Scalable Cascade Inference for Semantic Image Segmentation

Paul Sturgess[1]
paul.sturgess@brookes.ac.uk
L'ubor Ladický[2]
lubor@robots.ox.ac.uk
Nigel Crook[1]
ncrook@brookes.ac.uk
Philip H. S. Torr[1]
philiptorr@brookes.ac.uk

[1] Department of Computing
Oxford Brookes University
Oxford,UK

[2] Department of Engineering Science
University of Oxford
Oxford, UK

Semantic image segmentation (SIS) is a problem of simultaneous segmentation and recognition of an input image into regions and their associated categorical labels, such as person, car or cow. A popular way to achieve this goal is to assign a label to every pixel in the input image and impose simple structural constraints on the output label space. Such approaches have been successfully formulated as pairwise conditional random fields (CRF) and higher order CRFs [3]. These approaches are now practically solvable for some problems due to advances in inference techniques. Currently the α-expansion [1] algorithm has proved to be perhaps the most efficient approximation algorithm for the SIS problem and is amongst the state-of-the art for quantitative performance. Empirically the algorithm's runtime is linear in the number of labels, making it practical only when working in a specific domain that has few classes-of-interest ($10-20$ for example). However when working in a more general setting where the number of classes could easily reach tens of thousands, sub-linear complexity is required. In this paper we propose to meet this requirement by dividing the large label set into smaller more manageable ones, and then only solving for some of these subsets. Since the SIS problem is concerned with categorical labels a natural way to subdivide the label set is by building a hierarchy, or taxonomy. Given a hierarchy we propose a cascade architecture that can reject whole portions of the label space at the early stages of the optimisation. We also dynamically subdivide the image into smaller and smaller regions during inference to gain further efficiency. The use of a cascade is motivated by the observation that even with a large label domain, a single image will usually only contain a small subset of classes.

We demonstrate the effectiveness of the approach with quantitative evaluation of performance on the SUN09 database [2] that has 107 labels.

1 Cascaded Inference

In order to obtain scalable SIS we propose a to perform cascade style inference. In this section we specify the details of our approach. First we define two general functions:

$$\begin{aligned} \textit{variable selection} \quad & T_\delta : V \rightarrow V', \\ \textit{variable assignment} \quad & T_V : \delta \rightarrow \delta', \end{aligned}$$

that can be applied respectively to the variables (vertices) and the label domain of the cost function; T_δ transforms the current set of variables, given a domain; T_V modifies the current domain, given some variables. We can specify these transformation functions in different ways such that their evaluation performs a move for many move making algorithms. Here we are interested in specifying them in order to perform cascaded inference over a tree structured label space, or taxonomy τ. We define such a space with reference to an unstructured domain Δ as recursive subdivision into disjoint subsets δ such that the root node contains all the elements of Δ and leaf nodes contain the elementary labels $l \in \Delta$. Now, let δ denote a group of siblings, that is a set of children that share the same direct parent in the tree and thus forms a subdomain of Δ. Also let $\pi(\delta)$ signify the domain that the shared parent belongs to, i.e. If the domain $\Delta = \{cat, dog, car, van\}$, then we could have the following groupings that form our tree; The head node would be $everything = \{cat, dog, car, van\}$ and it may have two children, such as $animal = \{cat, dog\}$ and $vehicle = \{car, van\}$. In turn these would then have two leaf nodes as children. Then $\pi(vehicle)$ points to the domain $everything$ and $\pi(dog)$ points to the label domain $\{cat, dog\}$. Thus a tree defines a set of domains $\{\delta_1, ...\delta_{n+1}\}$, where n is the number of sibling groups. For convenience we also maintain an index δ_i^j to the j^{th} elementary label contained within the i^{th} domain, i.e. $vehicle^1 = car$, as does $everything^3$. Given these notations variable selection and assignment based on a tree is then defined as:

$$T_v(\delta) = \begin{cases} \delta_i & \text{if } \delta_i^j \in f^*_{\pi(\delta)} \text{ and } \delta \neq \emptyset \\ \emptyset & \text{otherwise,} \end{cases} \tag{1}$$

$$T_\delta(v) = v \in \{\mathbf{I}(f(T_\delta(v)) \neq \inf\} \tag{2}$$

where $\emptyset$ is the empty set, $\mathbf{I}$ is an indicator function, $f^*_{\pi(\delta)}$ is a given solution for the a labelling problem defined on the domain $\pi(\delta)$ and variables V' and

$$f(T_\delta(v)) = \begin{cases} c^\tau(v, f(v)) & \text{if } f(v) \in \delta \\ \infty & \text{otherwise} \end{cases}, \tag{3}$$

$$c^\tau(v, f(v)) = \arg \min_{f(v) \in \delta_i} c(v, f(v))). \tag{4}$$

For the first layer of the tree $f^*_{\pi(\delta)}$ is trivial since $\pi(\delta)$ is the single label domain of the head node, i.e. $f : V \rightarrow [1]$. This means that we have to solve a k label problem at the start of our cascade, where k is the number of children of the head node. In our running example this would be the $\{animal, vehicle\}$ domain on all variables V of the original graph. However when we visit all the nodes in the tree in the following fashion:-

for all i minimize:

$$Q(f) = \sum_{v \in T_{\delta_i}(v)} c^\tau(v, f(v)) + \sum_{(u,v) \in \mathcal{E}'} w(u,v) \cdot d(f(u), f(v))$$

subject to:

$$\begin{aligned} & f : v \rightarrow \alpha && \forall v \in T_{\delta_i},\ \exists \alpha \in T_v \\ & d(\alpha, \alpha) = 0 && \forall \alpha \in T_v \\ & d(\alpha, \beta) = d(\beta, \alpha) \geq 0 && \forall \alpha, \beta \in T_v \\ & d(\alpha, \beta) \leq d(\alpha, \gamma) + d(\gamma, \beta) && \forall \alpha, \beta, \gamma \in T_v \\ & w(u,v) \geq 0 && \forall u, v \in T_{\delta_i}, \end{aligned} \tag{5}$$

many sub-problems will be trivial such as:- no labels, $|\delta| = \emptyset$; a single label $|\delta| = 1$; no finite cost variables $\forall v \in T_v : c(v, f_\delta(v)) = \infty$. In these cases, we need not evaluate the function at all, saving computation time. In the cases where the cost is non-trivial with binary $\delta = \{\alpha, \beta\}$, or a multi-class domain with $|\delta| > 2$ and $\exists v \in T_v : c(v, f_\delta(v)) \neq \infty$. The cost function remains metric since we only modify the data term $c(.,.)$, thus we can approximately solve it using α-expansion or other suitable methods. We show that our cascaded approach achieves a good approximation, $Q(\bigcup_{i \in leafs} Q(f^*_{\delta_i})) \approx Q(f^*_\Delta)$.

[1] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23:2001, 2001.

[2] Myung Jin Choi, Joseph J. Lim, Antonio Torralba, and Alan S. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010.

[3] Lubor Ladicky, Chris Russell, Pushmeet Kohli, and P. H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.