Learning Co-Visibility Networks for Nuisance-Invariant Image Comparison & Occlusion Detection

Nikolaos Karianakis

UCLA Computer Vision Lab

July 11, 2013

Intrinsic Variability vs. Nuisance Variability

• The problem of nuisance variability is very acute in Computer Vision, where even the same object can yield a large variety of images depending on vantage point, illumination, partial occlusion etc.

• Much of the ventral stream is tasked with managing the infinite amount of nuisance variability.
T. Poggio. "How the ventral stream should work", 2011.

• The intrinsic variability of objects in images is infinitesimal compared to nuisance variability.
G. Sundaramoorthi, P. Petersen, V. S. Varadarajan & S. Soatto. "On the set of images modulo viewpoint and contrast changes", 2009.

• Deep learning architectures have shown the ability to learn class-specific variability despite significant nuisance variability.

Nuisance Invariant Deep Learning Architecture

• We are interested in testing the hypothesis that a deep learning architecture is able to *train away* nuisance variability, which is preset in images, owing to changes of viewpoint and illumination, noise, defects etc.

• We choose the simplest visual classification task with no intrinsic variability; that is *occlusion detection*, which is the binary classification task of determining the co-visiblity from different images (e.g. two sequential video frames) of the same underlying scene.

Empirical Conclusions in Occlusion Detection

• Our architecture based on the Gated Restricted Boltzmann Machine can eliminate a notable amount of nuisance variability.

• Our method satisfactorily recognizes the similarity (or dissimilarity, which means occlusion in this setting) between images from the same underlying scene.

• However, in general, we cannot outperform algorithms specifically *engineered* for occlusion detection yet.

The Model

A *similarity measure* among two images x, y and a hidden layer h is defined as:

$$S(\mathbf{x},\mathbf{y},\mathbf{h}) = \sum_{f=1}^{F}((\sum_{i=1}^{I} u_{if}x_i)(\sum_{j=1}^{J} v_{jf}y_j)(\sum_{k=1}^{K} w_{kf}h_k))$$

After adding a generalization and two offset elimination terms, the energy is:

$$E(\mathbf{x},\mathbf{y},\mathbf{h};\theta) = -S(\mathbf{x},\mathbf{y},\mathbf{h}) + 1/2 * \sum_{i=1}^{I}(x_i - a_i)^2 + 1/2 * \sum_{j=1}^{J}(y_j - b_j)^2 - \sum_{k=1}^{K} c_k h_k$$

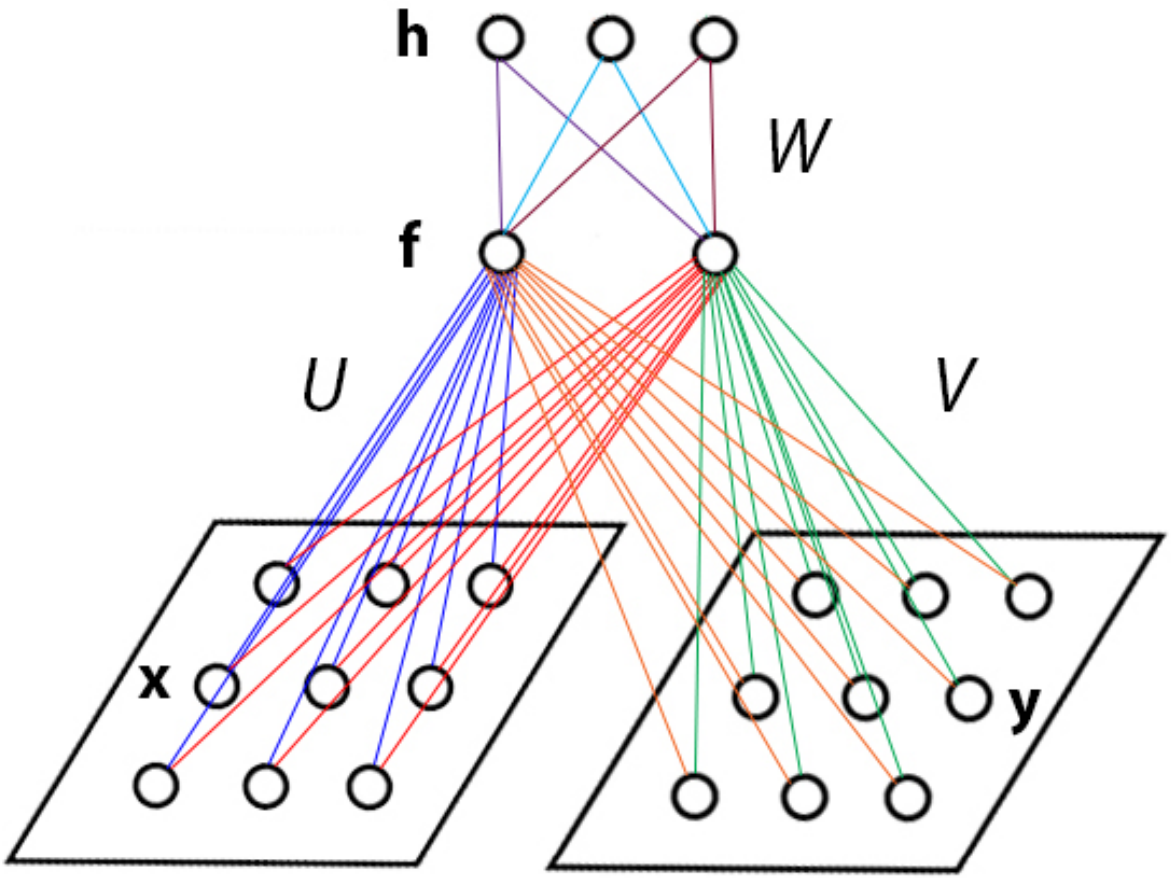
The joint probability over the triplets $(\mathbf{x},\mathbf{y},\mathbf{h})$:

$$P_\theta(\mathbf{x},\mathbf{y},\mathbf{h}) = \frac{e^{-E(\mathbf{x},\mathbf{y},\mathbf{h};\theta)}}{Z(\theta)}$$

where $\theta = \{\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{a},\mathbf{b},\mathbf{c}\}$ are the model parameters.

J. Susskind, G. E. Hinton, R. Memisevic and M. Pollefeys. "Modeling the joint density of two images under a variety of transformations", 2011.

Gated Restricted Boltzmann Machine



Conditional Distributions

Exploiting the conditional independence, the conditionals are the following:

$$P(\mathbf{x}|\mathbf{y},\mathbf{h}) = \prod_{i=1}^{I} \mathcal{N}(a_i + \sum_{f=1}^{F} u_{if}(\sum_{j=1}^{J} v_{jf} y_j)(\sum_{k=1}^{K} w_{kf} h_k); 1.0) \tag{1}$$

$$P(\mathbf{y}|\mathbf{x},\mathbf{h}) = \prod_{j=1}^{J} \mathcal{N}(b_j + \sum_{f=1}^{F} v_{jf}(\sum_{i=1}^{I} u_{if} x_i)(\sum_{k=1}^{K} w_{kf} h_k); 1.0) \tag{2}$$

$$P(\mathbf{h}|\mathbf{x},\mathbf{y}) = \prod_{k=1}^{K} ber(c_k + \sum_{f=1}^{F} w_{kf}(\sum_{i=1}^{I} u_{if} x_i)(\sum_{j=1}^{J} v_{jf} y_j)) \tag{3}$$

where $\mathcal{N}$ and *ber* stand for the Gaussian (Normal) and Bernoulli distributions, correspondingly.

Unsupervised Maximum Likelihood Learning

Given a set of *i.i.d.* training examples $D = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}), n = 1, \ldots, N\}$, we have to learn the model parameters θ. Our strategy is to maximize the penalized log-likelihood:

$$L(\theta) = \frac{1}{N}\sum_{n=1}^{N} logP(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}) - \frac{\lambda}{N}(\|\mathbf{U}\|^2_{Frob} + \|\mathbf{V}\|^2_{Frob} + \|\mathbf{W}\|^2_{Frob})$$

$$\frac{\partial L(\theta)}{\partial U_{if}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial}{\partial U_{if}} log(\sum_{\mathbf{h}} e^{-E(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{h}; \theta)}) - \frac{\partial logZ(\theta)}{\partial U_{if}} - \frac{2\lambda U_{if}}{N}$$

$$= -\frac{1}{N}\sum_{n=1}^{N} \left\langle \frac{\partial E(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \mathbf{h}; \theta)}{\partial U_{if}} \right\rangle_{\mathbf{h}} + \left\langle \frac{\partial E(\mathbf{x}, \mathbf{y}, \mathbf{h}; \theta)}{\partial U_{if}} \right\rangle_{\mathbf{x},\mathbf{y},\mathbf{h}}$$

The second term is *intractable*, as it is computed over all the exponentially many configurations $(\mathbf{x}, \mathbf{y}, \mathbf{h})$. The average can be approximated using *Gibbs* sampling.

3-way Contrastive Divergence

• Following the approach proposed in Susskind et al., samples are drawn alternatingly from the conditional distributions $P(\mathbf{h}|\mathbf{x},\mathbf{y})$, $P(\mathbf{x}|\mathbf{h},\mathbf{y})$ and $P(\mathbf{y}|\mathbf{h},\mathbf{x})$ in random order and the process is terminated before the equilibrium distribution.

• Given the tri-partite structure of the model, the learning process can be characterized as *3-way Contrastive Divergence*.

Occlusion Testing

• The model is trained with pairs of images from the same underlying scene, taken under different illumination and vantage point – yielding (self-) occlusions.

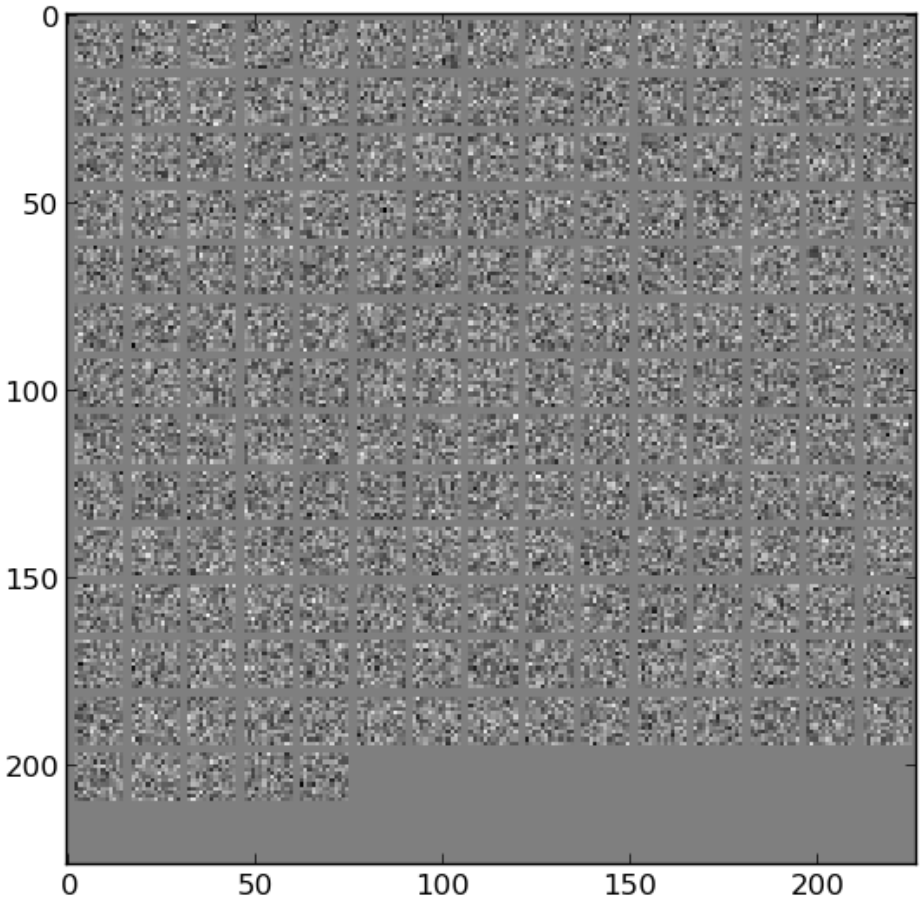
• The log-likelihood as a score to quantify co-visibility in a new pair is problematic:

$$logP(\mathbf{x},\mathbf{y}) = -logZ + \sum_{k=1}^{K} log(1 + exp(c_k + \sum_{f=1}^{F} w_{kf}(\sum_{i=1}^{I} u_{if}x_i)(\sum_{j=1}^{J} v_{jf}y_j))$$
$$- 1/2 * \sum_{i=1}^{I}(x_i - a_i)^2 - 1/2 * \sum_{j=1}^{J}(y_j - b_j)^2$$

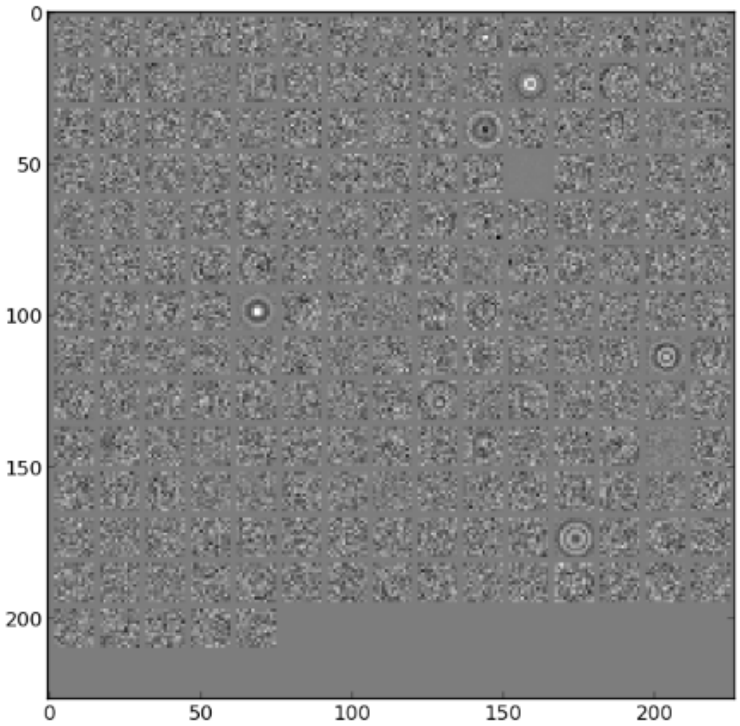
• In order to eliminate the demanding normalizing term $logZ$ and avoid the inconsistent rescaling in single images, the following metric is used:

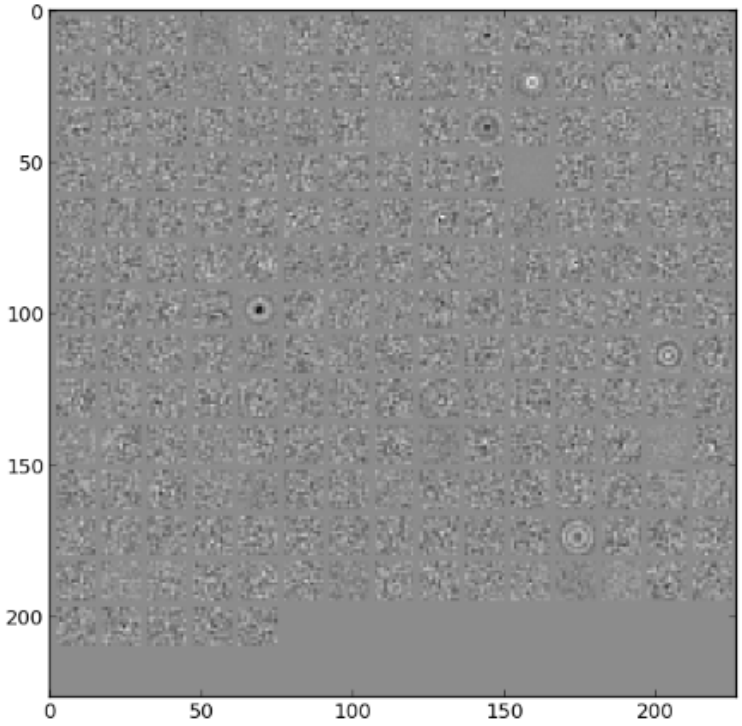
$$d(\mathbf{x},\mathbf{y}) = -logP(\mathbf{x},\mathbf{y}) - logP(\mathbf{y},\mathbf{x}) + logP(\mathbf{x},\mathbf{x}) + logP(\mathbf{y},\mathbf{y})$$

Shifted Filters



Rotated Filters





Naive Algorithm

Our Algorithm

Precision-Recall Comparison Statistics

	Venus	RubberWhale	Hydrangea	Grove2	Grove3
R [1]	0.66	0.20	0.20	0.55	0.45
P [1]	0.61	0.46	0.68	0.72	0.79
P [2]	0.69	0.91	0.96	0.96	0.86
P (ours)	0.64	0.93	0.95	0.78	0.83

[1] V. Kolmogorov and R. Zabih. "Computing Visual Correspondence with Occlusions via Graph Cuts", *ICCV01*.

[2] A. Ayvaci, M. Raptis and S. Soatto. "Occlusion Detection and Motion Estimation with Convex Optimization", *NIPS10*.

F1-Score Statistics on Middlebury Dataset

	Venus	RubberWhale	Hydrangea	Grove2	Grove3
F1-Score	0.83	0.80	0.81	0.70	0.86

Open questions

• Can a deep learning network outperform algorithms specifically engineered to detect occlusions or perform other Computer Vision tasks?

• Can a deep learning network be that effective when significant intrinsic variability competes with nuisance variability, such as in the detection, localization, and recognition tasks of object classes in multiple images?

• Can an activity recognition algorithm get more precise when "canonizing" the training set by leveraging a deep learning framework?

Future Work

- Using Self Diffusion/Self Smoothing Operator to better adapt at the structure of the underlying manifold.
- Symmetric model and richer training set.
- Introduce sparsity in the hidden layer.
- Multi-scale algorithm.