Embedded Systems and Applications 4190.303C
Seoul National University

Homework #(**2**)

Oh Hyun Seok
2014-13485

---

1 Analyze Pascal VOC 2012(only segmentation)

Segmentation: Generating pixel-wise segmentations giving the class of the object visible at each pixel, or ”background” otherwise.

1.1 Class names

The twenty object classes that have been selected are:
Person: person
Animal: bird, cat, cow, dog, horse, sheep
Vehicle: aeroplane, bicycle, boat, bus, car, motorbike, train
Indoor: bottle, chair, dining table, potted plant, sofa, tv/monitor

1.2 Image size

Images in dataset can have diverse sizes. They all have 3 channels (RGB) in common, but their width and height varies according to image. Their class segmentation and object segmentation images also have corresponding diverse image sizes. For instance, for segmentation train dataset there are 244 different image sizes. Their width varies from 112 pixels to 500 pixels, and height varies from 246 to 500 pixels.

1.3 Number of images of each dataset

train : 1464 images listed at VOC2012/ImageSets/Segmentation/train.txt
val : 1449 images listed at VOC2012/ImageSets/Segmentation/val.txt
trainval : 2913 images listed at VOC2012/ImageSets/Segmentation/trainval.txt
trainaug : dataset from VOC2012 are augmented with SegmentationClassAug dataset made by someone from Semantic Boundaries Dataset (SBD) leads to 10582 images for training.

1.4 Meaning of outline

On images at VOC2012/SegmentationObjects or VOC2012/SegmentationClass, you can see that images have white outlines in common. These outlines are used to calculate IoU which is common metric for segmentation evaluation. Also it indicates acceptable error of segmentation. When we compare ground truth image and our segmented image, our image should segment similar to ground truth image, with range of segmentation error on outline allowed. Segmented pixel covered by outline is masked and excluded on evaluation. I got this idea from PASCAL VOC 2012 devkit VOCevalseg.m file.

2 Evaluation Metric: mIoU

For each class, IoU metric is intersection of groundtruth segmentation area and our segmentation area divided by union of a class ground truth segmentation area and out segmentation area. This is equal to the number of correctly classified pixels divided by (number of correctly classified pixels + number of wrongly classified pixel + number of wrongly not classified pixel), equivalent to true positive divided by (true positive+ false positive + false negative). Mean IoU metric, also called mIoU is evaluating IoU under accumulated pixels through images.

2.1 How to calculate mIoU with outline

for a ground truth segmentation image "gtim", outlines have value 255 and segmented pixels which are not outline pixels have value range from 0(background) to 20 (equal to number of classes). Predicted segmentation image "resim" does not have outline pixels, but equal range of segmented pixels. We define "sumim" as $(1 + \text{gtim}) + (\text{resim} * (\text{number of classes} + 1))$, and make a histogram "hs" sized $(\text{number of classes} + 1)^2$ of sumim according to pixel value of sumim, since $1 \leq (1 + \text{gtim}[i, j]) + (\text{resim}[i, j] * (\text{number of classes} + 1)) \leq (\text{number of classes} + 1)^2$. In this case, the number of intersections of gtim and resim in class k $(0 \leq k \leq \text{number of classes})$ equals to $\text{hs}[k, k]$. Also number of pixels of gtim in class k is $\text{hs}[k, :]$, and and resim in class k is $\text{hs}[:, k]$. So union of pixels of gtim and resim in class k is $\text{hs}[k, :] + \text{hs}[:, k] - \text{hs}[k, k]$. We can calculate IoU of class k on one segmentation as

$$\text{IoU}[k[= \frac{\text{hs}[k, k]}{\text{hs}[k, :] + \text{hs}[:, k] - \text{hs}[k, k]}$$

Good thing is that we can accumulate hs for each images to calculate mean IoU for multiple images. At VOCevalseg.m file, hs for each image is accumulated in "confcounts" identifier, and mean IoU for each class k is calculated as

$$\text{accuracies}[k] = \frac{\text{confcounts}[k, k]}{\text{confcounts}[k, :] + \text{confcounts}[:, k] - \text{confcounts}[k, k]} * 100$$

On calculating hs, we calculate gtim < 255 which returns boolean gtim-sized array falsifying only outlines in gtim. Then we mask sumim with it before we make hs out of it, which means that segmentation error on outline area are acceptable.

3 Analyze DeepLab v3+

The architecture on Fig.1. uses Atrous convolution initially to extract low-level features. The feature map from atrous convolution goes through 1x1 convolution to become low-level features. After that, it uses layer structure similar to inception module. But it has difference from inception module in using atrous convolution of 3x3 conv rate 6,12,18 and Image pooling instead of normal convolution 3x3 or 5x5 and max pooling. This is called Atrous Spatial Pyramid Pooling. It concatenates feature maps just like Inception module, then applies 1x1 convolution for effective representation. This feature map is high level features.
On decoder level, high level features are upsampled by 4 and concatenated by low level features. Concatenated multilevel features go through 3x3 convolution and upsampled by 4 to form a per-pixel image segmentation prediction.

3.1 Discuss why DeepLab v3+ gets better accuracy than other segmentation networks

Atrous convolution is based on original convolution, but has a property called rate. it gets value for every stride of rate from input and multiply with kernel. original convolution could be seen as atrous convolution of rate 1. Using atrous convolution, we can increase the size of receptive field without increasing parameters or computations as in original convolution. With larger receptive field, we have larger field-of-view. We can obtain larger field of view also from down sampling, but down sampling decreases the resolution and causes loss of information. atrous convolution can acquire very large field of view using high rate without loss of information and resolution. On image segmentation, decision is done in pixel basis, so gathering context information for each pixel much as possible is very important, and larger field of view

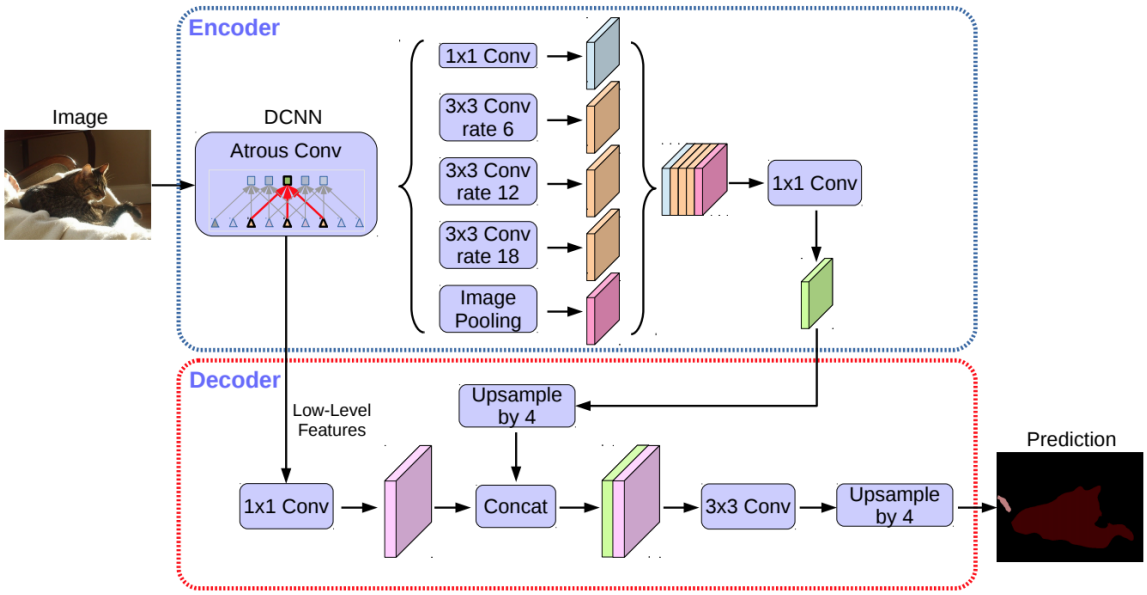


Fig. 2. Our proposed DeepLabv3+ extends DeepLabv3 by employing a encoder-decoder structure. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries.

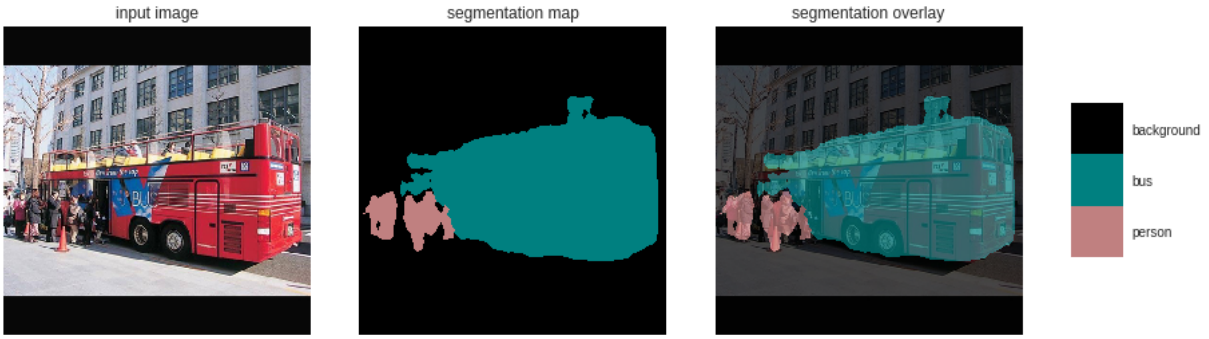
Figure 1: deeplabv3 architecture

gathers context information better. Compared to other segmentation networks using max pooling, frequent upsampling and original convolution, Deeplab v3+ uses atrous convolution majorly and uses less number of upsampling due to less pooling. Therefore Deeplab v3+ can attain much larger field of view and gather more context information in one pixel without losing information or decreasing resolution. This might be the reason Deeplab v3+ gets better accuracy than other segmentation networks.

3.2 Run on a few images on two networks

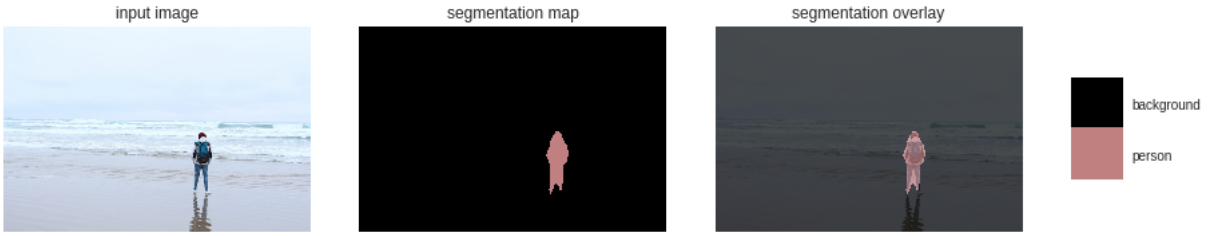
I have run images including one of 6 classes through both MobileNet V2 (mobilenetv2 _coco_voctrainaug) and Xception(xception_coco_voctrainaug) base. I have majorly posted images which failed to correctly segment images. From results, classes which are expected good accuracy seems to be segmented quite well, but both models failed to correctly segment people from image of crowds. Classes which are expected bad accuracy seems to be segmented not quite well, especially for dining table class. Figure 2 and 3 came from MobileNet V2, Figure 4 and 5 came from Xception.

Homework #(**2**)
Oh Hyun Seok
2014-13485

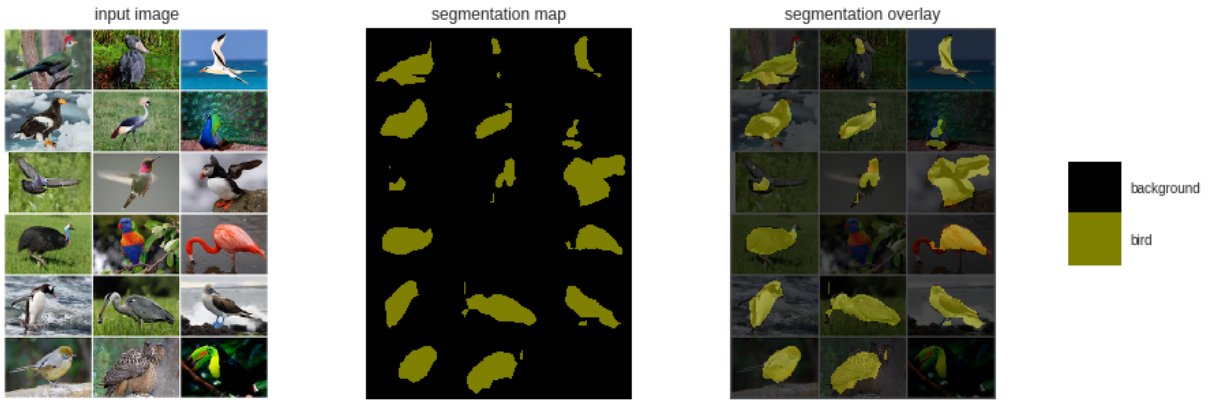


(a) bus

(b) people



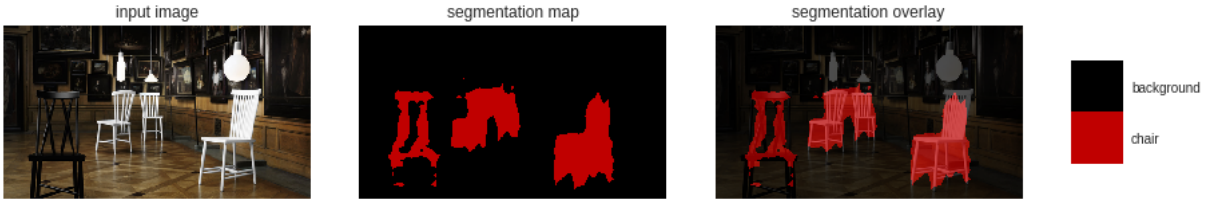
(c) person



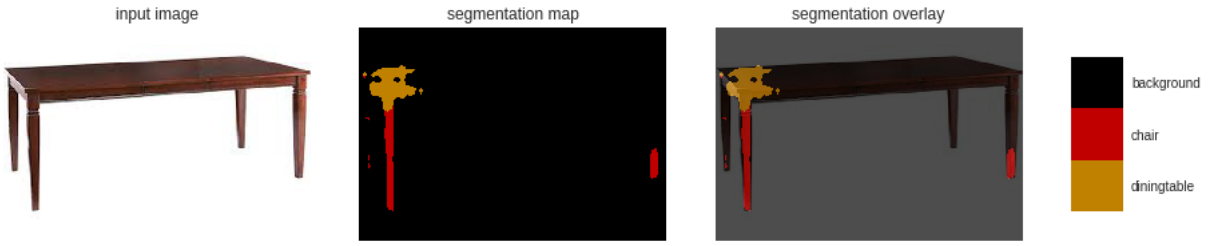
(d) birds

Figure 2: MobileNet v2 Good Accuracy

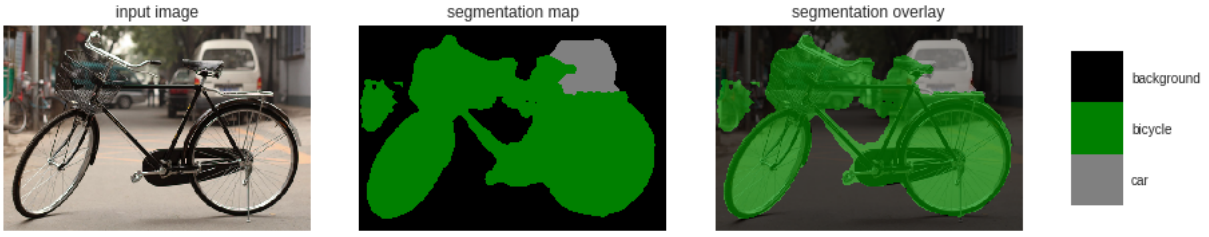
Homework #(**2**)
Oh Hyun Seok
2014-13485



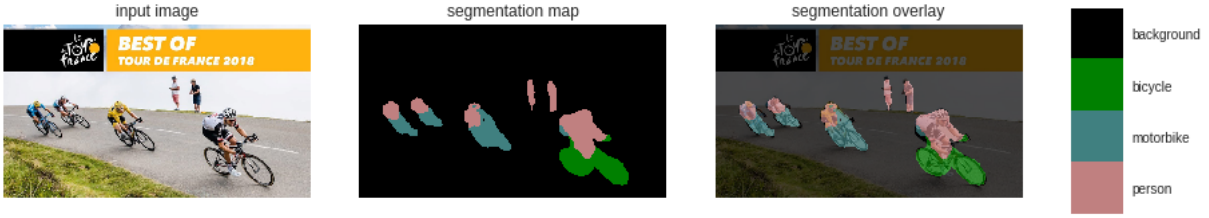
(a) chair



(b) dining table



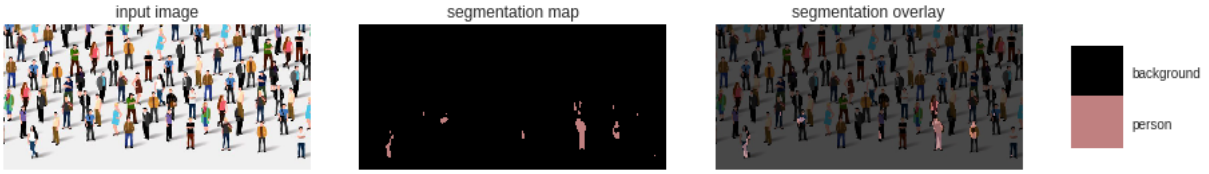
(c) bicycle



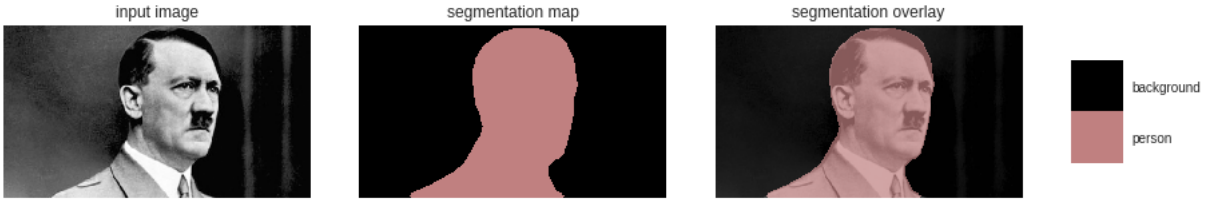
(d) bicycles

Figure 3: MobileNet v2 Bad Accuracy

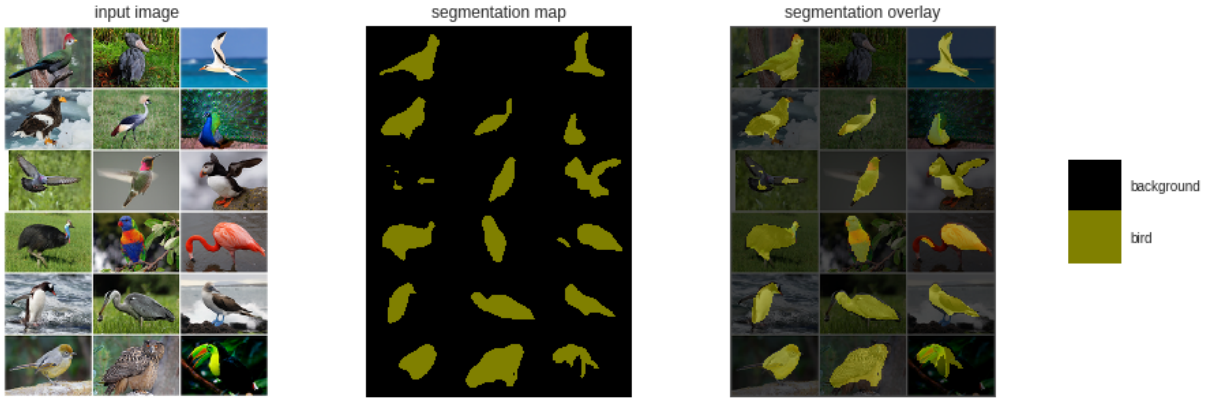
(a) bus



(b) people

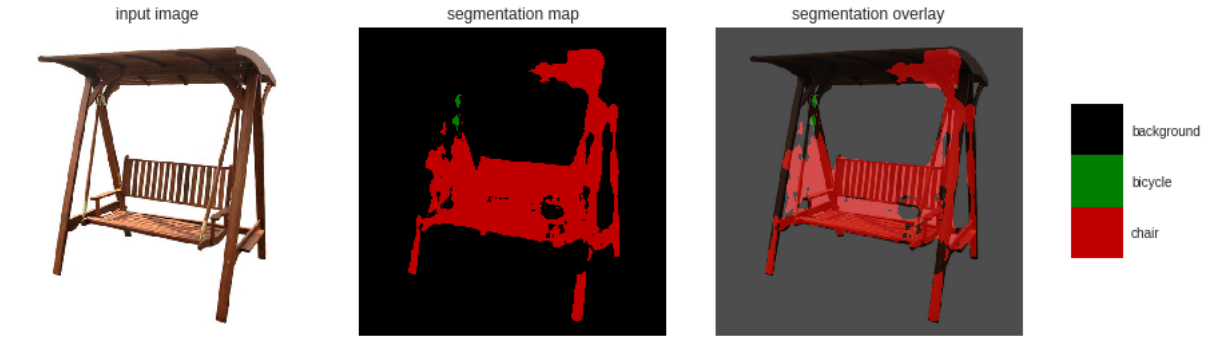


(c) person



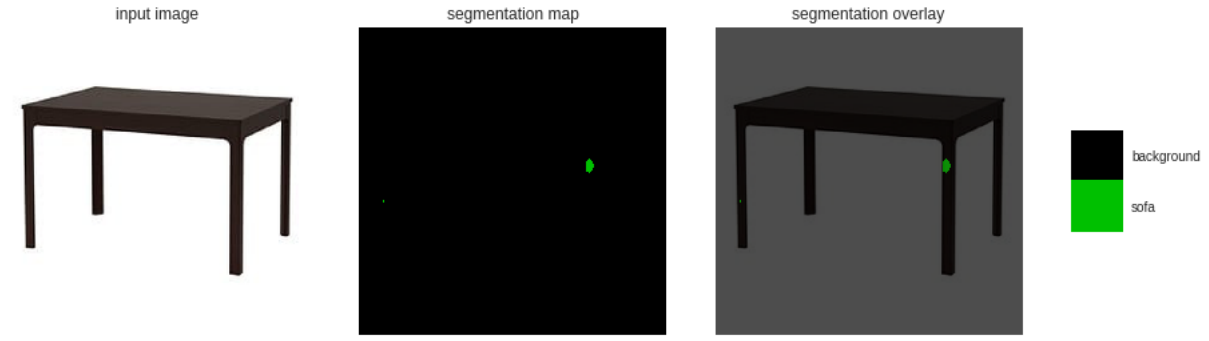
(d) birds

Figure 4: Xception Good Accuracy

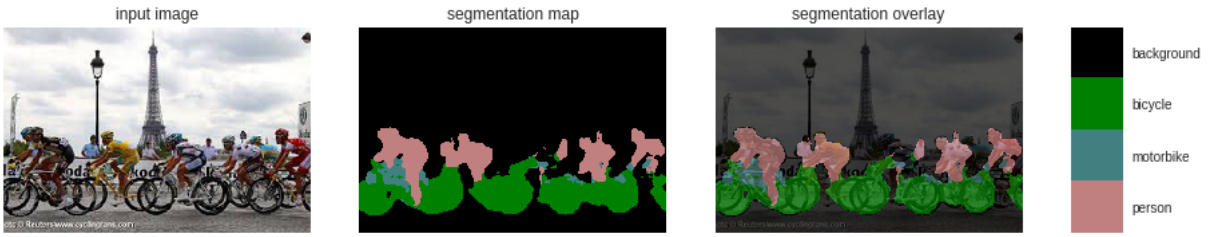


(a) chair

(b) dining table



(c) dining table



(d) bicycles

Figure 5: Xception Bad Accuracy