Universal_Semi_Supervised_Semantic_Segmenta

Universal Semi-Supervised Semantic Segmentation

ICCV 2019

5+ citations

- Major contributions:
  - Single model for segmentation across various domains.
  - Unlabeled data being used.
  - Use of few labels.
  - Works across different domain of label spaces.
- Domain shift is one of the major challenge in semantic segmentation(or any machine learning problem).
- We need to learn segmentation representations that may be shared across domains.
- Unsupervised domain adaptation is a related paradigm which leverages data from one or more source domains to learn a classifier for a new unsupervised target domain in the presence of a domain shift, but these approaches does not leverage target domain data to improve source performance. They are designed for the restrictive setting of large-scale labeled source domain and unlabeled target domain.
- Most of the domain adaptation assume that both source and target domain have overlapping label space.
- **The goal is to simultaneously limit training cost through reduced annotations and deployment cost by obtaining a single model to be used across domains.**
- Other ways:
  - *Fine-Tuning*: Requires large source labels and small amount of target data, **but there will be separate models for each target domain**
  - *Joint-Training*: Results in a unified model across domains, but does not leverage unlabeled data available in each domain.

Use of Entropy-Regularization:

- Large number of unlabeled examples are used to align pixel level deep feature representations from multiple domain using entropy regularization based objective functions.
- Similarity score vector between the features and the label embedding(computed from class prototypes) and minimize the entropy of this discrete distribution to positively align similar examples between the labeled and the unlabeled images.
- This alignment is done both within domian, as well as across domains.

Notes:

- Prototypes computed from class wise feature representations as the label embeddings
- Metric Learning based approaches such as fine grained classification, latent hierarchy learning and zero-shot prediction have benefited greatly by projecting images and labels into a common subspace.

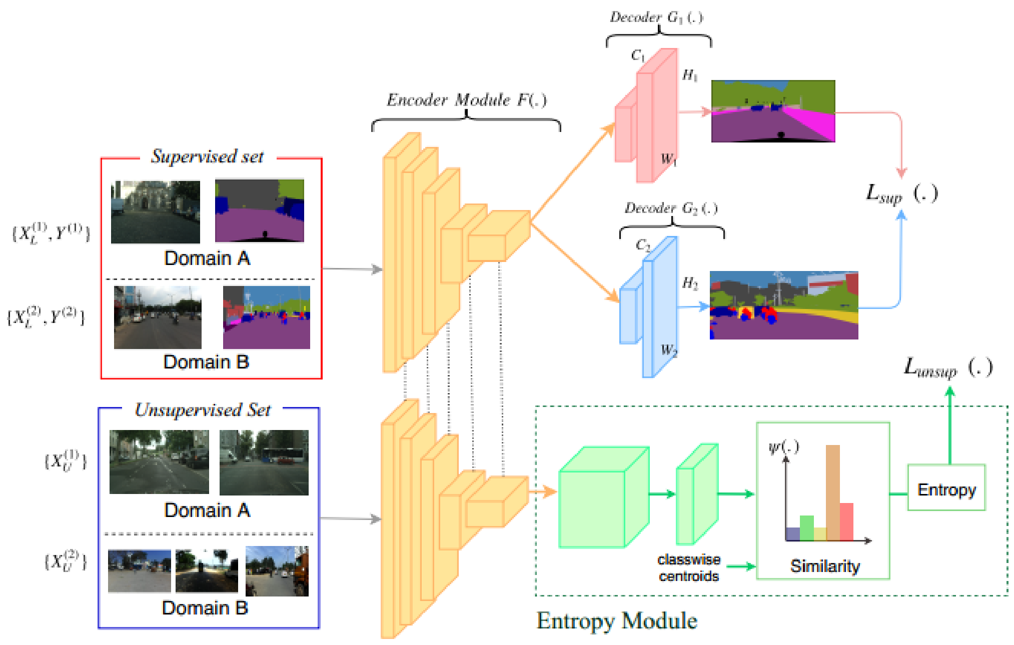

Figure 2: Different modules in the proposed universal semantic segmentation framework. $\{\mathrm{X}_l^{(1)}, \mathrm{Y}^{(1)}\}$, $\{\mathrm{X}_l^{(2)}, \mathrm{Y}^{(2)}\}$ are the labeled examples and $\mathrm{X}_u^{(1)}$, $\mathrm{X}_u^{(2)}$ are the unlabeled examples. Both the datasets share an encoder but have two different decoder layers $\mathcal{G}_1$ and $\mathcal{G}_2$ to enable prediction in their respective output spaces. The entropy module uses the unlabeled examples to perform alignment of pixel wise features from multiple domains by calculating pixel wise similarity with the labels, and minimizing the entropy of this discrete distribution.

- **Universal Segmentation:**
    - Universal Segmentations builds on the idea of training a single joint model that is useful across multiple semantic segmentation domains with possibly different label spaces to make use of transferable representations at lower levels of the network.
    - Pixel level class prototypes can be used for performing semantic transfer across domains.

$\{\mathrm{X}_l^{(i)}, \mathrm{Y}^{(i)}\}_{i=1}^{N_l^{(i)}}$, where $\mathrm{Y}^{(i)} \in \mathcal{Y}_i$ and $N_l^{(i)}$ is the number of labeled examples. The unlabeled images are represented by $\{\mathrm{X}_u^{(i)}\}_{i=1}^{N_u^{(i)}}$ where $N_u^{(i)}$ is the number of unlabeled examples. We work with domains with very few labeled examples, so $N_u^{(i)} \gg N_l^{(i)}$. We consider the general case of non-intersecting label spaces, that is $\mathcal{Y}_p \neq \mathcal{Y}_q$ for any p, q. The label spaces might still have a partial overlap between them, which is common in the case of segmentation datasets. For ease of notation, we consider the special case of two datasets $\{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}\}$, but similar idea can be applied for the case of multiple datasets as well.

The proposed universal segmentation model is summarized in Figure 2. We first concatenate the datasets $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$ and randomly select samples from this mixture in each mini-batch for training. Deep semantic segmentation architectures generally consist of an encoder module which aggregates the spatial information across various resolutions and a decoder module that up samples the image to enable pixel wise predictions at a resolution that matches the input. In order to enable joint training with multiple datasets, we modify this encoder decoder architecture by having a shared encoder module $\mathcal{F}$ and different decoder layers $\mathcal{G}_1(.)$, $\mathcal{G}_2(.)$ for prediction in different label spaces. For a labeled input image x_l, the pixel wise predictions are denoted by $\hat{y}^{(k)} = \mathcal{G}_k(\mathcal{F}(\mathrm{x}_l))$ for $k = 1, 2$ which, along with the labeled annotations, gives us the supervised loss. To include the visual information from the unlabeled examples X_u, we propose an entropy regularization module $\mathcal{E}(.)$. This entropy module takes as input the output of the encoder $\mathcal{F}(.)$ to give pixel wise embedding representations. The entropy of the pixel level similarity scores of these embedding representations with the label embeddings result in the unsupervised loss term. Each of these loss terms is explained in detail in the following sections.

Supervised Loss The supervised loss is calculated separately at each output softmax layer as a masked cross entropy loss between the predicted segmentation mask at that layer $\hat{y}$ and the corresponding pixel wise ground truth segmentation masks for all labeled examples. Specifically, for the output softmax layer k which corresponds to dataset k,

$$\mathcal{L}_S^{(k)} = \frac{1}{N_l^{(k)}} \sum_{i, \mathrm{x}_{l_i} \in \mathcal{D}^{(k)}} \psi_k \left(\mathrm{y}_i, \mathcal{G}_k \left(\mathcal{F}\left(\mathrm{x}_{l_i}\right)\right)\right), \quad (1)$$

where ψ_k is the softmax cross entropy loss function over the label space $\mathcal{Y}_k$, which is averaged over all the pixels of the segmentation map. $\mathcal{L}_S^{(1)}$ and $\mathcal{L}_S^{(2)}$ together comprise the supervised loss term $\mathcal{L}_S$.

deepak

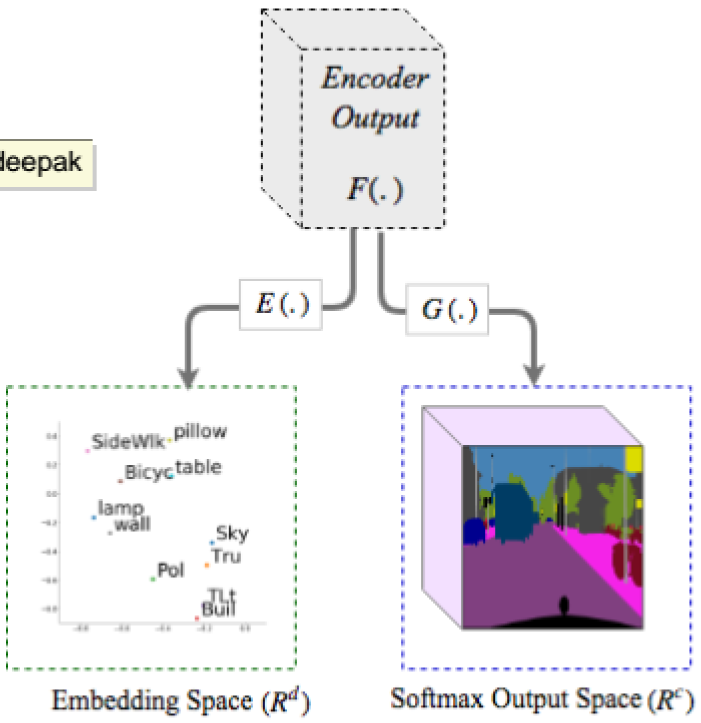

Figure 3: In addition to a traditional decoder layer that outputs predictions in the respective label spaces $\mathbb{R}^c$, the proposed network consists of a domain matching entropy module $\mathcal{E}(.)$ that first maps the features of both the domains into a common embedding space $\mathbb{R}^d$, and then calculates similarity scores with the label embeddings of respective datasets.

the unsupervised images. We introduce a pixel level entropy regularization module, which helps in aligning the visually similar pixel level features from both the datasets calculated from the segmentation network close to each other in an unsupervised way.

The entropy module $\mathcal{E}$ takes as input the encoder output $\mathcal{F}(\mathrm{x})$ and projects the representation at each pixel into a d dimensional embedding space $\mathbb{R}^d$. A similarity metric $\phi(.,.)$ which operates on each pixel is then used to calculate the similarity score of the embedding representations with each of the d dimensional label embeddings using the equation

$$[v_{ij}]_k = \phi\left(\mathcal{E}\left(\mathcal{F}\left(\mathrm{x}_u^{(i)}\right)\right), \mathrm{c}_k^{(j)}\right) \ \ \forall k \in \{|\mathcal{Y}_j|\}, \qquad (2)$$

where $\mathrm{x}_u^{(i)}$ is an image from the i^{th} unlabeled set, $\mathrm{c}_k^{(j)} \in \mathbb{R}^d$ is the label embedding corresponding to the k^{th} label from the j^{th} dataset and $[v_{ij}] \in \mathbb{R}^{|\mathcal{Y}_j|}$. When $i = j$, the scores correspond to the similarity scores within a dataset, and when $i \neq j$, they provide the cross dataset similarity scores.

To calculate the vector representation of the labels of a dataset, we first train an end to end segmentation model with all the supervised training data available from that dataset. Using the output of the encoder of this network, we calculate the centroid of the vector representation of all the pixels belonging to a label to obtain the label embedding for that particular label. These centroids have then been used to initialize the label embeddings for the universal segmentation model. In our experiments, the label embeddings have been

•

crucial differences exists between the two. First, the entropy module projects the encoder features into a large d dimensional embedding space, while the decoder projects the features into a much smaller label space. Also, unlike the decoder, features corresponding to both the datasets are projected into a common embedding space by the entropy module.

•

Unsupervised Loss We have two parts for the unsupervised entropy loss. The first part, the cross dataset entropy loss, is obtained by minimizing the entropy of the cross dataset similarity vectors.

$$\mathcal{L}_{US,c} = \mathcal{H}(\sigma([v_{12}])) + \mathcal{H}(\sigma([v_{21}])), \tag{3}$$

where $\mathcal{H}(.)$ is the entropy measure of a discrete distribution, $\sigma(.)$ is the softmax operator and the similarity vector $[v]$ is from Eq (2). Minimizing $\mathcal{L}_{US,c}$ makes the probability distribution *peaky* over a single label from a dataset, which helps to align visually similar images across datasets close to each other improving the overall prediction of the network. In addition, we also have a within dataset entropy loss given by

$$\mathcal{L}_{US,w} = \mathcal{H}(\sigma([v_{11}])) + \mathcal{H}(\sigma([v_{22}])) \tag{4}$$

which aligns the unlabeled examples within the same domain.

The total loss $\mathcal{L}_T$ is the sum of the supervised loss from Eq (1), and the unsupervised loss terms from Eq (3) and Eq (4), written as

$$\begin{aligned}\mathcal{L}_T = \; & \mathcal{L}_S(\mathrm{X}_l^{(1)}, \mathrm{Y}^{(1)}, \mathrm{X}_l^{(2)}, \mathrm{Y}^{(2)}) + \alpha \cdot \mathcal{L}_{US,c}(\mathrm{X}_u^{(1)}, \mathrm{X}_u^{(2)}) \\ & + \beta \cdot \mathcal{L}_{US,w}(\mathrm{X}_u^{(1)}, \mathrm{X}_u^{(2)}) \end{aligned} \tag{5}$$

where α and β are a hyper parameters that control the influence of the unsupervised loss in the total loss.

•

Multiple Centroids Many labels in a segmentation dataset often appear in more than one visual form or modalities. For example, *road* class can appear as dry road, wet road, shady road etc., or a class labeled as *building* can come in different structures and sizes. To better capture the multiple modalities involved in the visual information of the label, we propose using multiple embeddings for each label instead of a single mean centroid. This is analogous to polysemy in vocabulary, where many words can have multiple meanings and can occur in different contexts, and context specific

Inference For a query image $\mathrm{q}^{(k)}$ from dataset k during test time, the output of the k^{th} decoder $\hat{\mathrm{y}}^{(k)} = \mathcal{G}_k(\mathcal{F}(\mathrm{q}^{(k)}))$ is used to obtain the segmentation map over the label set $\mathcal{Y}_k$ which gives us the pixel wise label predictions. Although we calculate feature and label embeddings in our method and metric based inference schemes like nearest neighbor search might enable prediction in a label set agnostic manner, calculating pixel wise nearest neighbor predictions can prove very slow and costly for images with high resolution.

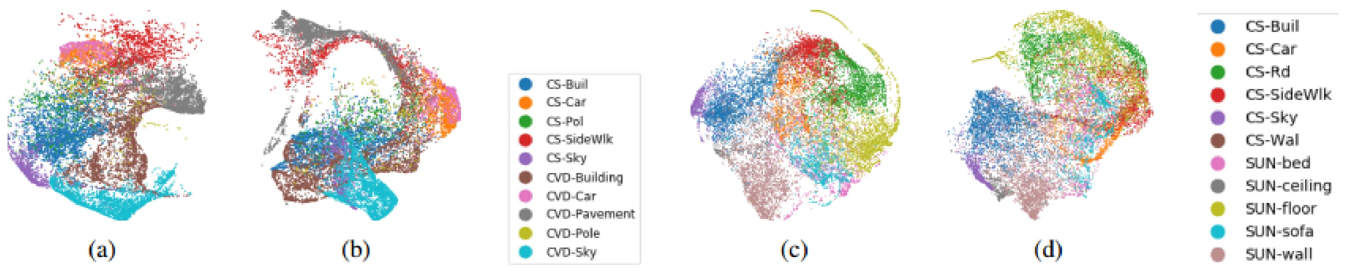

Figure 4: tSNE visualizations of the encoder output representations for majority classes from CS, CVD and SUN datasets. Plots (a) and (b) are for the Univ-basic and Univ-full model from CS-CVD datasets. Observe that the feature embeddings for large classes like *CS:Building-CVD:Building*, *CS:SideWalk-CVD:Pavement*, *CS:Sky-CVD:Sky* align a lot better with universal model. Plots (c) and (d) are for the Univ-basic and Univ-full model from CS-SUN datasets, and labels with similar visual features like *CS:Road - SUN:Floor* show better feature alignment. Best viewed in color and zoom.

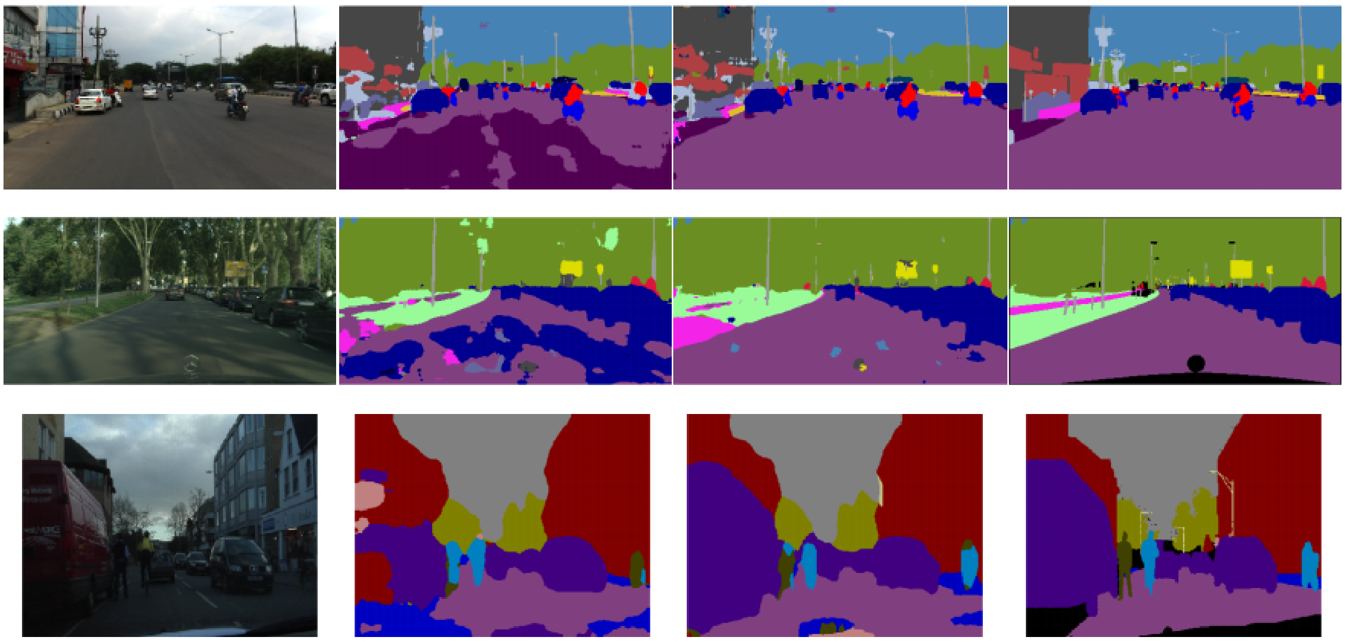
(a) Original Image (b) Without Entropy Module (c) With Entropy Module (d) Ground Truth Segmentation

Figure 5: Qualitative examples from each dataset training with and without the proposed entropy regularization module. The first row shows example from IDD dataset, second row from Cityscapes and the third from the CamVid dataset.