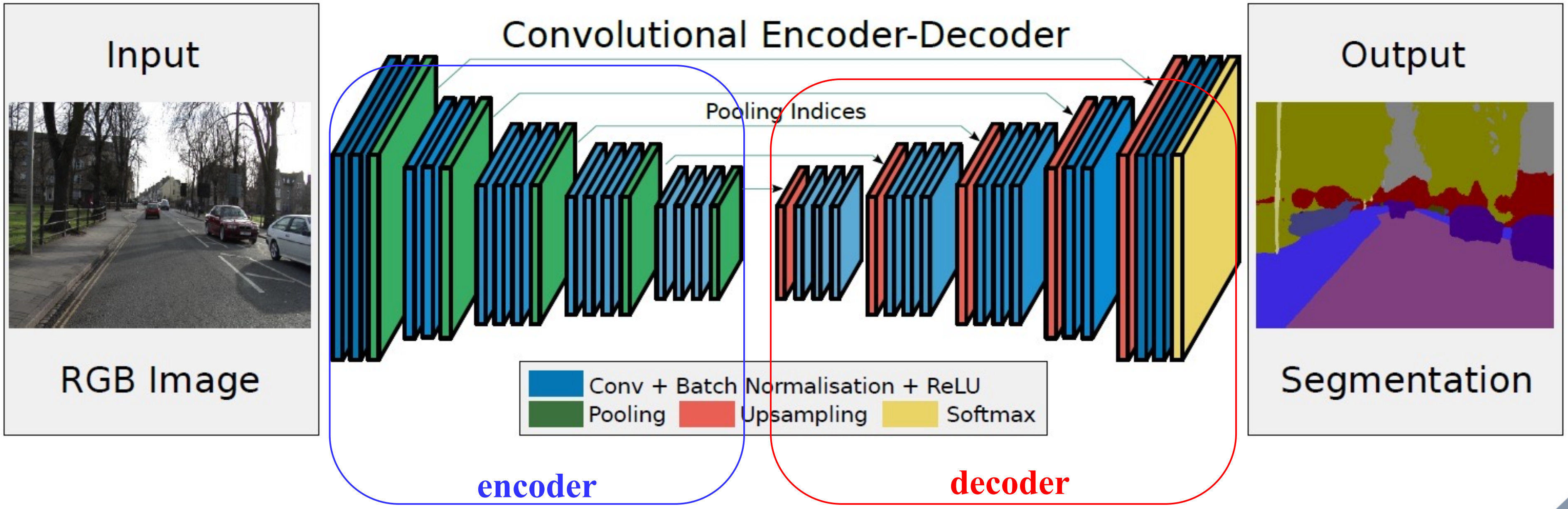
EECS 230 Deep Learning

Lecture 9: Image Segmentation Part II

From last lecture: Fully-supervised Segmentation



Segnet: A deep convolutional encoder-decoder architecture for image segmentation
Badrinarayanan, Kendall, Cipolla – TPAMI 2017

From last lecture: Transpose convolution

❑Bilinear interpolation is a special case

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Kernel

0.25	0.5	0.25
0.5	1	0.5
0.25	0.5	0.25

kernel=3x3
stride=2
padding=1

Output Image

~~0~~	~~0~~	~~0.25~~	~~0.5~~	~~0.75~~	~~1~~	~~1.25~~	~~1.5~~	~~0.75~~
~~0~~	0	0.5	1	1.5	2	2.5	3	~~1.5~~
~~1~~	2	2.5	3	3.5	4	4.5	5	~~2.5~~
~~2~~	4	4.5	5	5.5	6	6.5	7	~~3.5~~
~~3~~	6	6.5	7	7.5	8	8.5	9	~~4.5~~
~~4~~	8	8.5	9	9.5	10	10.5	11	~~5.5~~
~~5~~	10	10.5	11	11.5	12	12.5	13	~~6.5~~
~~6~~	12	12.5	13	13.5	14	14.5	15	~~7.5~~
~~3~~	~~6~~	~~6.25~~	~~6.5~~	~~6.75~~	~~7~~	~~7.25~~	~~7.5~~	~~3.75~~

This lecture

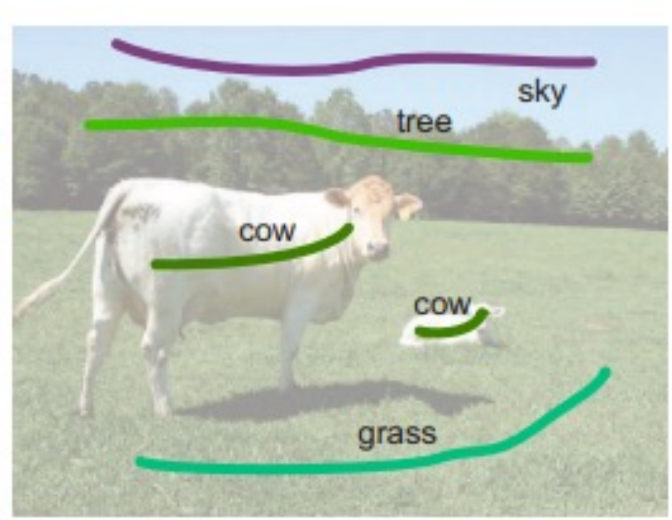
- ❑ Weakly-supervised Semantic Segmentation
  - ❑ Scribble-supervised
  - ❑ Image-tags supervised
- ❑ Contrastive Language-Image Pre-training (CLIP)
- ❑ Open-vocabulary Semantic Segmentation

Weakly-supervised Semantic Segmentation

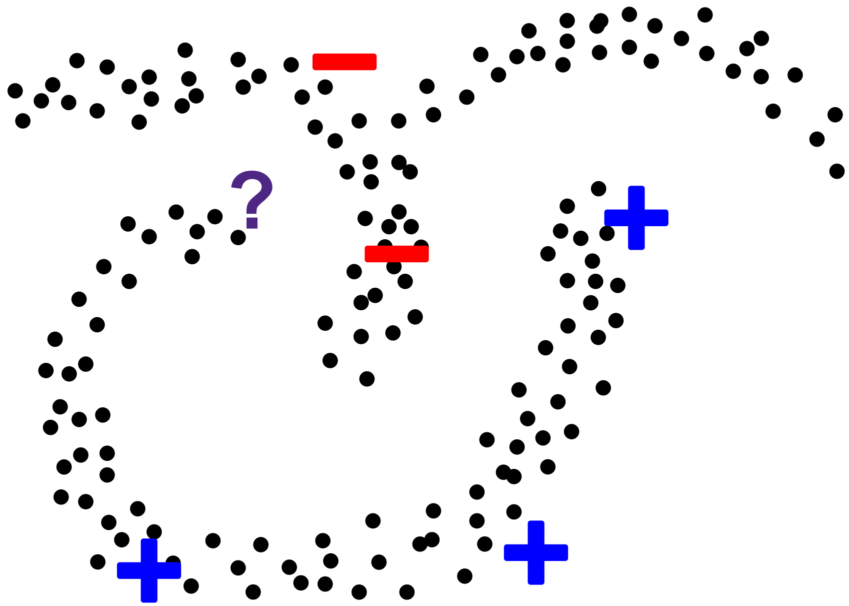
Weakly Supervised Semantic Segmentation

Key Idea

Weakly-supervised segmentation

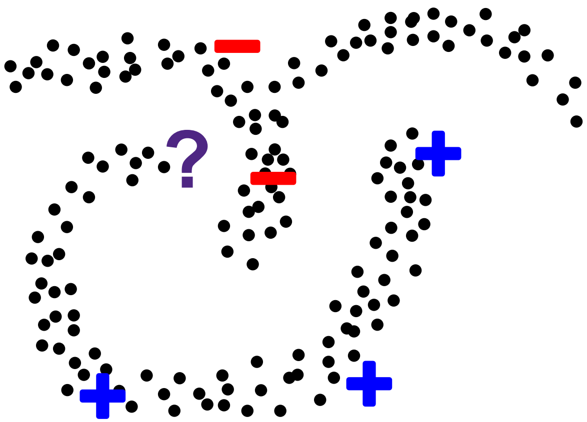


Semi-supervised learning



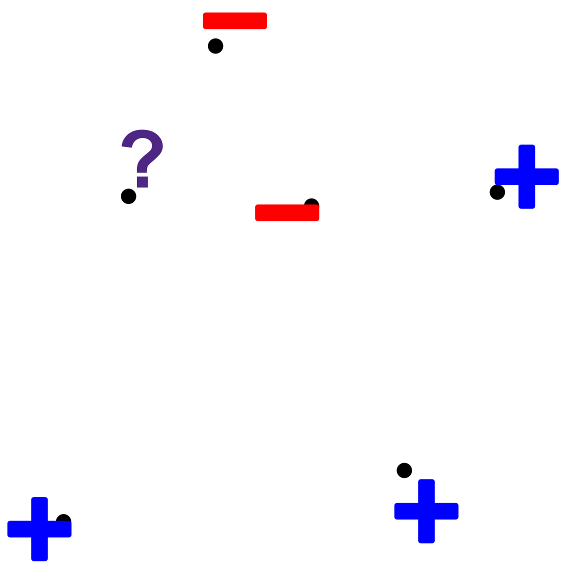
Semi-supervised learning

Definition Given M labeled data $(x_i, y_i) \in (\mathcal{X}, \mathcal{Y})$,$i = 1, ..., M$ and U unlabeled data $x_i, i = M+1, ..., M+U$, learn $f(x) : \mathcal{X} \to \mathcal{Y}$.

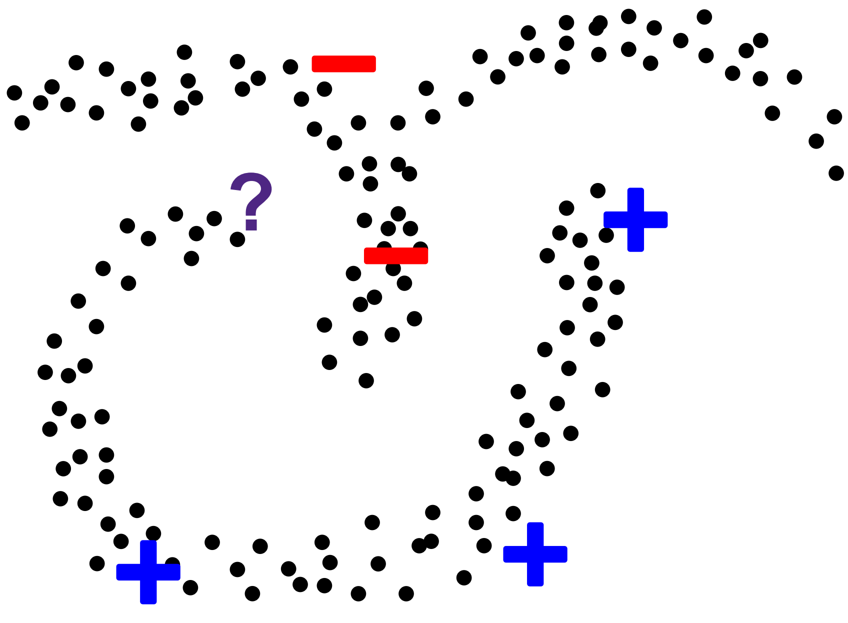


[Zhu & Goldberg, "Introduction to semi-supervised learning", 2009]
[Chapelle, Scholkopf & Zien, "Semi-supervised learning", 2009]

Does unlabeled data matter?



w/o unlabeled data



w/ unlabeled data

Semi-supervised Learning Methods

Self-training

Graph-based Semi-supervised learning

Entropy minimization

Many others…

[Zhu & Goldberg, “Introduction to semi-supervised learning”, 2009]
[Chapelle, Scholkopf & Zien, “Semi-supervised learning”, 2009]

Graph-Based Semi-supervised Learning

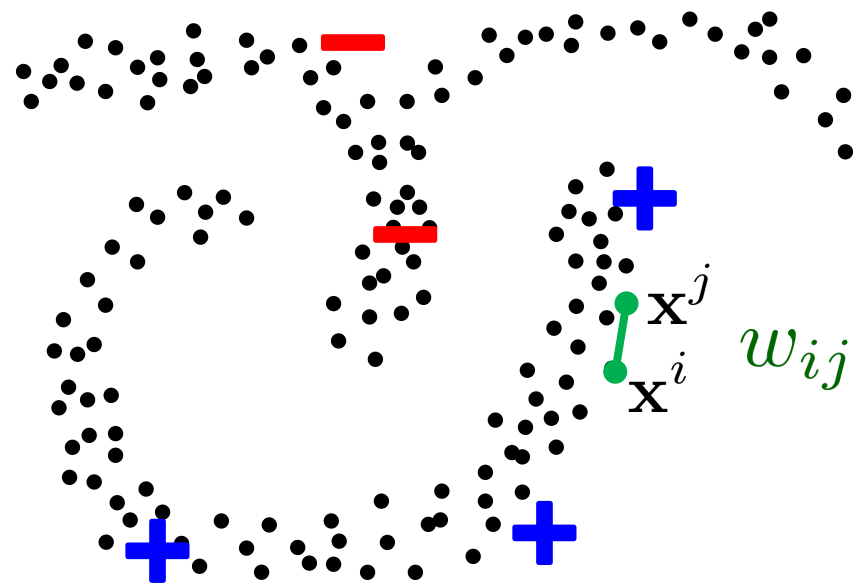
Loss function ?

- labelled points should have **consistency with the target** e.g.

$$\sum_{i=1}^{M} \delta(f(\mathbf{x}^i) \neq \mathbf{y}^i)$$

- unlabeled points should be labeled so that there is some agreement between neighbors i.e. **pairwise regularization**:

$$\sum_{ij \in \mathcal{N}} w_{ij} \; ||f(\mathbf{x}^i) - f(\mathbf{x}^j)||^2$$

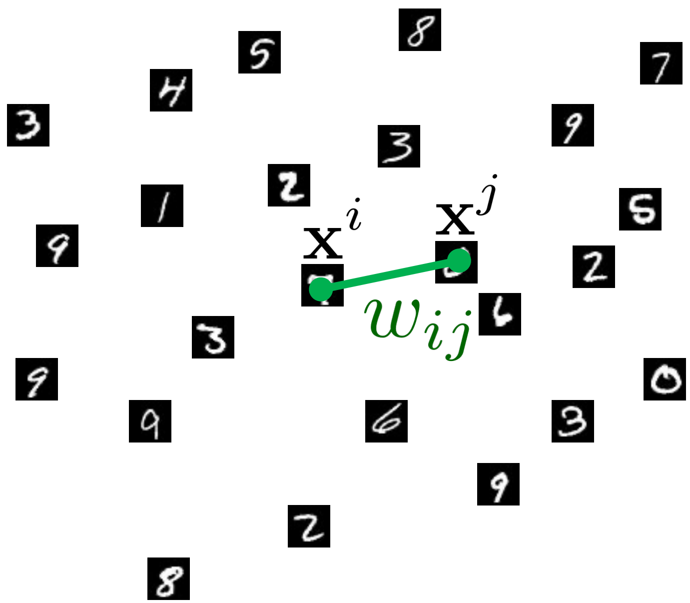


w_{ij} - pre-computed penalty, e.g. based on distance between feature vectors $\mathbf{x}^i$ and $\mathbf{x}^j$

Deep Semi-supervised Learning

Classification

(Weston et al. 2012)



e.g. for **classification CNN** output

$$f(\mathbf{x}^i) = \bar{\sigma}^i \equiv (\bar{\sigma}^i_1, \ldots, \bar{\sigma}^i_K)$$

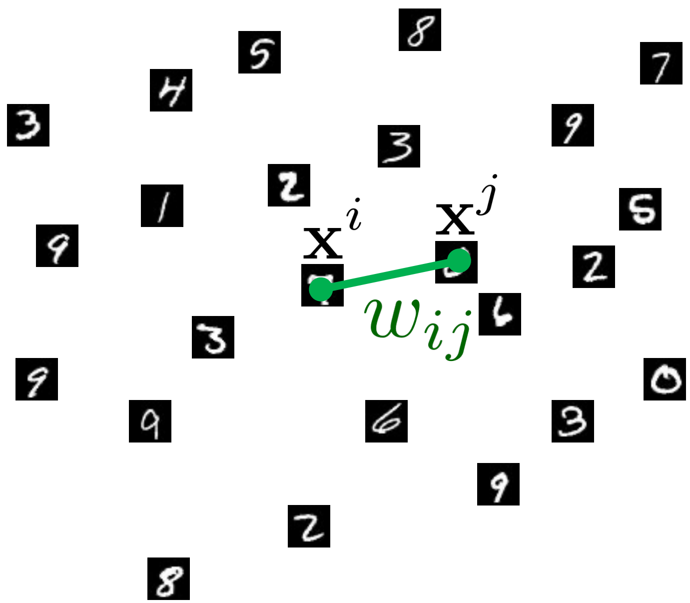
class probabilities at point i

$$\sum_{ij \in \mathcal{N}} w_{ij} \quad ||\bar{\sigma}^i - \bar{\sigma}^j||^2$$

Deep Semi-supervised Learning

Classification

(Weston et al. 2012)



e.g. for **classification CNN** output

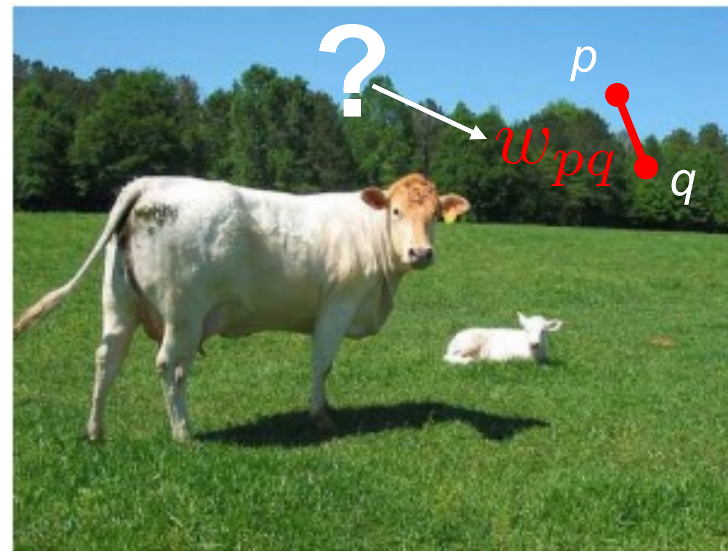
$$f(\mathbf{x}^i) = \bar{\sigma}^i \equiv (\bar{\sigma}^i_1, \ldots, \bar{\sigma}^i_K)$$

class probabilities at point i

$$\sum_{ij \in \mathcal{N}} w_{ij} \quad ||\bar{\sigma}^i - \bar{\sigma}^j||^2$$

Segmentation

(Tang et al. CVPR18, ECCV18)



e.g. for **segmentation CNN** output

$$\bar{\sigma}^p \equiv (\bar{\sigma}^p_1, \ldots, \bar{\sigma}^p_K)$$

class probabilities at pixel p

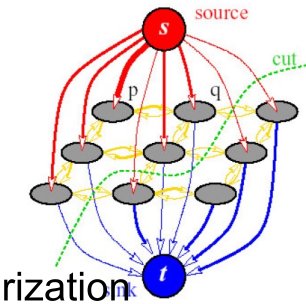
$$\sum_{pq \in \mathcal{N}} w_{pq} \quad ||\bar{\sigma}^p - \bar{\sigma}^q||^2$$

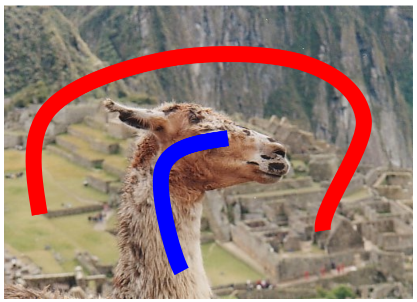
Regularized Loss Functions

We can use regularization ideas from
unsupervised and interactive segmentation
to exploit low-level segmentation cues
(contrast alignment, boundary regularity, regional color consistency, etc.)
for unlabeled parts of an image

low-level segmentation

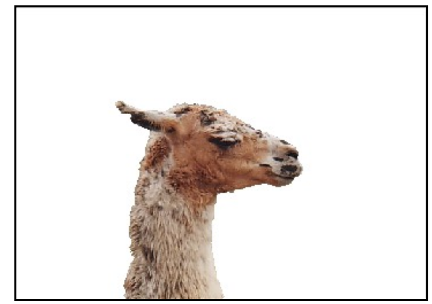
Markov Random Field for Segmentation

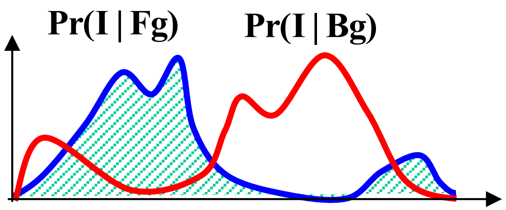




Without Regularization

With Regularization





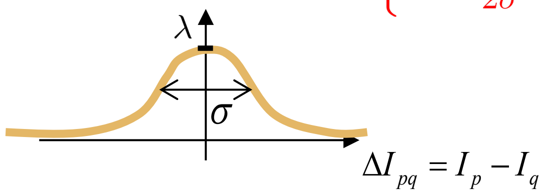
$$E(S,\theta_0,\theta_1) = \sum_{k=0,1}\sum_{p\in S^k} -\ln \mathrm{P}(I_p|\theta_k) + \boxed{\lambda\cdot\sum_{pq\in\mathcal{N}} w_{pq}\cdot[s_p\neq s_q]}$$

MRF regularization

[Boykov, Jolly, *ICCV* 2001]

Regularization energies

$w_{pq} = \lambda \exp\left\{ -\frac{\|I_p - I_q\|^2}{2\sigma^2} \right\}$ - contrast weights w_{pq} from topic 9

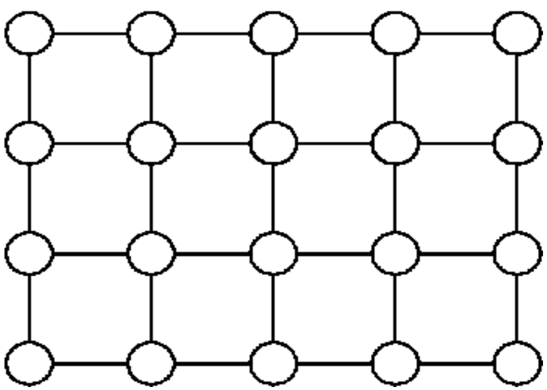


coherence between
discrete labels
at pixels p and q

$$\sum_{pq \in \mathcal{N}} w_{pq} \; [S^p \neq S^q]$$

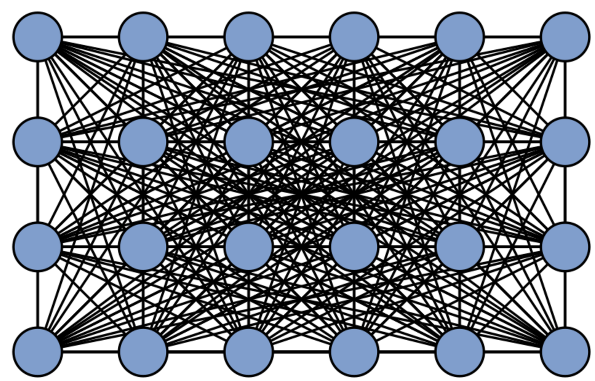
Iverson brackets

Examples of neighborhood systems $\mathcal{N}$ on pixel grid



sparsely connected

[Geman&Giman'81, BVZ PAMI'01, B&J ICCV'01]



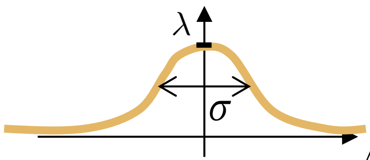
densely connected

[Dense CRF, Krähenbühl & Koltun, NIPS 2011]

Regularization Loss

$$w_{pq} = \lambda \exp\left\{-\frac{\|I_p - I_q\|^2}{2\sigma^2}\right\}$$

- contrast weights w_{pq} from topic 9



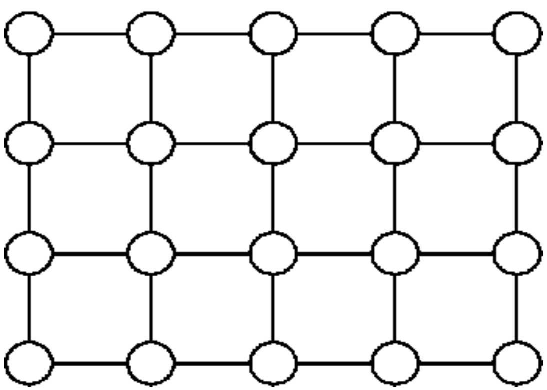
$\Delta I_{pq} = I_p - I_q$

coherence between **probabilistic predictions** at pixels *p* and *q*

$$\sum_{pq \in \mathcal{N}} w_{pq} \; \|\bar{\sigma}^p - \bar{\sigma}^q\|^2$$

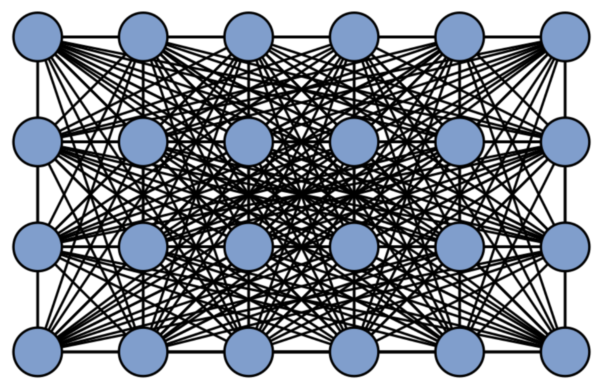
relaxation of Iverson brackets for probabilistic predictions

Examples of neighborhood systems $\mathcal{N}$ on pixel grid



sparsely connected

[Geman&Giman'81, BVZ PAMI'01, B&J ICCV'01]



densely connected

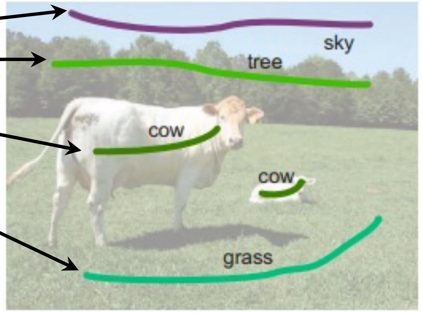
[Dense CRF, Krähenbühl & Koltun, NIPS 2011] weakly-supervised CNN segmentation:

Partial Cross Entropy Loss

cross entropy

over seeds only

$$- \sum_{p \in seeds} \ln \bar{\sigma}^p_{\mathbf{y}^p}$$



$$\bar{\sigma}^p \equiv (\bar{\sigma}^p_1, \ldots, \bar{\sigma}^p_K)$$

predicted "probabilities" for p
to be in each class, e.g. (0,0,..,1,...) in **one-hot** case

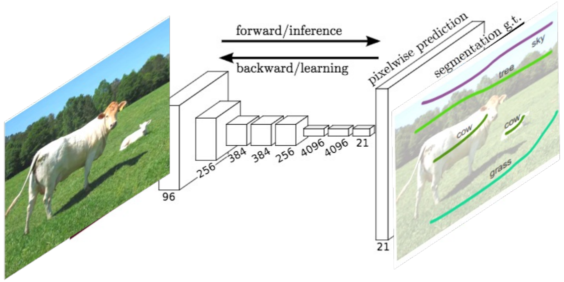
NOTE: if prediction is one-hot
then cross entropy at seed p
is equivalent to $0/\infty$ hard constraint

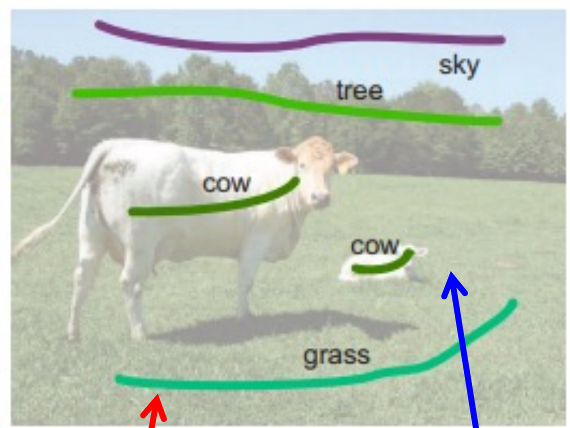
$$\sum_{p \in seeds} \delta(\bar{\sigma}^p \neq \bar{\mathbf{y}}^p)$$

hard constraint
on seed p

weakly-supervised CNN segmentation:

Total **Regularized Loss**





scribbles / seeds

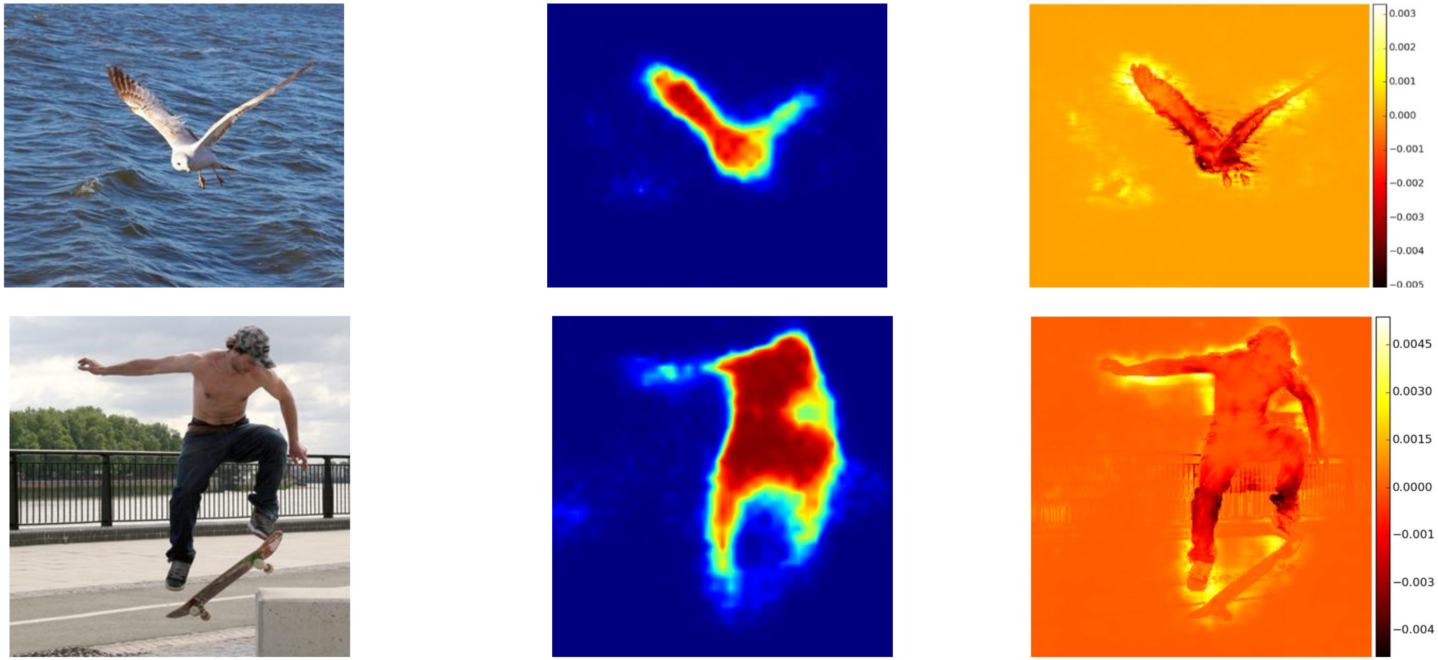
unlabeled pixels

$$L(\bar{\sigma}) = \boxed{-\sum_{p \in seeds} \ln \bar{\sigma}^p_{\mathbf{y}^p}} + \boxed{\sum_{\substack{pq \in \mathcal{N} \\ \text{n-links}}} w_{pq} \; ||\bar{\sigma}^p - \bar{\sigma}^q||^2}$$

Partial Cross Entropy (PCE)

Regularization Loss

Regularization Loss Gradients



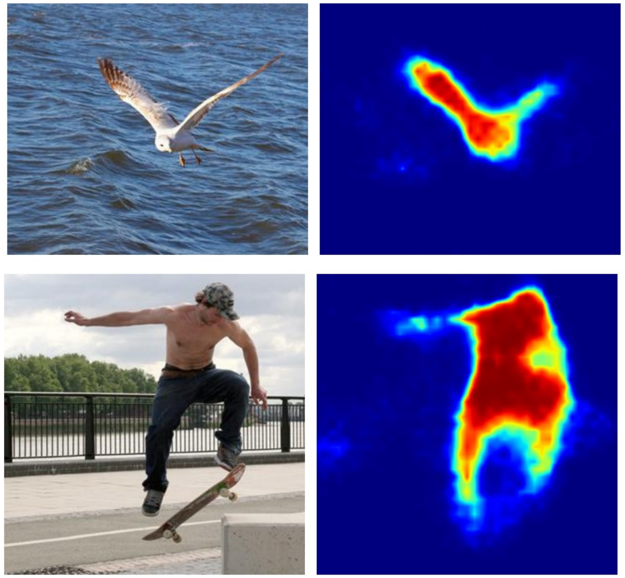
input

network prediction for class k during training $\bar{\sigma}_k^p$

regularization loss gradient $\frac{\partial R(\sigma)}{\partial \sigma_k}$

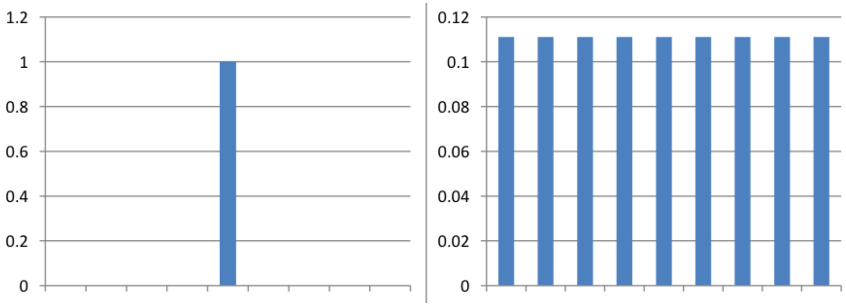
$$R(\sigma) = \sum_{pq \in \mathcal{N}} w_{pq} \cdot ||\bar{\sigma}^p - \bar{\sigma}^q||^2$$

CNN Segmentation may be blurred



Pointwise Entropy Regularization

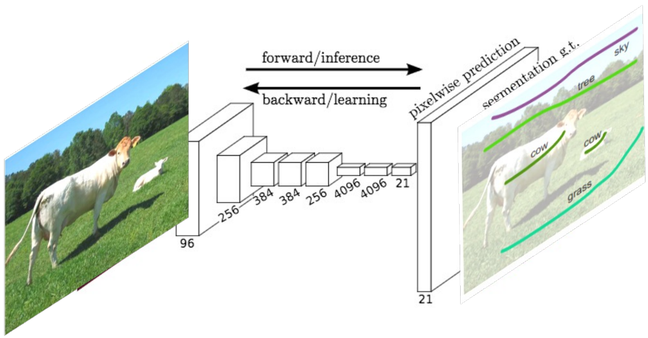
$$\sum_i H(f(x_i))$$

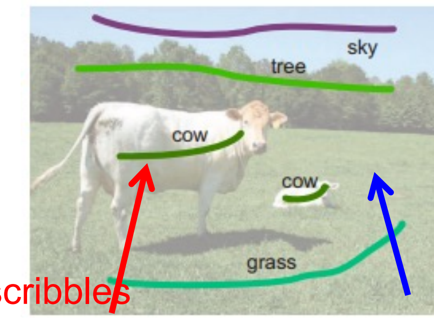


Low entropy High entropy

$$\mathrm{H}(P) = \sum_{k=0}^{K} -P_k \cdot logP_k$$

Regularized loss for weakly-supervised CNN segmentation





scribbles

unknown pixels

empirical risk Loss
for *labeled* data

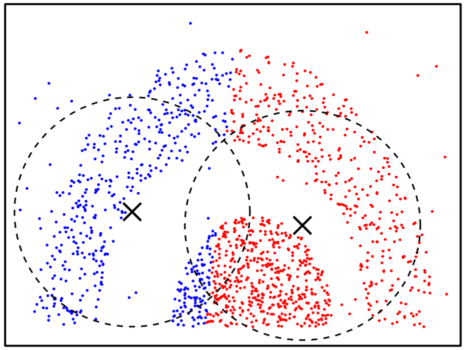
regularization Loss
for *unlabeled* data

$$\boxed{\sum_{i=1}^{M} \ell(f_\theta(x_i), y_i)} \; + \; \lambda \cdot \boxed{R(f)}$$

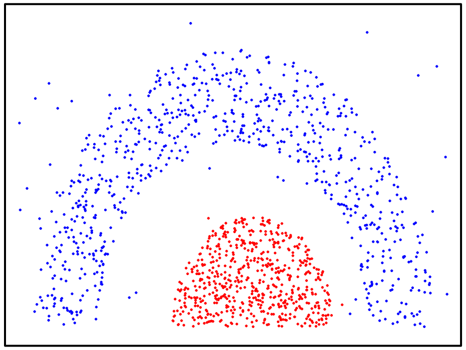
partial Cross Entropy (PCE)

Clustering and Segmentation are Largely Synonym

Linear Clustering



Nonlinear Clustering



Normalized Cut
Segmentation

Kernel K-means

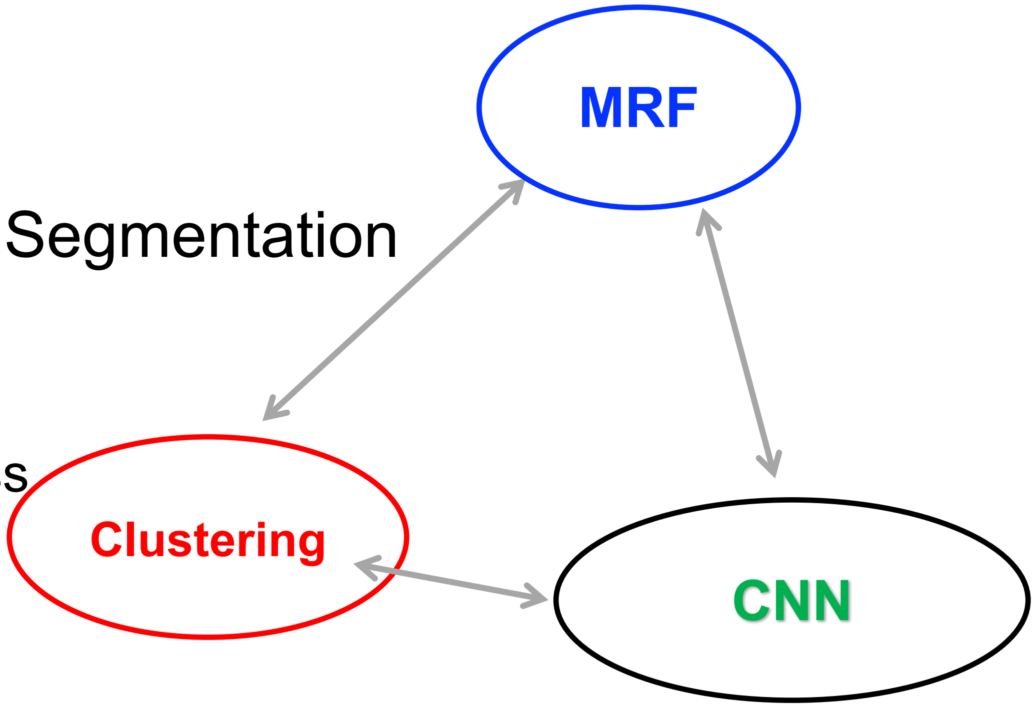
$$\sum_{p \in \mathbf{S}} \|\phi(I_p) - \mu_{\mathbf{S}}\|^2 + \sum_{p \in \overline{\mathbf{S}}} \|\phi(I_p) - \mu_{\overline{\mathbf{S}}}\|^2$$

$$\stackrel{c}{=} \quad -\frac{\sum_{p,q \in \mathbf{S}} k(I_p, I_q)}{|\mathbf{S}|} - \frac{\sum_{p,q \in \bar{\mathbf{S}}} k(I_p, I_q)}{|\bar{\mathbf{S}}|}$$

Regularized Losses

Regularized Loss for CNN Segmentation

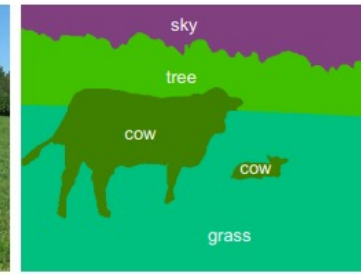
- Pointwise entropy loss
- Pairwise MRF loss
- High-order Clustering loss



Experiments

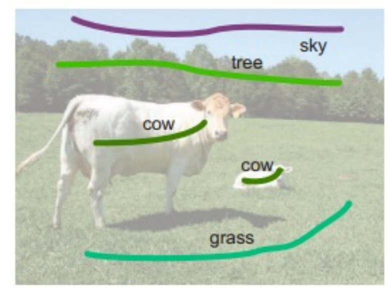
PASCAL VOC 2012 Segmentation Dataset

- 10K training images (full masks)
- 1.5K validation images
- 1.5K test images

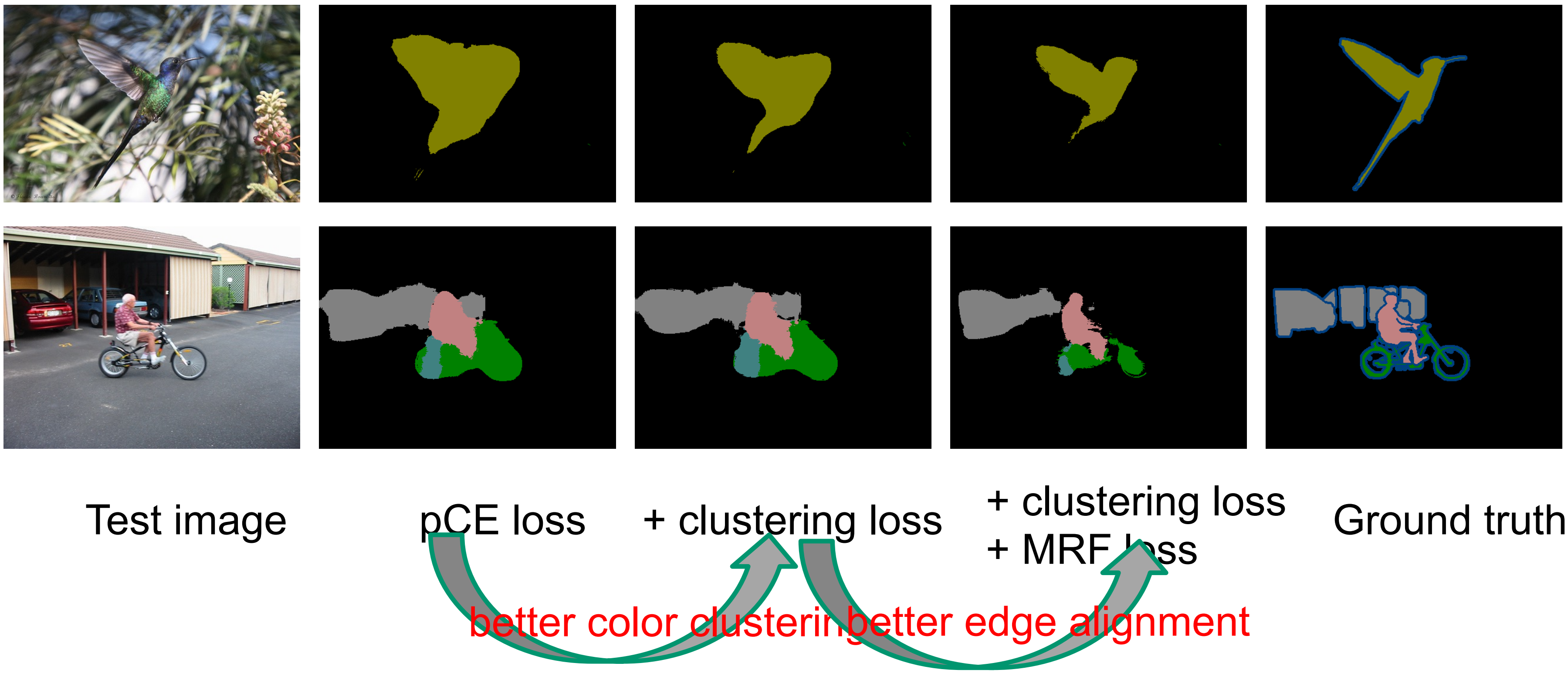


ScribbleSup Dataset [Dai *et al.* ICCV 2015]

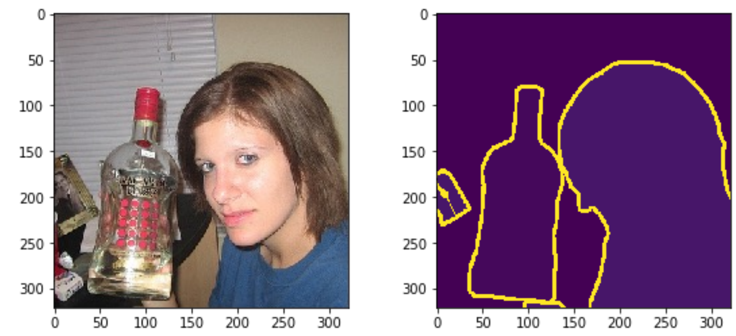
- scribbles for each object
- ~3% of pixels labelled

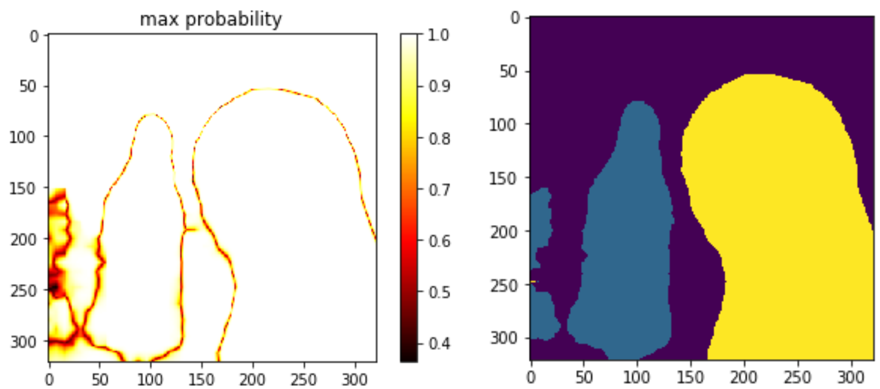


Training with combination of losses

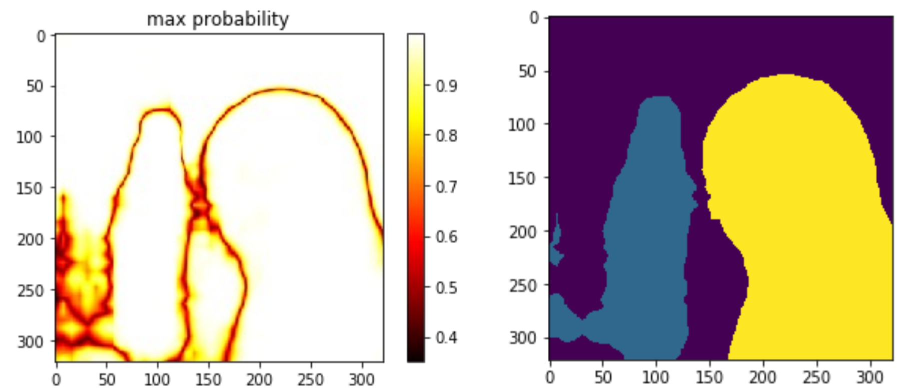


Peakedness of distribution





w/ entropy regularization



w/o entropy regularization

Compare weak and full supervision

Almost as good as full supervision!

network	Full supervision	Weak supervision			
		PCE	PCE+CRF [1]	PCE+ENTROPY	PCE+CRF+ENTROPY
Deeplab2-largeFOV	63.0	55.8	62.2	59.9	**63.0**
Deeplab2-Msc-largeFOV	64.1	56.0	63.1	n/a	**63.5**
Deeplab2-VGG16	68.8	60.4	64.4	63.3	**65.5**
Deeplab2-Resnet101	75.6	69.5	72.9	73.1	**74.4**
Deeplab3+-Resnet101	78.6	71.9	74.6	74.0	**75.6**

PCE: partial cross entropy. CRF: pairwise conditional random field

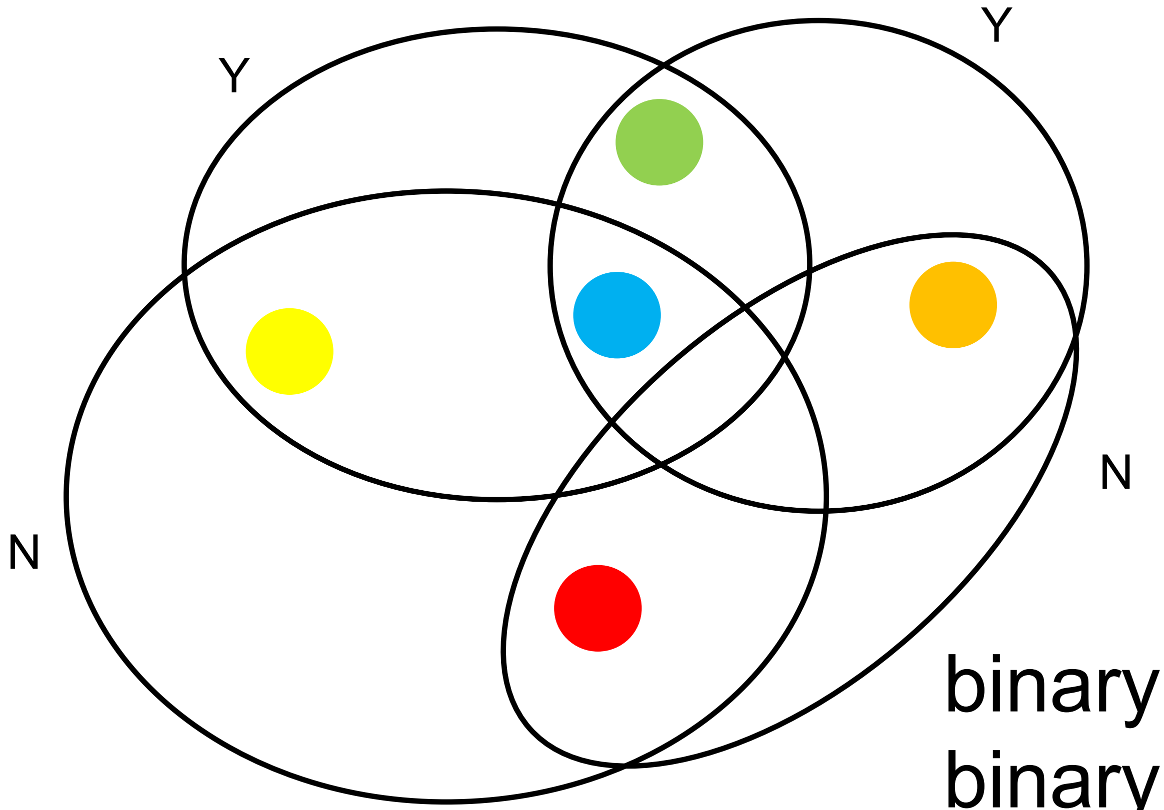
[1] Tang et al., "On Regularized Losses for Weakly-supervised CNN Segmentation", in *ECCV* 2018.

What if **image-level labels only** ?

First, consider a simple related example:

find working molecule (drug discovery)

instead of individual examples, training labels are available only for sets (bags) of examples



binary tags
binary classification

Multiple Instance Learning (MIL)

What if **image-level labels only** ?

For simplicity, assume pixel colors are discriminative enough features.

To segment, we have to learn **what color is sky, grass, and sand** ?

matching **green to grass**, **blue to sky**, and **beige to sand**.

{ sky, grass, sand }

{ sky, sand }

{ grass, sand }

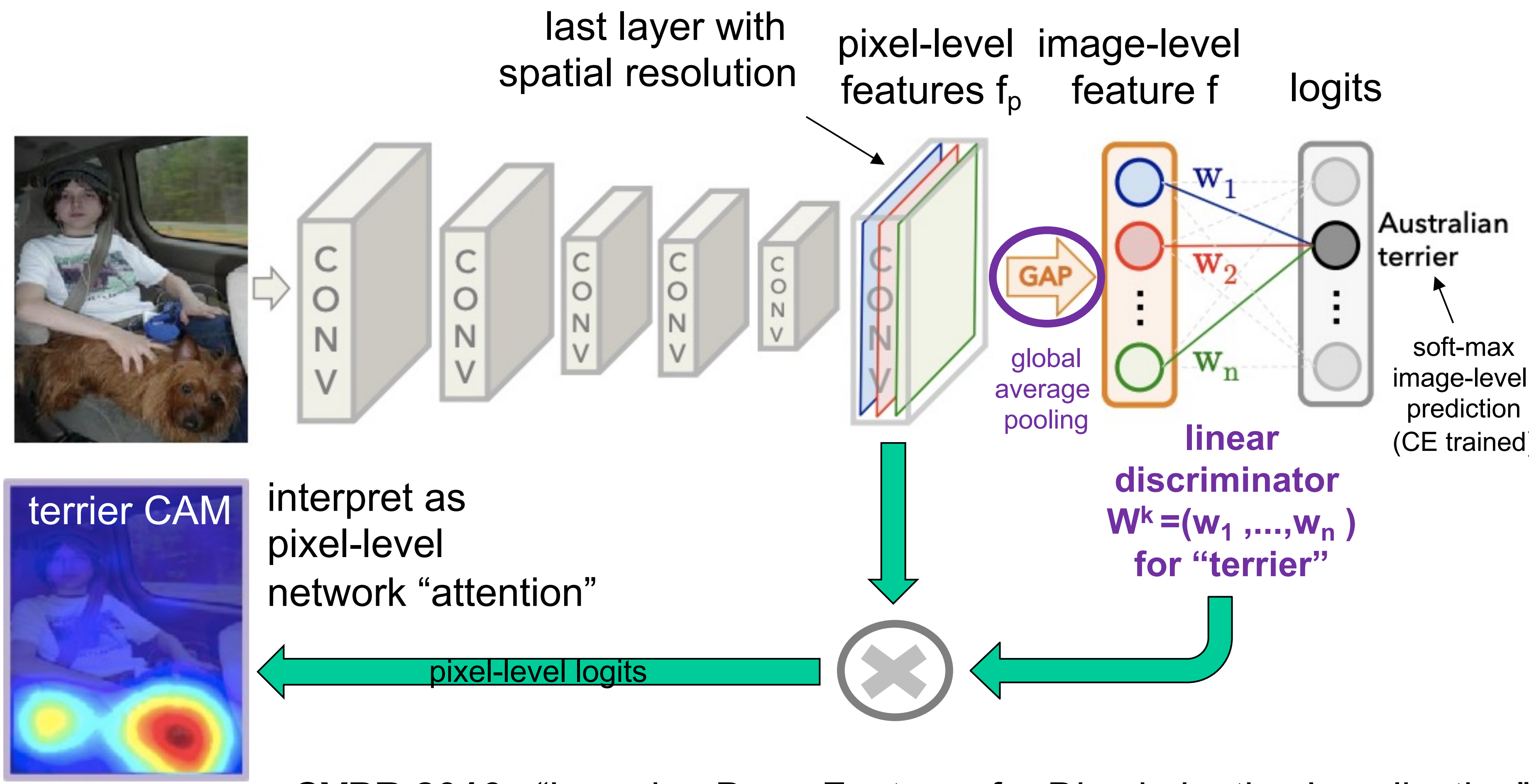
image-level tags

multi-class tags

multi-class classification

How to match pixel to class?

Class-activation Map (CAM)



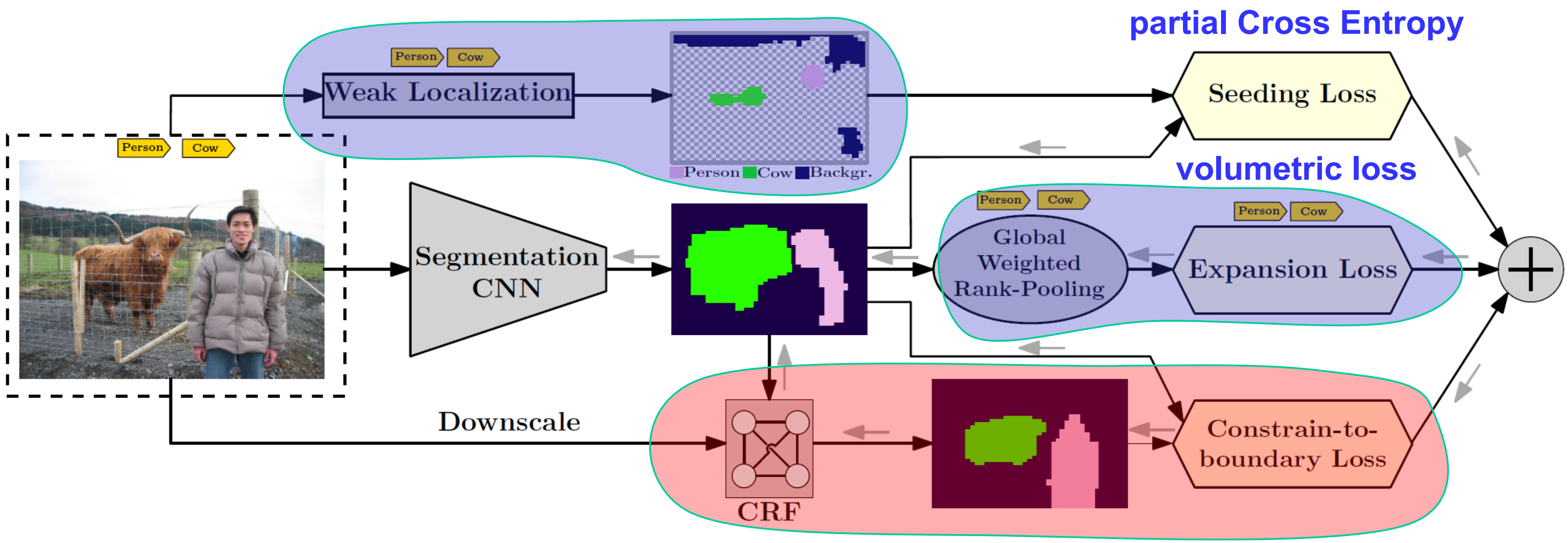
CVPR 2016: "Learning Deep Features for Discriminative Localization"
B.Zhou, A.Khosla, A. Lapedriza, A.Oliva, A.Torralba

NOTE: motivates ideas for **object localization**, as well as **image-level supervision for semantic segmentation**

What if **image-level labels only** ?

Some ideas: [Kolesnikov & Lampert ECCV 2016]

seeds from "network attention"
see CAM at the end of Topic 10



Can be simplified using
regularization loss
in the previous slides

Contrastive Language-Image Pre-training (CLIP)

What Is CLIP and What Can CLIP Do?

- CLIP stands for Contrastive Language–Image Pre-training.
- It is a network that can be directly used for **image classification**.
- It is suitable for **zero-shot learning**. This network does not require fine-tuning when predicting labels on new images.
- The classification accuracy is **more robust** across a wide range of image datasets. This is crucial because well-trained models sometimes perform poorly during the real-world deployment.

How to train CLIP?

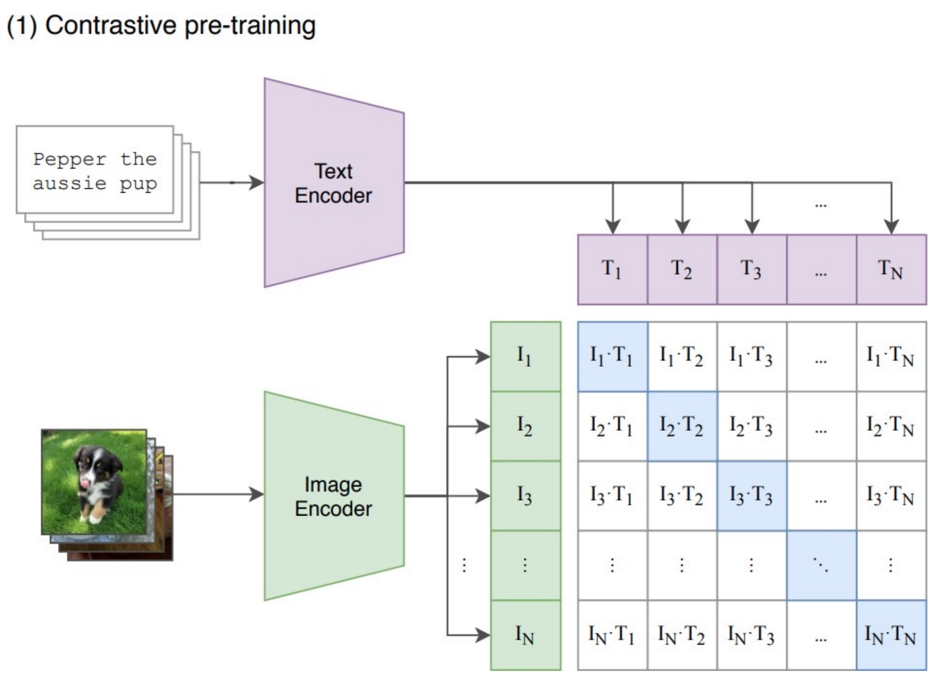


Figure: Contrastive Pre-training of language-image pairs. The text encoder is a standard transformer encoder. The extracted feature is the embedding of the CLS token. The image encoder is either a ResNet-50 or a Vision Transformer (ViT).

How to train CLIP for classification?

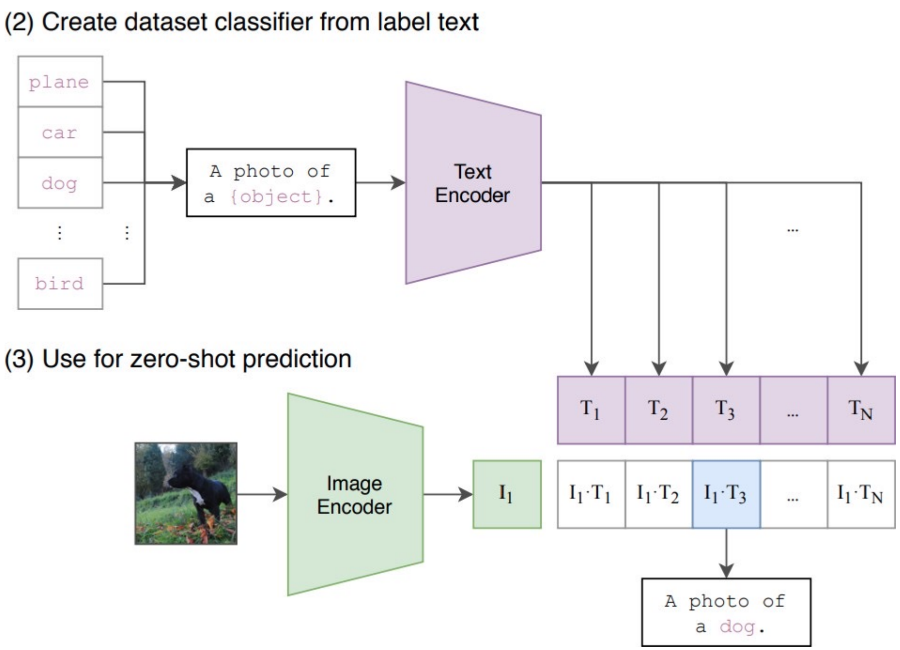


Figure: At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes. This prediction setup is already very interesting because we don't need to finetune or train a top-layer classifier. As long as we include the correct label in our prediction option, this framework can perform the classification task.

How to Use CLIP for Classification?

FOOD101

guacamole (90.1%) Ranked 1 out of 101 labels

✓ a photo of **guacamole**, a type of food.

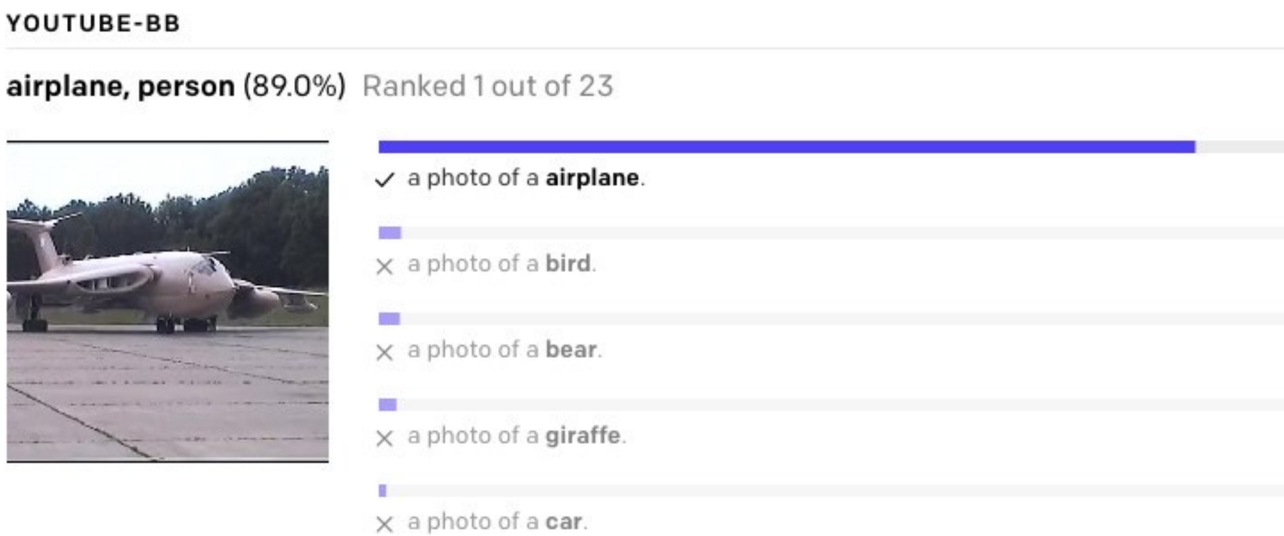
× a photo of **ceviche**, a type of food.

× a photo of **edamame**, a type of food.

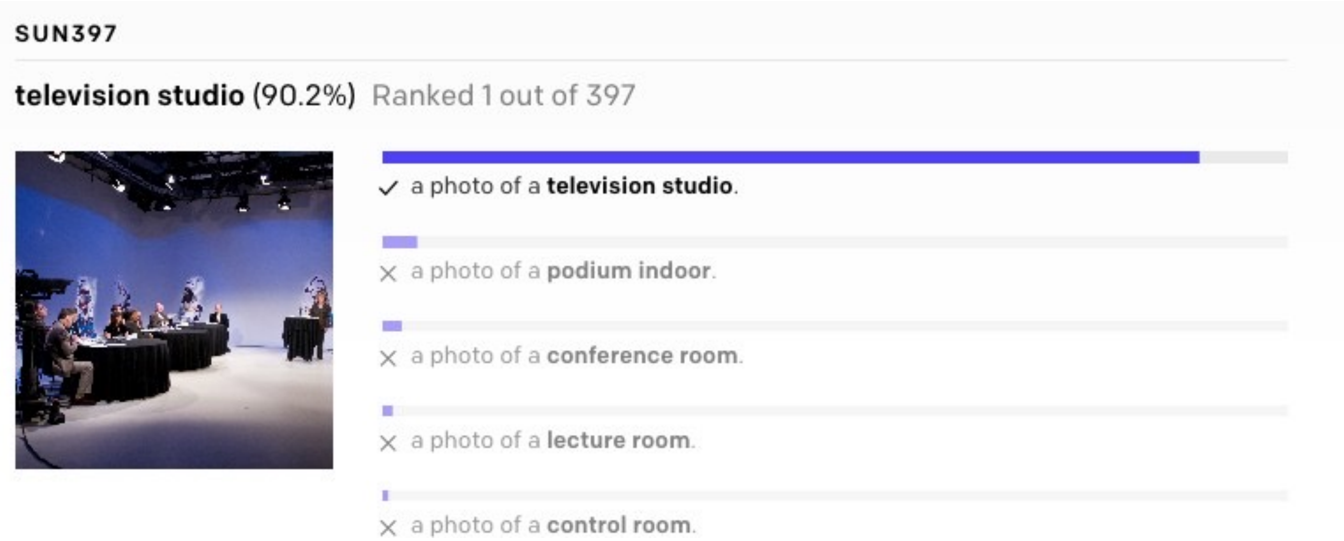
× a photo of **tuna tartare**, a type of food.

× a photo of **hummus**, a type of food.

How to Use CLIP for Classification?



How to Use CLIP for Classification?



How to Use CLIP for Classification?

EUROSAT

annual crop land (12.9%) Ranked 4 out of 10

× a centered satellite photo of **permanent crop land**.

× a centered satellite photo of **pasture land**.

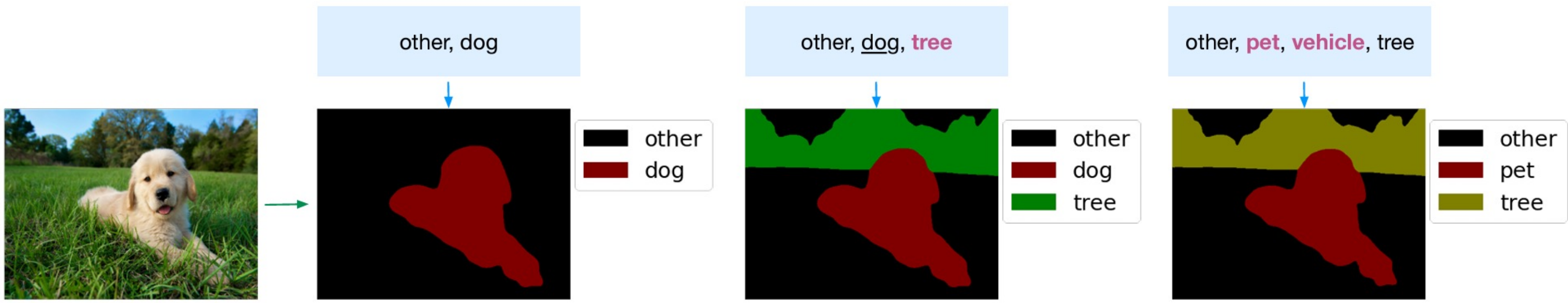
× a centered satellite photo of **highway or road**.

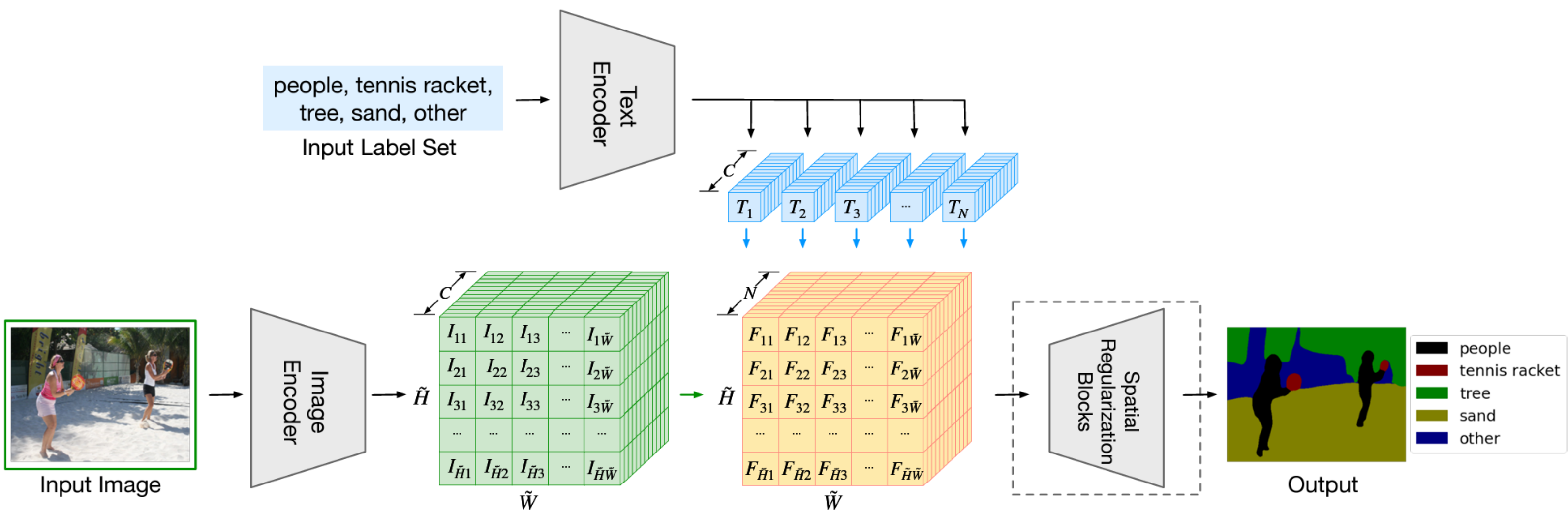
✓ a centered satellite photo of **annual crop land**.

× a centered satellite photo of **brushland or shrubland**.

Zero-shot Semantic Segmetnation

- ❑Language driven semantic segmentation





https://arxiv.org/pdf/2201.03546.pdf