16 network designed for object classification. We can therefore initialize the training process from weights trained for classification on large datasets. We can also discard the fully connected layers in favour of retaining higher resolution feature maps at the deepest encoder output. This also reduces the number of parameters in the SegNet encoder network significantly (from 134M to 14.7M) as compared to other recent architectures. Each encoder layer has a corresponding decoder layer and hence the decoder network has 13 layers. The final decoder output is fed to a multiclass softmax classifier to produce class probabilities for each pixel independently.

Each encoder in the encoder network performs convolution with a filter bank to produce a set of feature maps. These are then batch normalized). Then an element-wise rectified linear non-linearity (ReLU) max(0, x) is applied. Following that, max-pooling with a 2 × 2 window and stride 2 (non-overlapping window) is performed and the resulting output is sub-sampled by a factor of 2. Max-pooling is used to achieve translation invariance over small spatial shifts in the input image. Sub-sampling results in a large input image context (spatial window) for each pixel in the feature map. While several layers of max-pooling and subsampling can achieve more translation invariance for robust classification correspondingly there is a loss of spatial resolution of the feature maps. The increasingly lossy (boundary detail) image representation is not beneficial for segmentation where boundary delineation is vital. Therefore, it is necessary to capture and store boundary information in the encoder feature maps before sub-sampling is performed. If memory during inference is not constrained, then all the encoder feature maps (after subsampling) can be stored. This is usually not the case in practical applications and hence we propose a more efficient way to store this information. It involves storing only the max-pooling indices, i.e, the locations of the maximum feature value in each pooling window is memorized for each encoder feature map. In principle, this can be done using 2 bits for each 2 × 2 pooling window and is thus much more efficient to store as compared to memorizing feature map(s) in float precision. As we show later in this work, 5 this lower memory storage results in a slight loss of accuracy but is still suitable for practical applications. The appropriate decoder in the decoder network upsamples its input feature map(s) using the memorized max-pooling indices from the corresponding encoder feature map(s). This step produces sparse feature map(s).

This SegNet decoding technique is illustrated in Fig. 3. These feature maps are then convolved with a trainable decoder filter bank to produce dense feature maps. A batch normalization step is then applied to each of these maps. Note that the decoder corresponding to the first encoder (closest to the input image) produces a multi-channel feature map, although its encoder input has 3 channels (RGB). This is unlike the other decoders in the network which produce feature maps

with the same number of size and channels as their encoder inputs. The high dimensional feature representation at the output of the final decoder is fed to a trainable softmax classifier. This soft-max classifies each pixel independently. The output of the softmax classifier is a K channel image of probabilities where K is the number of classes. The predicted segmentation corresponds to the class with maximum probability at each pixel. We add here that two other architectures, DeconvNet and U-Net share a similar architecture to SegNet but with some differences. DeconvNet has a much larger parameterization, needs more computational resources and is harder to train end-to-end, primarily due to the use of fully connected layers (albeit in a convolutional manner) We report several comparisons with DeconvNet later in the paper Sec. 4. As compared to SegNet, U-Net (proposed for the medical imaging community) does not reuse pooling indices but instead transfers the entire feature map (at the cost of more memory) to the corresponding decoders and concatenates them to upsampled (via deconvolution) decoder feature maps. There is no conv5 and max-pool 5 block in U-Net as in the VGG net architecture. SegNet, on the other hand, uses all of the pre-trained convolutional layer weights from VGG net as pre-trained weights.

The major challenge is to perform this task on an image with multiple persons and background noise. While most of the deep learning techniques perform data augmentation to increase data for training, the type of transformations needed for real time use are not considered. We try to overcome the problems in existing networks using background removal using Grabcut Segmentation and using YOLOv3 to detect persons and performing classification using Parallel Processing. The whole aim is to make sure the process is fast enough for commercial usage.

Section 2.3: Architectures

We use SegNet architecture to perform Segmentation Task on ModaNet. The architecture is described in Fig. 6 and 8. Since SegNet is on the top of FCN, thus we show the architecture of FCN in Fig. 7. For real time application, we use Grabcut Segmentation to perform background removal and localization as shown in Fig. 9. In general the flow process goes like this:

1. Ground Truth Generation from the ModaNet dataset using Polygon Annotations, labels and bounding boxes.
2. Training data along with ground truth labels using given architecture (depending on application).

Ground truth generation depends on the application and for our application, we have to use polygon annotations to produce segmentation labels.

GrabCut Segmentation in Python

```
# select ROI
rect = cv.selectROI(img)

 # in case no rectangle selected
 if rect==(0,0,0,0):
    print("Exiting...")
    break

    # initialize background and foreground masks
 bgdmodel = np.zeros((1, 65), np.float64)
 fgdmodel = np.zeros((1, 65), np.float64)

 # perform grabcut segmentation
 cv.grabCut(img, mask, rect, bgdmodel, fgdmodel, 5, cv.GC_INIT_WITH_RECT)

 mask = np.where((mask == 1) + (mask == 3), 255, 0).astype('uint8')
 redImg = np.zeros(img.shape, img.dtype)
 redImg[:,:] = (np.random.randint(0, 255), np.random.randint(0, 255), np.random.randint(0, 255))
 redMask = cv.bitwise_and(redImg, redImg, mask=mask)

 output = cv.addWeighted(redMask, 1, img, 1, 0, 0)
```

GrabCut Segmentation in C++

```
Rect2d rect = selectROI(img2);
if(rect == Rect2d(0, 0, 0, 0)) {
    // no rectangle selected
    break;
}
// cv::Mat mask(img.size(),CV_8UC3,cv::Scalar(255,255,255));
Mat result(img.size(),CV_8UC1);
Mat fg,bg;

grabCut(img,result,rect,bg,fg,2,GC_INIT_WITH_RECT);
compare(result,cv::GC_PR_FGD,result,cv::CMP_EQ);

 Mat redImg(img.size(), CV_8UC3, Scalar(int(rng.uniform(0, 255)), int(rng.uniform(0, 255)),
int(rng.uniform(0, 255))));

 Mat redMask;
 bitwise_and(redImg, redImg, redMask, result);
 Mat output;
```

```
addWeighted(redMask, 1.0, img2, 1.0, 0.0, output);
img2 = output;
```

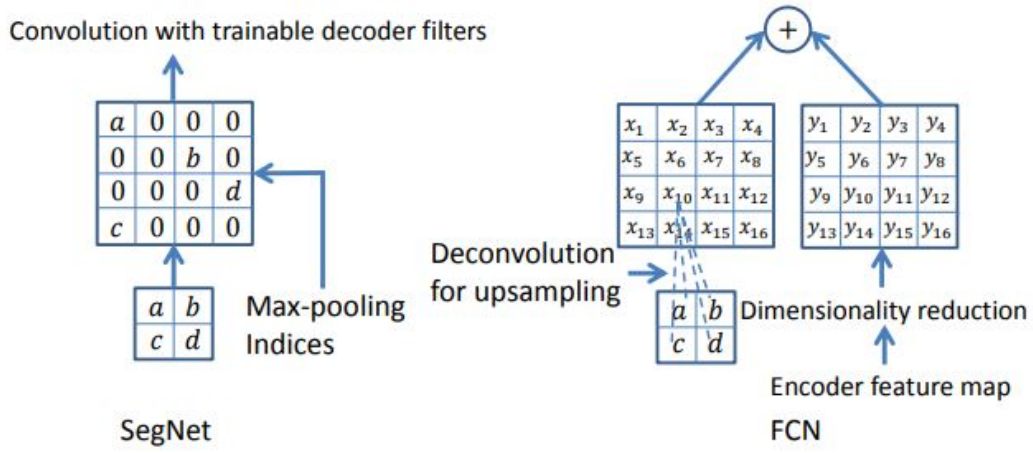

Fig. 6: (Left to Right): SegNet Decoder Illustration and FCN Decoder Illustration. (a, b, c, d) are values in a feature map. SegNet uses the max pooling indices to upsample (without learning) the feature map(s) and convolves with a trainable decoder filter bank. FCN upsamples by learning to deconvolve the input feature map and adds the corresponding encoder feature map to produce the decoder output. This feature map is the output of the max-pooling layer (includes sub-sampling) in the corresponding encoder. Note that there are no trainable decoder filters in FCN.

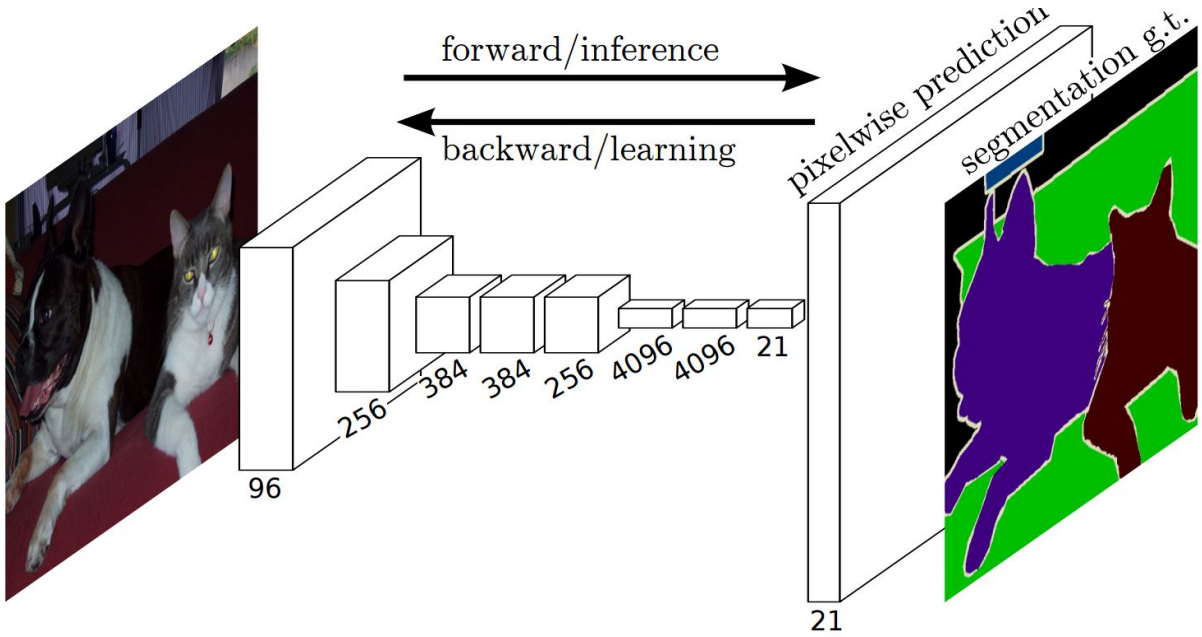

Fig. 7: FCN Architecture for Segmentation. Reference: http://deeplearning.net/tutorial/fcn_2D_segm.html

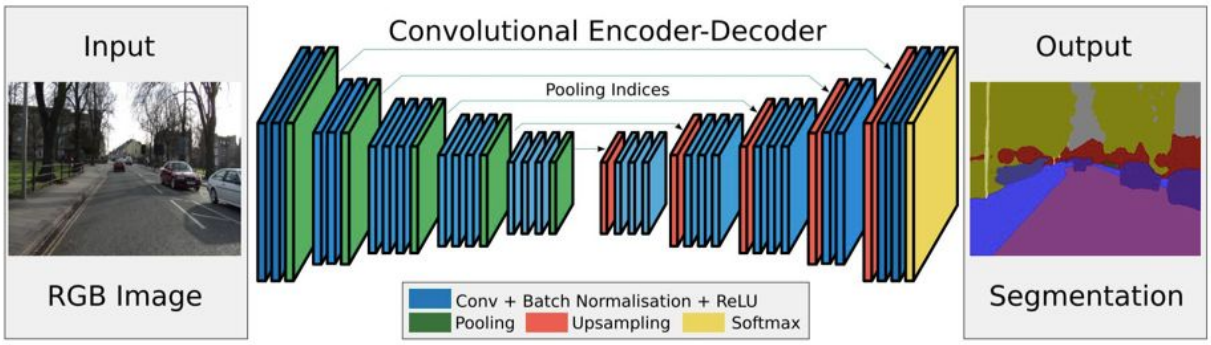

Fig. 8: SegNet Architecture for Segmentation

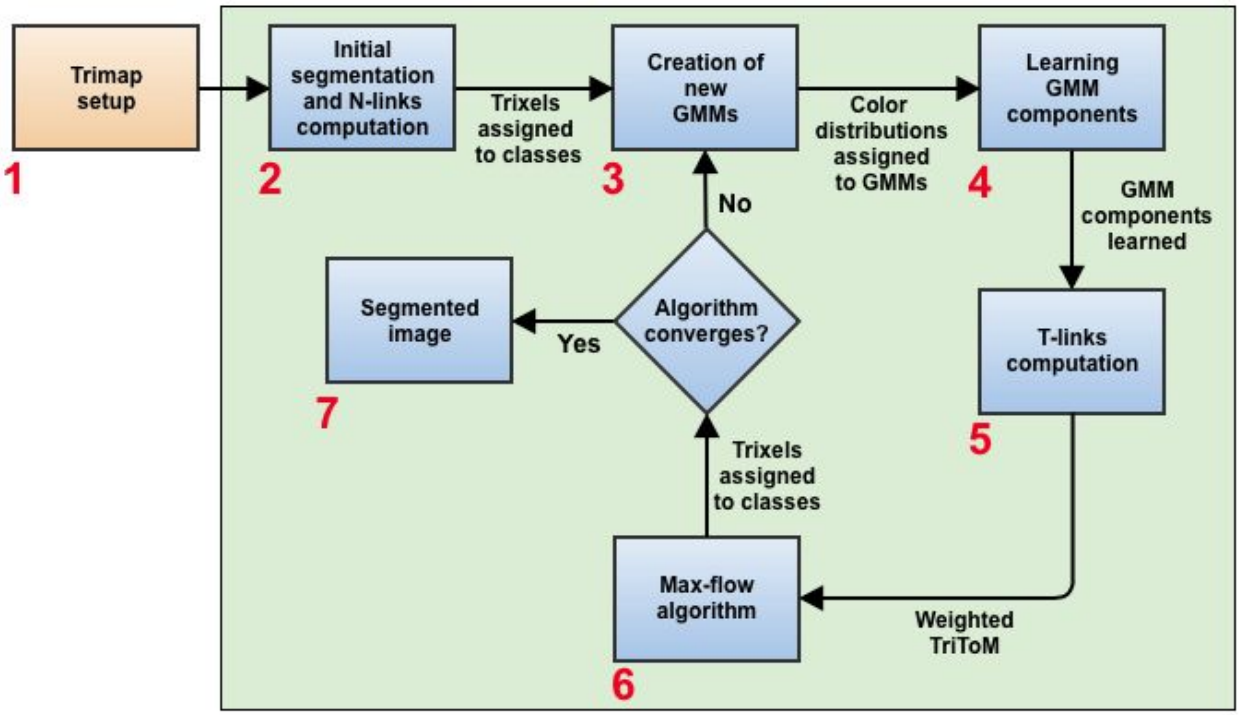

Fig. 9: Flow Chart of the whole process of Grabcut Segmentation

Section 2.3: Evaluation

There are commonly three evaluation metrics used for evaluation of classification algorithms like here. Below are the mentioned metrics:

1. F1-Score
2. Precision and Recall
3. IoU Per Category

We give brief introduction to the common metrics used.

- Precision: It is the number of correct positive results divided by the number of positive results predicted by the classifier. Mathematically, it can be expressed as:
    - $Precision = True\ Positives\ /\ (True\ Positives + False\ Positives)$
- Recall: It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). Mathematically, it can be expressed as:
    - $Recall = True\ Positives\ /\ (True\ Positives + False\ Negatives)$
- F1-Score is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1].
    - It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).
    - High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify.
    - The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as :
    - $F1 - Score = 2 * \frac{1}{1/precision + 1/recall}$
    - F1 Score tries to find the balance between precision and recall.

There have been lots of studies on statistic based models, and we discuss their importance and significance in the application of Classification algorithms.

1. Z-Score
2. Binomial Test
3. Omnibus Test
4. F-Test

Z-Score: Also known as Standard Deviation (SD) score is the measure of dispersion/relative deviance of the data from the mean/median value i.e. measure of the distance between the child's value and value of the reference population.

Fig. Classification Table of Z-Score

Binomial Test: It is an exact test of the statistical significance of deviations from a theoretically expected distribution of observations into two categories. One common use of the binomial test is in the case where the null hypothesis is that two categories are equally likely to occur (such as a coin toss). Tables are widely available to give the significance observed numbers of observations in the categories for this case. However, as the example below shows, the binomial test is not restricted to this case. Where there are more than two categories, and an exact test is

Measurements	Classifications
Weight for Age	
< -2 Z score	Underweight
< -2 to -3 Z score	Moderate Underweight
< -3 Z score	Severe Underweight
Height for Age	
< -2 Z score	Stunting
< -2 to -3 Z score	Moderate Stunting
< -3 Z score	Severe Stunting
Weight for Height	
< -2 Z score	Wasting (known as Global Acute Malnutrition)
< -2 to -3 Z score	Moderate Wasting
< -3 Z score	Severe Wasting

required, the multinomial test, based on the multinomial distribution, must be used instead of the binomial test.

Omnibus Test: Omnibus tests are a kind of statistical test. They test whether the explained variance in a set of data is significantly greater than the unexplained variance, overall. One example is the F-test in the analysis of variance. For instance, in a model with two independent variables, if only one variable exerts a significant effect on the dependent variable and the other does not, then the omnibus test may be non-significant. This fact does not affect the conclusions that may be drawn from the one significant variable. In order to test effects within an omnibus test, researchers often use contrasts.

F-Test: An F-test is any statistical test in which the test statistic has an F-distribution under the null hypothesis. It is most often used when comparing statistical models that have been fitted to a data set, in order to identify the model that best fits the population from which the data were sampled. Exact "F-tests" mainly arise when the models have been fitted to the data using least squares.

Section 3.5: Results

One application of semantic segmentation is to predict the color attribute name given a fashion product. We develop such a prototype based on ModaNet dataset. We first conduct semantic segmentation, and then predict the color attribute names by mapping the mean RGB values for each segment to a fine-grained color name space.

Method	bg	bag	belt	boots	footwear	outer	dress	sunglasses	pants	top	shorts	skirts	headwear	scarf&tie
FCN-32 [34]	0.95	0.27	0.12	0.32	0.33	0.36	0.28	0.25	0.51	0.38	0.40	0.28	0.33	0.17
FCN-16 [34]	0.96	0.26	0.19	0.32	0.38	0.35	0.25	0.37	0.51	0.38	0.40	0.23	0.41	0.16
FCN-8 [34]	0.96	0.24	0.21	0.32	0.40	0.35	0.28	0.41	0.51	0.38	0.40	0.24	0.44	0.18
FCN-8satonce [34]	0.96	0.26	0.20	0.31	0.40	0.35	0.29	0.36	0.50	0.39	0.38	0.26	0.44	0.16
CRFasRNN [46]	0.96	0.30	0.18	0.41	0.39	0.43	0.32	0.36	0.56	0.40	0.44	0.26	0.45	0.22
DeepLabV3+ [5]	0.98	0.42	0.28	0.40	0.51	0.56	0.52	0.46	0.68	0.55	0.53	0.41	0.55	0.31

Table 1: IoU per category of evaluated semantic segmentation approaches

Method	bg	bag	belt	boots	footwear	outer	dress	sunglasses	pants	top	shorts	skirts	headwear	scarf&tie
FCN-32 [34]	0.97	0.52	0.37	0.59	0.51	0.53	0.44	0.53	0.60	0.53	0.56	0.45	0.47	0.46
FCN-16 [34]	0.97	0.52	0.42	0.64	0.58	0.47	0.38	0.66	0.64	0.59	0.58	0.35	0.65	0.36
FCN-8 [34]	0.97	0.42	0.51	0.66	0.58	0.52	0.46	0.74	0.61	0.49	0.62	0.45	0.74	0.53
FCN-8satonce [34]	0.97	0.51	0.43	0.75	0.59	0.52	0.45	0.74	0.59	0.52	0.51	0.44	0.72	0.55
CRFasRNN [46]	0.96	0.49	0.57	0.66	0.68	0.58	0.58	0.65	0.76	0.61	0.61	0.37	0.73	0.44
DeepLabV3+ [5]	0.99	0.62	0.53	0.75	0.62	0.70	0.67	0.74	0.75	0.69	0.69	0.56	0.74	0.51

Table 2: Precision per category of evaluated semantic segmentation approaches

Method	bg	bag	belt	boots	footwear	outer	dress	sunglasses	pants	top	shorts	skirts	headwear	scarf&tie
FCN-32 [34]	0.98	0.43	0.23	0.48	0.53	0.62	0.60	0.44	0.81	0.63	0.66	0.53	0.67	0.31
FCN-16 [34]	0.98	0.42	0.37	0.47	0.60	0.69	0.61	0.54	0.78	0.57	0.64	0.59	0.62	0.36
FCN-8 [34]	0.99	0.47	0.37	0.44	0.64	0.63	0.57	0.53	0.81	0.71	0.61	0.47	0.57	0.28
FCN-8satonce [34]	0.99	0.42	0.38	0.39	0.63	0.63	0.58	0.48	0.83	0.68	0.69	0.51	0.59	0.24
CRFasRNN [46]	0.99	0.53	0.26	0.59	0.52	0.70	0.50	0.54	0.71	0.58	0.66	0.62	0.60	0.41
DeepLabV3+ [5]	0.99	0.61	0.48	0.50	0.78	0.78	0.73	0.60	0.88	0.74	0.74	0.69	0.72	0.54

Table 3: Recall per category of evaluated semantic segmentation approaches.

Method	bg	bag	belt	boots	footwear	outer	dress	sunglasses	pants	top	shorts	skirts	headwear	scarf&tie
FCN-32 [34]	0.98	0.58	0.43	0.67	0.58	0.67	0.70	0.53	0.81	0.67	0.79	0.74	0.65	0.50
FCN-16 [34]	0.98	0.57	0.50	0.65	0.66	0.69	0.68	0.65	0.82	0.66	0.79	0.72	0.71	0.52
FCN-8 [34]	0.98	0.56	0.52	0.65	0.69	0.67	0.68	0.70	0.83	0.67	0.78	0.67	0.71	0.51
FCN-8satonce [34]	0.98	0.57	0.52	0.65	0.69	0.68	0.69	0.65	0.84	0.68	0.80	0.70	0.72	0.47
CRFasRNN [46]	0.98	0.62	0.48	0.74	0.63	0.74	0.68	0.63	0.83	0.69	0.82	0.77	0.73	0.60
DeepLabV3+ [5]	0.99	0.72	0.59	0.72	0.78	0.82	0.84	0.74	0.90	0.78	0.85	0.85	0.81	0.70

Table 4: F-1 score per category of evaluated semantic segmentation approaches

The image and the predicted masks are shown on the left, and the predicted text including the color attribute names and the fashion object category names is shown on the right.

Final output

Fig. 10: Output of FCN Pretrained Models on single persons image (Real Time)

What's wrong?

- The current work on Paperdoll Dataset doesn't work well with Multiple Persons.

False Labels:

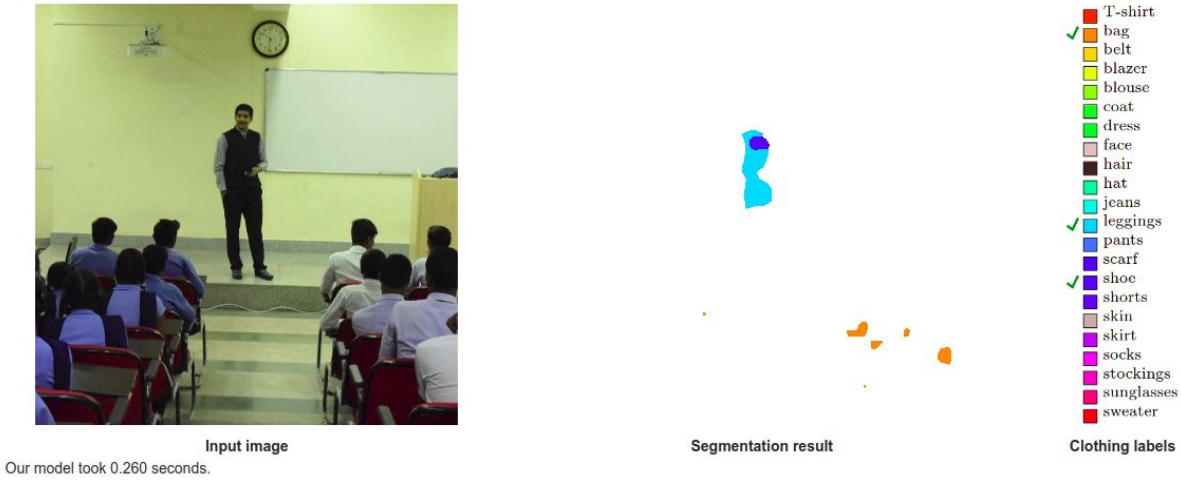

Fig. 11: Output of Paperdoll Pretrained Models on Multiple Persons

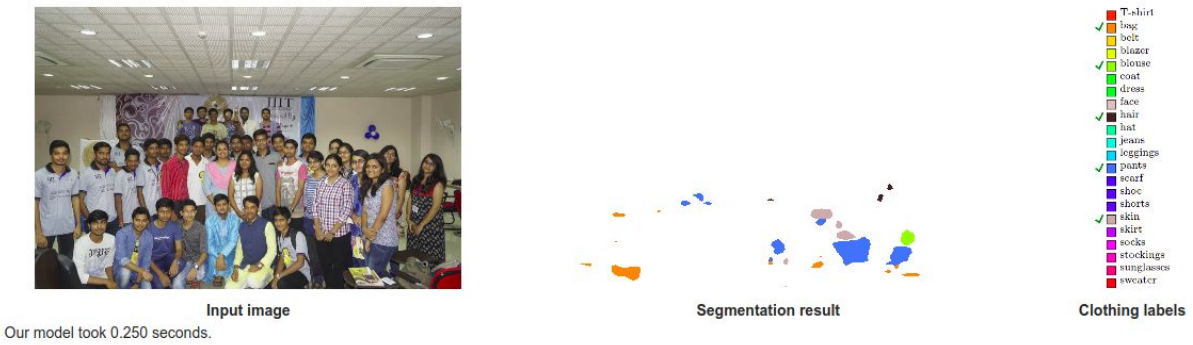

Fig. 12: Output of Paperdoll Pretrained Models on Multiple Persons

We show use of background removal (Grabcut Segmentation) which gives more accurate and better results. (See Fig. 13).

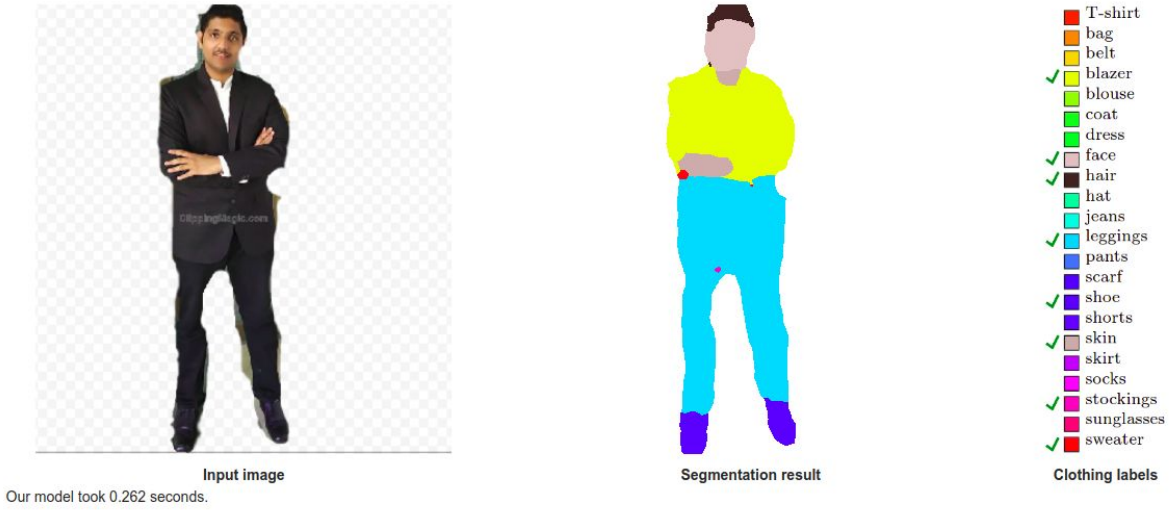

Fig. 13: Effect of Background Removal (using Grabcut Segmentation)

Section 3: Future Work

Currently the model takes ~0.2 seconds for each image which can be reduced using single channel images instead of three RGB Channel Image, thus reducing number of computations. GrabCut Segmentation is semi-manual as it requires a human to make required edits to the automated background subtraction, and thus for real time applications, it's either necessary for user to be provided with an interface to perform background subtraction or find another automated alternative for this task. Since for each person, it takes ~0.2 seconds, for an image with multiple persons, the model is supposed to take greater than 1 seconds, which can prove costly in practical applications. Thus, it's also important to have a high end computational server.

Section 4: Conclusion

We conclude that the use of preprocessing techniques on real time data drastically improves accuracy and time for the multi label classification. We also show comparison of existing networks using Precision Recall for object detection which is used for multiple persons scenarios in Real Time applications. We use ModaNet Dataset for our experimentation, and can conclude positively that the Dataset can be used for Real Time applications. ModaNet has wide varieties of poses for street fashion dataset and is suitable for practical cases. The whole process can be commercialized easily on a larger scale.

Section 5: References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. 2018. Efficient Interactive Annotation of Segmentation Datasets with Polygon-RNN++. In CVPR.

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. 2017. SoftNMS - Improving Object Detection with One Line of Code. In ICCV. 5562–5570.

[3] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. 2017. Annotating Object Instances with a Polygon-RNN. In CVPR.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2018. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018), 834–848. Issue 4.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In CVPR.

[6] Francois Chollet. 2017. Xception: Deep Learning with Depthwise Separable Convolutions. In CVPR.

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

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable Convolutional Networks. In ICCV. 764–773.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In CVPR.

[10] Jian Dong, Qiang Chen, Wei Xia, Zhongyang Huang, and Shuicheng Yan. 2013. A deformable mixture parsing model with parselets. In ICCV.

[11] Ross B. Girshick. 2015. Fast R-CNN. In ICCV. 1440–1448.

[12] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In CVPR. 580–587.
[13] Kota Hara, Vignesh Jagadeesh, and Robinson Piramuthu. 2016. Fashion apparel detection: The role of deep convolutional neural network and pose-dependent priors. In WACV. 1–9.
[14] Bharath Hariharan, Pablo Arbelaez, Ross Girshick, and Jitendra Malik. 2014. Simultaneous Detection and Segmentation. In ECCV.
[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition.
[16] Xuming He, Richard S Zemel, and Miguel A Carreira-Perpinan. 2014. Multiscale Conditional Random Fields for Image Labeling. In CVPR.
[17] Junshi Huang, Rogério Schmidt Feris, Qiang Chen, and Shuicheng Yan. 2015. Cross-Domain Image Retrieval with a Dual Attribute-Aware Ranking Network. In ICCV. 1062–1070.
[18] M. Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C. Berg, and Tamara L. Berg. 2015. Where to Buy It: Matching Street Clothing Photos in Online Shops. In ICCV. 3343–3351.
[19] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and Philip HS Torr. 2009. Associative hierarchical crfs for object class image segmentation. In ICCV.
[20] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Don, Liang Lin, and Shuicheng Yan. 2015. Deep Human Parsing with Active Template Regression. IEEE Trans. Pattern Anal. Mach. Intell. 37, 12 (2015), 2402–2414. [21] Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Si Liu, Jinhui Tang, Liang Lin, and Shuicheng Yan. 2015. Human Parsing with Contextualized Convolutional Neural Network. In ICCV.
[22] G. Lin, C. Shen, A. van den Hengel, and I. D. Reid. 2017. âĂIJExploring context with deep structured models for semantic segmentation. In CVPR.
[23] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. 2017. Feature Pyramid Networks for Object Detection. In CVPR. 936–944.
[24] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. Focal Loss for Dense Object Detection. In ICCV. 2999–3007.
[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In ECCV.
[26] Si Liu, Jiashi Feng, Csaba Domokos, Hui Xu, Junshi Huang, Zhenzhen Hu, and Shuicheng Yan. 2014. Fashion Parsing With Weak Color-Category Labels. IEEE Trans. Multimedia 16, 1 (2014), 253–265.
[27] Si Liu, Xiaodan Liang, Luoqi Liu, Ke Lu, Liang Lin, and Shuicheng Yan. 2014. Fashion Parsing with Video Context. In ACM MUltiMedia.
[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. 2016. SSD: Single Shot MultiBox Detector. In ECCV.

[29] Ziwei Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. 2017. Deep learning markov random field for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell.
[30] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In CVPR. 1096–1104.
[31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In CVPR.
[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In NIPS.
[33] A. G. Schwing and R. Urtasun. 2015. Fully connected deep structured networks. In arXiv:1503.02351.
[34] Evan Shelhamer, Jonathon Long, and Trevor Darrell. 2016. Fully Convolutional Networks for Semantic Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (2016), 640–651. Issue 4.
[35] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. 2006. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In ECCV.
[36] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. 2016. Training Region-Based Object Detectors with Online Hard Example Mining. In CVPR. 761–769.
[37] Marcel Simon, Erik Rodner, and Joachim Denzler. 2016. ImageNet pre-trained models with batch normalization. arXiv preprint arXiv:1612.01452 (2016).
[38] K. Simonyan and A. Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In ICLR.
[39] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi.
2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In AAAI.
[40] Kota Yamaguchi, M. Hadi Kiapour, and Tamara L. Berg. 2013. Paper Doll Parsing: Retrieving Similar Styles to Parse Clothing Items. In ICCV. 3519–3526.
[41] Kota Yamaguchi, M. Hadi Kiapour, Luis E. Ortiz, and Tamara L. Berg. 2012. Parsing clothing in fashion photographs. In CVPR. 3570–3577.
[42] Wei Yang, Ping Luo, and Liang Lin. 2014. Clothing Co-parsing by Joint Image Segmentation and Labeling. In CVPR. 3182–3189.
[43] Aron Yu and Kristen Grauman. 2014. Fine-Grained Visual Comparisons with Local Learning. In CVPR. 192–199.
[44] Fisher Yu and Vladlen Koltun. 2016. Multi-Scale Context Aggregation by Dilated Convolutions. In ICLR.
[45] HengShuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid Scene Parsing Network. In CVPR. 1063–6919.

[46] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. Conditional Random Fields as Recurrent Neural Networks. In ICCV. 1529–1537.