Regularizing Deep Networks by Modeling and Predicting Label Structure

Mohammadreza Mostajabi Michael Maire Gregory Shakhnarovich

Toyota Technological Institute at Chicago

CVPR 2018 · Salt Lake City · June 18-22

Overview

Contribution:

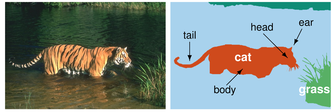
- New training procedure for convolutional neural networks (CNNs)
  - Phase 1: Learn a model of the label (output) space
  - Phase 2: Attach model as regularizer during standard training
- Regularizer forces network to learn high-level abstractions
- Zero computational cost at test time (unlike [2]); **free performance increase**

Experiments:

- Accuracy boost on PASCAL semantic segmentation
- Useful alone as well as when combined with pretraining and/or data augmentation

Intuition:

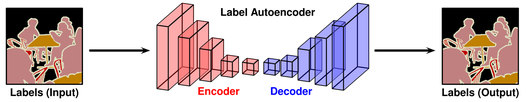
- Detailed annotation contains informative structure beyond that explicitly named (white labels)
- Region shape → parts (black)
- Object co-occurrence
- Extract and use such properties!



Training Algorithm

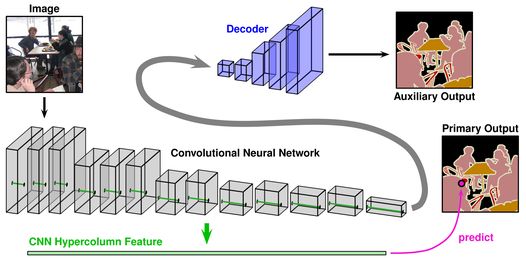
Training Phase 1:

- Ignore actual images
- Build convolutional autoencoder model of the ground-truth annotation
- Capacity constraints force model to compress label representation



Training Phase 2:

- Attach decoder from phase 1 to a standard CNN, to act as a regularizer
- Primary prediction pathway: hypercolumn feature per pixel
- Auxiliary prediction pathway: generate representation to decode
- Backpropagate from losses on both pathways; decoder parameters are frozen



After Training: Discard decoder and auxiliary output pathway

Semantic Segmentation Results

Qualitative:

- Corrects large-scale global labeling errors with respect to baseline CNN
- Contrast to CRF-like post-processing, which primarily refines boundaries

Quantitative:

- Large accuracy boost (>10% relative) over baseline when training from scratch
- Additional improvement atop ImageNet pretraining + data augmentation
- Gains consistent over many choices of CNN architecture

Architecture	Data-Aug?	Auxiliary Regularizer	mIoU
VGG-16-hypercolumn	no	none	37.3
	no	Encoder (conv1 & conv3)†	41.1
	no	**Decoder (32 channel)†**	**42.4**
	yes	none	55.2
	yes	**Decoder (128 channel)**	**57.1**
VGG-16-FCN8s	yes	none	51.5
	yes	**Decoder (128 channel)**	**54.1**
DenseNet-67-hypercolumn	no	none	40.5
	no	Encoder (conv1 & conv3)†	44.0
	no	**Decoder (32 channel)†**	**45.2**
	no	Decoder (128 channel)	42.5
	yes	none	58.8
	yes	Decoder (32 channel)	59.4
	yes	**Decoder (128 channel)**	**60.6**
	yes	Decoder (256 channel)	59.8

PASCAL mIoU without ImageNet pretraining

Architecture	Data-Aug?	Auxiliary Regularizer	mIoU
VGG-16-hypercolumn	no	none	67.1
	no	**Decoder (32 channel)**	**68.8**
DenseNet-121-hypercolumn	yes	none	71.6
	yes	**Decoder (128 channel)**	**71.9**
ResNet-101-PSPNet	yes	none	75.4
	yes	**Decoder (128 channel)**	**75.9**

PASCAL mIoU with ImageNet pretraining

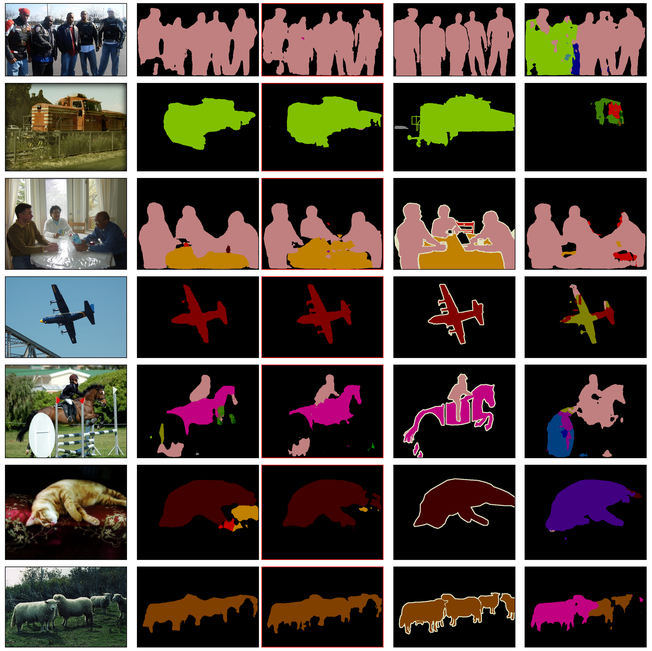


Image | Auxiliary Output | Primary Output | Ground-truth | Baseline DenseNet-67

Ours: DenseNet-67 trained with regularizer

Label Model Introspection

Does the learned label-space representation capture parts?

Visualize nearest neighbors in the learned label space:

- Pick query location (green dot) in image
- Compute label-space representation, either:
  - Encode corresponding label map (left examples)
    OR
  - Run CNN; extract decoder input (right examples)
- Retrieve regions with similar representations (search dataset built using encoder on label maps)

Commonalities with poselets [1]:

- Both provide supplementary training targets
- Both capture part-like structure

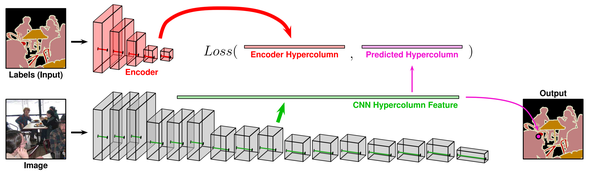
Query | Neighbors

Labels → Encoder → Matches

Query | Neighbors

Image → CNN → Matches

Alternative Regularization Approach



Encoder-based Phase 2:

- Encoder output ↔ decoder input
- Directly predict encoder representation instead of decoder input
- Train with an auxiliary regression problem (MSE loss); freeze encoder parameters
- Works well, but not quite as well as using the decoder pathway

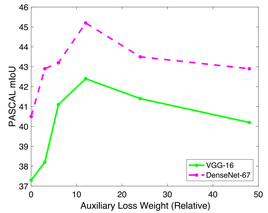
Advantages of Decoder-based Regularizer:

- Decoder output is visually interpretable
- Cross-entropy loss for decoder output mirrors usual semantic segmentation loss

Ablation Experiments

Auxiliary Loss Weighting

- Beyond autoencoder architecture, applying our regularizer involves choice of one additional hyperparameter
- Can choose relative weight of auxiliary branch loss with respect to primary loss
- Robust to this parameter: wide range of weights on the auxiliary loss improves accuracy over the baseline; magnitude of improvement also stable



Decoder Effectiveness

- Regularization effect deteriorates if decoder parameters not held fixed
- Performance gains due to label-space model, not dual output pathway architecture

Architecture	Data-Aug?	Auxiliary Regularizer	mIoU
DenseNet-67-hypercolumn	yes	none	58.8
	yes	**Decoder (128 channel)**	**60.6**
	yes	Unfrozen Decoder	60.2
	yes	Random Init. Decoder	58.8

PASCAL mIoU with different decoder initialization and update schemes

References

[1] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. *ICCV*, 2009.

[2] S. Xie, X. Huang, and Z. Tu. Top-down learning for structured labeling with convolutional pseudoprior. *ECCV*, 2016.