Learning to Refine Object Segments

mingzailao

Wed Jan 4 15:45:02 2017

Outline

Network Structure

DeepMask

Fully convolutional networks for semantic segmentation

SharpMask

Learning to Refine Object Segments

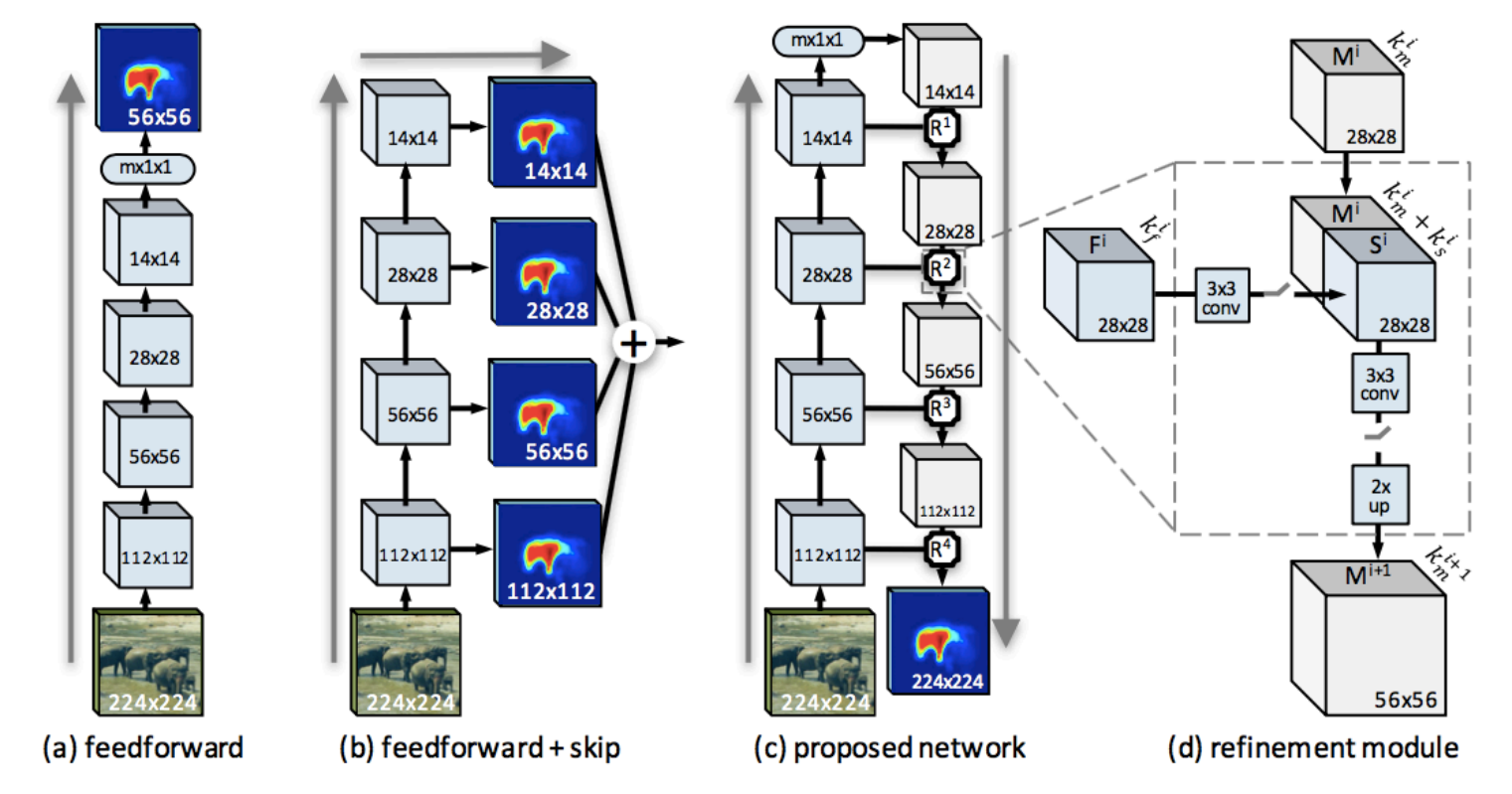
Network Structure

DeepMask

Fully convolutional networks for semantic segmentation

SharpMask

Network Structure



Network Structure

- Feedforward nets, such as DeepMask[1] (a)
- with skip architectures,such as Fully convolutional networks for semantic segmentation[2] (b)
- This model (c,d)

---

[1]Pinheiro P O, Collobert R, Dollar P. Learning to segment object candidates[C]//Advances in Neural Information Processing Systems. 2015: 1990-1998.

[2]Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 3431-3440.

Learning to Refine Object Segments

Network Structure

DeepMask

Fully convolutional networks for semantic segmentation

SharpMask

DeepMask Structure

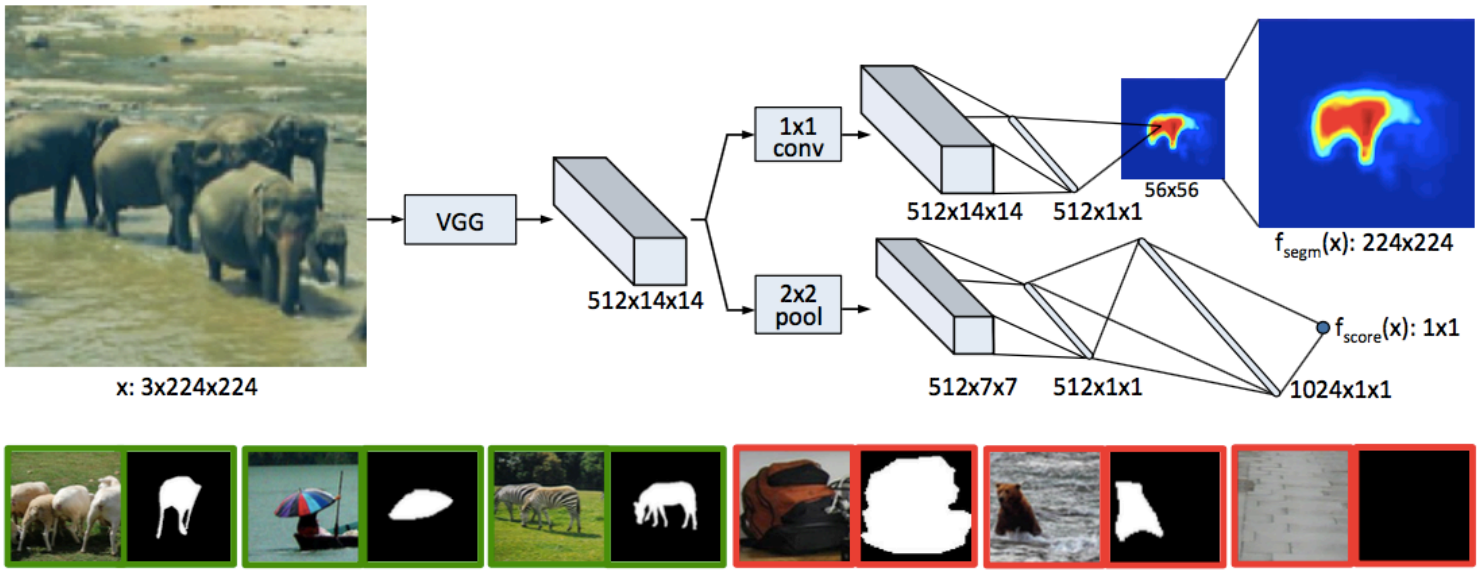


Figure 1: **(Top)** Model architecture: the network is split into two branches after the shared feature extraction layers. The top branch predicts a segmentation mask for the the object located at the center while the bottom branch predicts an object score for the input patch. **(Bottom)** Examples of training triplets: input patch x, mask m and label y. Green patches contain objects that satisfy the specified constraints and therefore are assigned the label $y = 1$. Note that masks for negative examples (shown in red) are not used and are shown for illustrative purposes only.

DeepMask notations

Each sample k in the training set is a triplet containing

- the RGB input x_k
- the binary mask corresponding to the input patch m_k ($m_k^{ij} \in \{-1, +1\}$)
- a label y_k which specifies whether the patch contains an object

Specifically, a patch x_k is given label $y_k = 1$ if it satisfies the following constraints

1. the patch contains an object roughly centered in the input patch
2. the object is fully contained in the patch and in a given scale range

DeepMask Segmentation Branch

Points

- The branch of the network dedicated to segmentation is composed of a single 1×1 convolution layer(ReLU non-linearity) followed by a classification layer.
- The classification layer consists of $h \times w$ pixel classifiers
- Decompose the classification layer into two linear layers with no non-linearity in between.

set the output of the classification layer to be $h^o \times w^o$ with $h^o < h$ and $w^o < w$

DeepMask Loss

Given an input patch $x_k \in \mathcal{I}$, the model is trained to jointly infer a pixel-wise segmentation mask and an object score.

$$\mathcal{L}(\theta) = \sum_k \Big(\frac{1+y_k}{2 * w^o * h^o} \sum_{i,j} \log\big(1 + e^{-m_k^{ij} f_{segm}^{ij}(x_k)}\big) \quad (1)$$

$$+ \quad \lambda \log\big(1 + e^{-y_k f_{score}(x_k)}\big)\Big) \quad (2)$$

DeepMask Implementation Details

Look for the paper

Learning to Refine Object Segments

Network Structure

DeepMask

Fully convolutional networks for semantic segmentation

SharpMask

PASS

PASS

Learning to Refine Object Segments

Network Structure

DeepMask

Fully convolutional networks for semantic segmentation

SharpMask

Refinement Overview

each module R^i takes as input a mask encoding M^i generated in the top-down pass, along with mathcing features F^i in the bottom-up pass, and learn to merge the information to generate a new upsampled object encoding M^{i+1}.

$$M^{i+1} = R^i(M^i, F^i) \tag{3}$$

Refinement Details

- the simplest way: concatenate M^i and F^i

Two problem :

1. $k_f^i >> k_m^i$,loss the information of M^i
2. In mordern CNN, k_f^i can be quite large, using F^i directly would be computationallyexpensive.

Refinement Details

- solution

by a 3×3 convolutional module s.t. $k_s^i << k_f^i$

KEYS: the convolutional modules are shared

Refinement Details

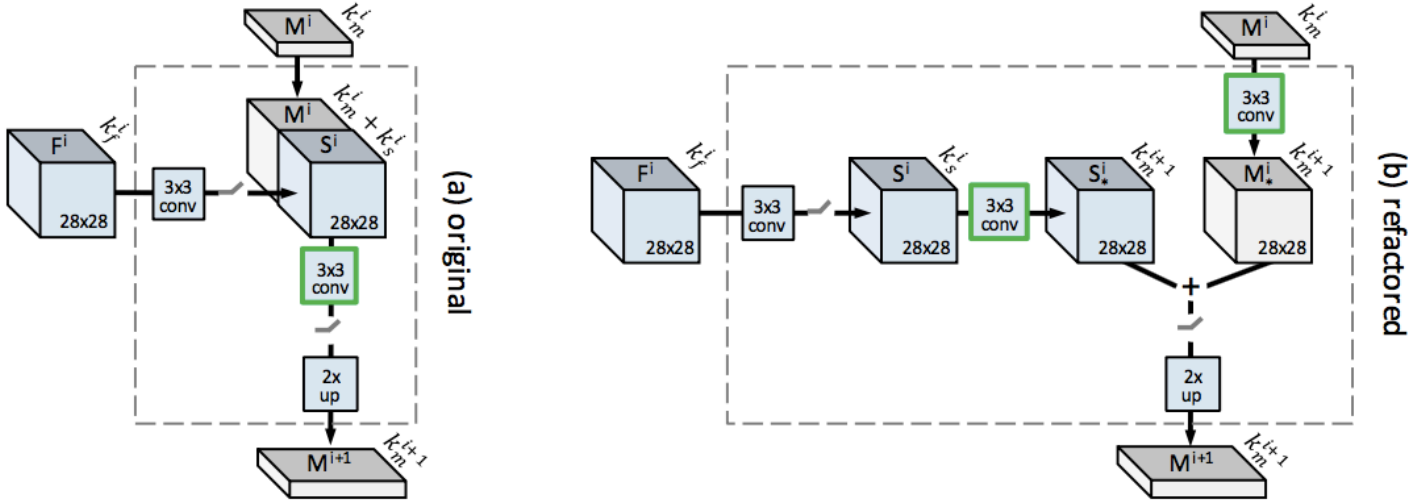


Fig. 7: (a) Original refinement model. (b) Refactored but *equivalent* model that leads to a more efficient implementation. The models are equivalent as concatenating along depth and convolving along the spatial dimensions can be rewritten as two separate spatial convolutions followed by addition. The green 'conv' boxes denote the corresponding convolutions (note also the placement of the ReLUs). The refactored model is more efficient as skip features (both S^i and S^i_*) are shared by overlapping refinement windows (while M^i and M^i_* are not). Finally, observe that setting $k^i_m = 1$, $\forall i$, and removing the top-down convolution would transform our refactored model into a standard 'skip' architecture (however, using $k^i_m = 1$ is not effective in our setting).

Training and Inference

- first, the model is trained to jointly infer a coarse pixel-wise segmentation mask and an object score;
- second, the feedforward path is 'frozen' and the refinement modules trained

Once learning of the first stage converges, the final mask prediction layer of the feedforward network is removed and replaced with a linear layer that generates a mask encoding M^1 in place of the actual mask output.

Feedforward Architecture

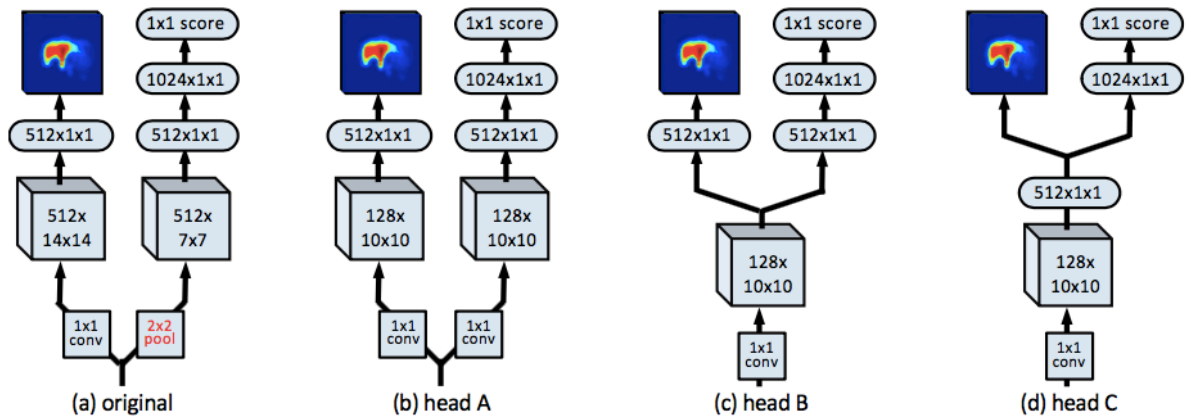


Fig. 3: Network head architecture. (a) The original DeepMask head. (b-d) Various head options with increasing simplicity and speed. The heads share identical pathways for mask prediction but have progressively simplified score branches.

Experiments Details

READ THE PAPER