Object-Contextual-Representations-for-Semantic-Segmentation

December 27, 2019

1 Object-Contextual Representations for Semantic Segmentation

1.1 Paper Details

1. **Authors**: Yuhui Yuan, Xilin Chen, Jingdong Wang
2. **Paper Link**: https://arxiv.org/pdf/1909.11065v2.pdf
3. **Category**: Semantic Segmentation

1.2 Introduction

Semantic Segmentation: Assigning a label to each pixel in an image.

Approach: Contextual Aggregation.

Motivation: Class label assigned to one pixel is the category of the object that the pixel belongs to.

1.2.1 What does Context mean?

The context of one position typically refers to a set of positions, e.g., the surrounding pixels.

If we refer to another paper (**Context Based Object Categorization: A Critical Survey**), Contextual Features are used to represent the interaction of an object with its surroundings. It can be divided into the following 3 categories: 1. **Semantic Context** - This focuses on object co-occurence and allows to correct label of one object without affecting the label of other objects. For example, a tree is more likely to co-occur with a plant than a whale. 2. **Spatial Context** - This focuses on the position of objects. For example, a dog is more likely to be present above grass and below sky rater than above sky and below grass. 3. **Scale Context** - This focuses on relative size of objects. For example, a car is relatively smaller than a truck and not the other way round.

1.2.2 Approach

The approach discussed in the paper consists of the following 3 steps:

1. **Coarse Soft Segmentation** - This involves dividing the contextual pixels (surrounding pixels) into soft object regions. The word "soft" here means that our focus is NOT on carrying out accurate segmentation.
2. **Object Region Representation** - We use the soft segmentation obtained from the above step and the pixel representation to represent each object region.

3. **Object-Contextual Representation** (OCR) - We use the output from the above 2 steps along with Pixel-Region relation to obtain the augmented representations.

Figure 1: The pipeline of the approach discussed in the paper. Source

1.2.3 Differences

OCR vs Multi-Scale Context

1. OCR differentiates contextual pixels which belong to the same class to the contextual pixels which belong to different class.
2. Multi-Scale Context approach only differentiates pixels present at different positions.

OCR vs other Relational Context schemes

The approach discussed in the paper considers not only the object region representations but also the pixel and pixel-region relations, unlike other approaches.

It should also be mentioned here that the current approach is also a relational context approach.

OCR vs Coarse-to-fine Segmentation

While "Coarse-to-fine Segmentation" is also followed in the current approach, the difference is the way the coarse segmentation is used. The OCR approach uses the coarse segmentation to generate a contextual representation, whereas the other approaches use it directly as an extra representation.

OCR vs Region-wise Segmentation

The region-wise segmentation first groups the pixels into **super pixels** which are then assigned a label. OCR on the other hand, uses the grouped regions to learn a better labelling for the pixels, instead of directly using them for segmentation.

1.3 Approach

It's now time to go into the mathematical details of the approach.

1.3.1 Semantic Segmentation - Problem Statement

Given K classes, assign each pixel p_i of image I a label l_i (which is one of the K unique classes).

Multi-Scale Context (Optional)

Multi-Scale context can be represented by the following equation:

$$y_i^d = \sum_{p_s = p_i + d\Delta_t} K_t^d x_s \tag{1}$$

Where,

y_i^d is the output representation of position p_i for the dth dilated convolution,

d is the dilation rate,

t is the index of convolution (-1,0,1 for a 3x3 convolution),

$\Delta_t = (\Delta_w, \Delta_h) \mid \Delta_w = -1, 0, 1, \Delta_h = -1, 0, 1$ for a 3x3 convolution,

x_s is the representation at p_s,

K^d is the kernel for dth dilated convolution

Relational Context (Optional)

Relational context can be represented by the following equation:

$$y_i = \rho\left(\sum_{s \in I} w_{is}\delta(x_s)\right) \tag{2}$$

Where,
y_i is the output representation of position p_i,
I refers to the image,
w_{is} is the relation between x_i and x_s,
$\delta(\cdot)$ and $\rho(\cdot)$ are transform functions,
x_s is the representation at p_s
Next comes my favorite part, the formulation of the current approach.

1.3.2 Step 1: Soft Object Regions

The image I is partitioned into K soft object regions: $\{M_1, M_2, ..., M_K\}$ where M_i corresponds to class i.

M_i is a 2D map where each entry represents the probability of the corresponding pixel belonging to the class i.

1.3.3 Step 2: Object Region Representation

The representations of all the pixels obtained in step 1 are aggregated as follows:

$$f_k = \sum_{i \in I} m_{ki} x_i \tag{3}$$

Where, m_{ki} represents the **normalized** probability of the pixel p_i belonging to class k.

1.3.4 Step 3: Object Contextual Representation

First we obtain the relation between a pixel and an object region:

$$w_{ik} = \frac{e^{\kappa(x_i, f_k)}}{\sum_{j=1}^{K} e^{\kappa(x_i, f_k)}} \tag{4}$$

Where,
$\kappa(\cdot)$ is a relation function,
Finally, we can obtain the object contextual representation y_i for pixel p_i as shown below:

$$y_i = \rho\left(\sum_{k=1}^{K} w_{ik}\delta(f_k)\right) \tag{5}$$

Notice the similarity between equation (5) and equation (2).

1.3.5 Step 4: Augmented Representation

The final representation for pixel p_i is calculated as follows:

$$z_i = g([x_i^T y_i^T]^T) \tag{6}$$

$g(\cdot)$ here is a transform function with the only purpose to join the effect of original representation x_i and object contextual representation y_i

1.4 References

1. Object-Contextual Representations for Semantic Segmentation - https://arxiv.org/pdf/1909.11065v2.pdf
2. Context Based Object Categorization: A Critical Survey - https://vision.cornell.edu/se3/wp-content/uploads/2014/09/context_review08_0.pdf
3. Jupyter Markdown - https://www.ibm.com/support/knowledgecenter/en/SSGNPV_1.1.3/dsx/markd-jupyter.html