Fully Convolutional Networks for Image Segmentation

Cristina Isabel González Osorio
ci.gonzalez10@uniandes.edu.co

Raúl Santiago Molina Rodríguez
rs.molina10@uniandes.edu.co

1. Results

Table 1: Evaluation of the networks's performance in test set.

Fully Convolutional Network	Jaccard index
32s from vgg weights	
32s from scratch	
16s from 32s weights	
16s from vgg weights.	
16s from scratch	

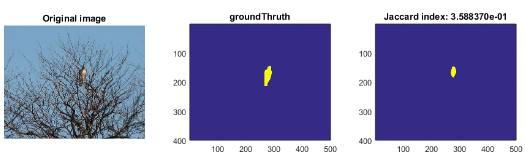

Figure 1: Result from the network 16s with 32s weights.

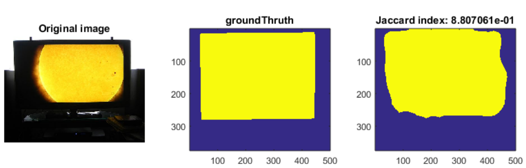

Figure 2: Result from the network 16s with 32s weights.

2. Discussion

The scale of the network and the training datasets require great amounts of resources and time. Network training required >3GB of GPU memory with a batch size of 1, so each training epoch took around 5 hours given these restraints. Using pre-trained weights decreases the amount of time necessary in order to achieve acceptable results. Training any of the architectures from scratch would take time in the order of weeks for it to be effective, an inefficient approach when there are limited resources.

The scale in the final prediction layer of the segmentation determines the level of detail obtained in it. However, it is possible to include information from lower layers, which, in turn, decreases the number of pixels of the limit scale at the output. However, in segmentation tasks the optimal scale at which this process is carried out is determined mainly by each specific problem. In this way, the size of the objects to be segmented allows us to estimate the scale at which the best results would be obtained. In this sense, as the objects of interest are larger, it is possible to obtain appropriate segmentations from scales of a greater number of pixels.This can be seen illustrated in the results of the figures 1and 2

The 32s uses less resources for training and does not need implementation of a DAG like the 16s or 8s architectures, although it might not give fine-grain results, its accurate enough for practical use, an embedded model might use lower resources (thus less energy over time) while running in real time, for example a self-driving car (on a budget).

References