Computer Vision Programming Assignment 3

Segmentation

20195053 Dongjun Nam
june6423@gm.gist.ac.kr

Abstract

This assignment delves into the realm of image segmentation, leveraging advanced neural network architectures to refine coarse segmentations obtained using U-Net. By employing CSPN (Convolutional Spatial Propagation Network) and DYSPN (Dynamic Spatial Propagation Network), this study aims to enhance the precision and accuracy of segmentation outputs. The refinement process involves iteratively optimizing segmentation outputs to achieve higher-quality delineation of object boundaries and fine-grained image details.

1. Introduction

Image segmentation remains a fundamental task in computer vision, enabling the precise delineation and identification of objects within images. This assignment explores refining method using neural networks.

The primary goal is to refine coarse segmentations generated by the U-Net architecture. To achieve this refinement, we employ advanced techniques such as CSPN and DYSPN. CSPN makes segmentation more accurate by working together with the information, while DYSPN refines the results by analyzing scenes dynamically.

This report presents an iterative approach to segmentation refinement, aiming to enhance the segmentation outputs by iteratively optimizing and refining the coarse segmentations obtained from U-Net. Through the application of these advanced neural network architectures, this assignment endeavors to achieve more precise object delineation and finer segmentation details.

2. Methodology

We used the SimpleOxfordPet dataset for this experiment. The Simple Oxford Pets Dataset is a collection of images of pets, specifically focusing on 37 different categories of animals. This dataset is typically used for image classification, segmentation, and other computer vision tasks. It's one of the most popular benchmark dataset due to its diversity in pet types, different poses, backgrounds, and lighting conditions present in the images. We trained U-Net using the input images and segmentation groundtruth from SimpleOxfordPet dataset.

2.1. U-Net

The U-Net architecture is a convolutional neural network (CNN) designed for semantic image segmentation. Its unique structure consists of a contracting path, which captures context information, and an expansive path, facilitating precise localization. This architecture is highly effective for medical image segmentation tasks and has found applications across various domains due to its ability to handle small training datasets effectively. The U-Net implementations utilized in this report are referenced from the following links.

https://github.com/milesial/Pytorch-UNet

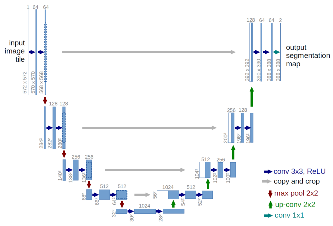


Figure 1: The structure of U-Net

We employed a minimally trained U-Net to generate a preliminary coarse segmentation. Subsequently, we developed a new U-Net model capable of processing both the RGB image and the initial coarse segmentation. This refined U-Net outputs crucial information—such as the affinity or attention tensor—required for the subsequent steps. This output serves as input for CSPN or DYSPN, and the resulting output is compared against the segmentation Ground Truth (GT) of the Oxford Pets dataset. This process facilitates the end-to-end training of the entire model.

2.2. CSPN

The CSPN(Convolutional Spatial Propagation Network) is designed for diverse depth estimation tasks. CSPN functions as an efficient linear propagation model, employing recurrent convolutional operations to facilitate pixel-level propagation. This propagation relies on learned affinity from a deep Convolutional Neural Network (CNN). In contrast to its predecessor, Spatial Propagation Networks (SPN), CSPN showcases considerable practical advantages, offering a 2 to 5 times speed improvement in practical implementations.

In our approach, the CSPN model is employed to refine the coarse segmentation initially obtained from U-Net. This refinement process requires two key inputs: the affinity matrix and the coarse segmentation. To facilitate this, we modified the U-Net architecture to accept both the RGB image and coarse segmentation as input and generate the affinity as output. Subsequently, the CSPN utilizes this affinity alongside the coarse segmentation and the current segmentation from the previous step to update the segmentation iteratively. Initializing with the coarse segmentation, we iteratively refine the segmentation over multiple iterations (in this case, iter=6). The resulting refined segmentation is then employed to train the complete model, consisting of the modified U-Net and CSPN, using segmentation ground truth and Cross Entropy Loss.

2.3. DYSPN

The Dynamic Spatial Propagation Network (DYSPN) represents a paradigm shift in refining segmentation masks by harnessing the attention matrix derived from U-Net and adopting an end-to-end learning strategy. Unlike explicitly reducing importance based on kernel distance, DYSPN operates by integrating the attention matrix obtained from preceding U-Net layers into its architecture, enabling seamless learning from end to end.

Central to DYSPN's functionality is the assimilation of the attention matrix, which encapsulates crucial spatial information regarding pixel relevance and contextual importance within the segmentation mask. By leveraging the attention matrix derived from U-Net, DYSPN capitalizes on the intricate spatial relationships embedded in the matrix, allowing it to dynamically modulate the influence of neighboring pixels during the refinement process.

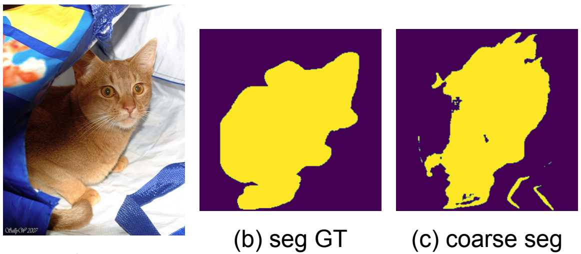
DYSPN's strength lies in its holistic approach, where the network seamlessly integrates the attention matrix to refine segmentation masks iteratively. This integration allows DYSPN to adaptively learn and refine segmentation outputs based on the spatial cues encoded in the attention matrix, leading to more context-aware and precise segmentation refinements.

In our approach, the DYSPN model is employed to refine the coarse segmentation initially obtained from U-Net. This refinement process requires three key inputs: the affinity matrix, the attention matrix and the coarse segmentation. To facilitate this, we modified the U-Net architecture to generate the affinity and attention as output. Subsequently, the DYSPN utilizes this affinity and the attention alongside the coarse segmentation and the current segmentation from the previous step to update the segmentation iteratively. Initializing with the coarse segmentation, we iteratively refine the segmentation over multiple iterations (in this case, iter=6). The resulting refined segmentation is then employed to train the complete model, consisting of the modified U-Net and DYSPN, using segmentation ground truth and Cross Entropy Loss.

3. Results

3.1. Coarse segmentation

The SimpleOxfordPet dataset was utilized to train both the training and validation sets for 3 epochs, resulting in a coarse segmentation. The mIoU achieved on the validation dataset was **0.69**. Fig. 2 illustrates the output specifically for Abyssinian100 image.



(a) RGB image (b) seg GT (c) coarse seg

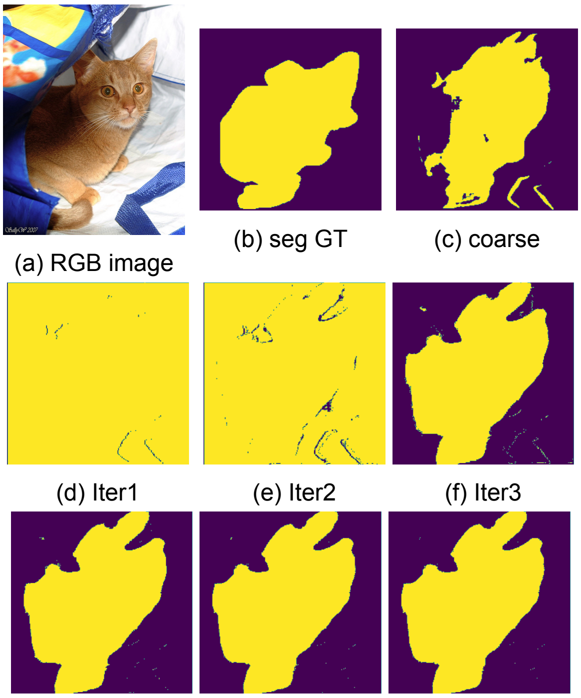
Figure 2: RGB image, segmentation GT and coarse segmentation

3.2. CSPN results

At iter=6 with CSPN and epoch=40, we achieved a validation mIoU of **0.77**. Compared to the initial validation mIoU of **0.69** for coarse segmentation, a significant refinement is evident. Specifically, in Fig. 3, the results for Abyssinian100 are displayed, showcasing an initial coarse segmentation mIoU of **0.66**. However, following the application of CSPN, the mIoU surged to **0.79**, marking a remarkable improvement.

3.3. DYSPN results

At iter=6 with CSPN and epoch=40, we achieved a validation mIoU of **0.48**. Specifically, in Fig. 3, the results for Abyssinian100 are displayed, showcasing an initial coarse segmentation mIoU of **0.66**. However, following the application of CSPN, the



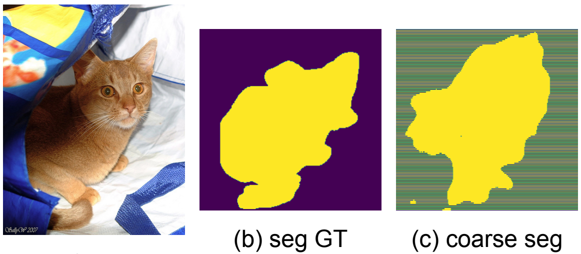
(a) RGB image (b) seg GT (c) coarse

(d) Iter1 (e) Iter2 (f) Iter3

(g) Iter4 (h) Iter5 (i) Iter6

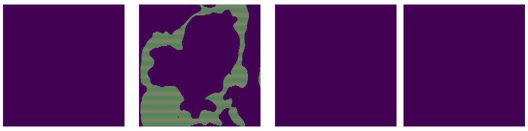
Figure 3: CSPN results on Abyssinian 100

mIoU surged to **0.43**. While this may seem like a very poor refine, the results in Fig. 4 show that it does a good job of capturing the edges of the object, except for the horizontal line error. Our DYSPN result are displayed on Fig.4 and attention are visualized on Fig.5 - Fig.10.



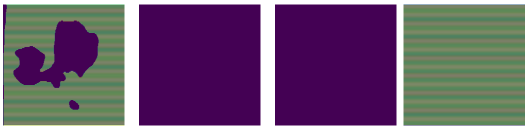
(a) RGB image (b) seg GT (c) coarse seg

Figure 4: RGB image, GT and DYSPN result



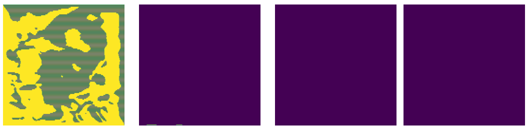
(a) Kernel 0 (b) Kernel 3 (c) Kernel 5 (d) Kernel 7

Figure 5: Attention on iter1



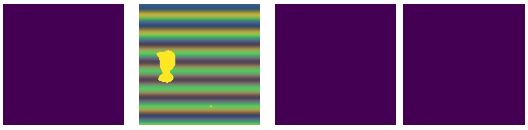
(a) Kernel 0 (b) Kernel 3 (c) Kernel 5 (d) Kernel 7

Figure 6: Attention on iter2



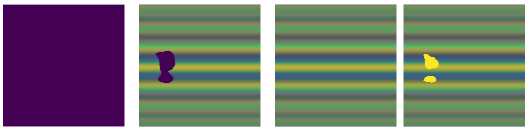
(a) Kernel 0 (b) Kernel 3 (c) Kernel 5 (d) Kernel 7

Figure 7: Attention on iter3



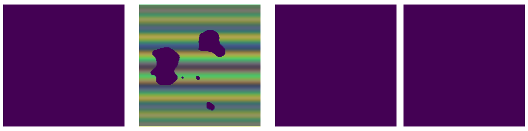
(a) Kernel 0 (b) Kernel 3 (c) Kernel 5 (d) Kernel 7

Figure 8: Attention on iter4



(a) Kernel 0 (b) Kernel 3 (c) Kernel 5 (d) Kernel 7

Figure 9: Attention on iter5



(a) Kernel 0 (b) Kernel 3 (c) Kernel 5 (d) Kernel 7

Figure 10: Attention on iter6

4. Reference

Ronneberger et al. ”U-Net: convolutional networks for biomedical image segmentation”. In Proceedings of 18th International Conference on Medical Image Computing and Computer-Assisted Intervention. 2015.

Y. Chen et al. “Dynamic convolution: Attention over convolution kernels,” CVPR. 2020.

Cheng X, Wang P, Yang R. ”Learning depth with convolutional spatial propagation network”. IEEE TPAMI, 2020.