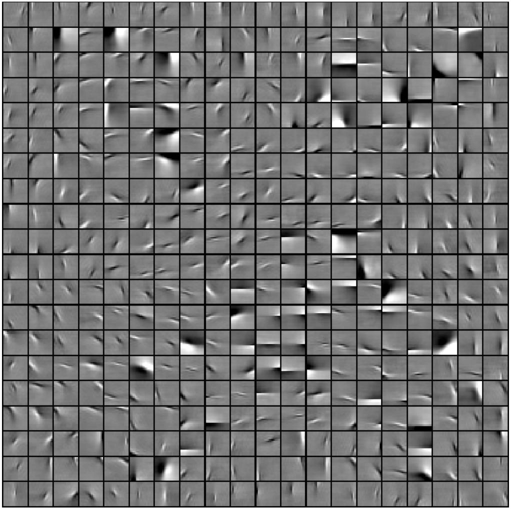
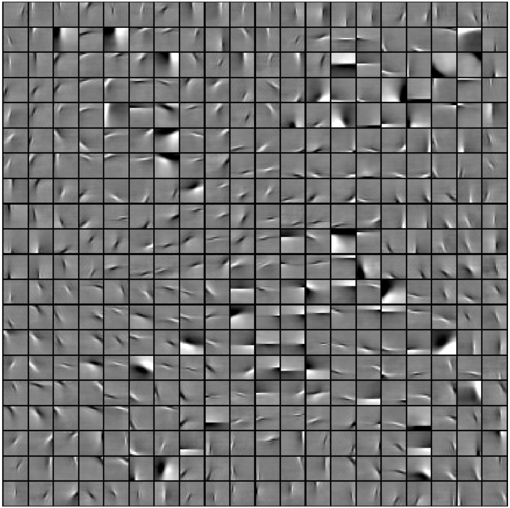
Outline

Outline

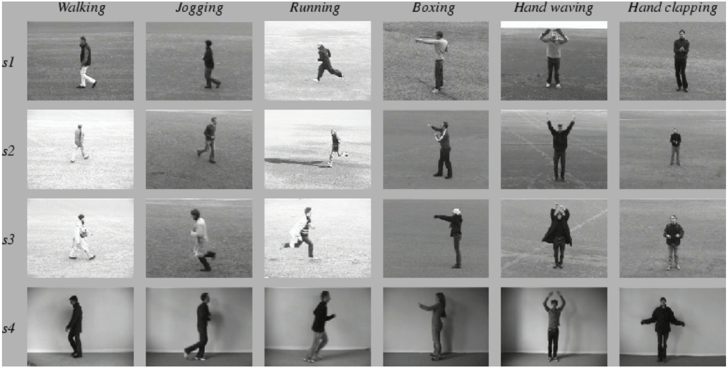
Bag-Of-Warps



Bag-Of-Warps



KTH Actions dataset



- Collapsing all hidden representations at monocular SIFT keypoints (across all keypoints and time frames) and performing logistic regression yields 80.56% correct.

Convolutional GBM

- Convolutional GBM (Taylor et al., 2010):

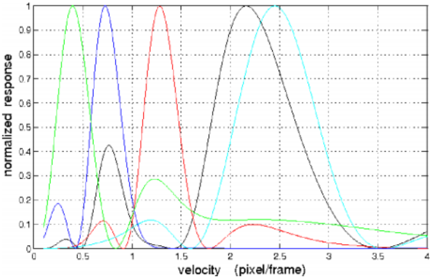
Prior Art	Accuracy	Convolutional architectures	Accuracy
HOG3D-KM-SVM	85.3	32convGRBM$^{16\times16}$-128F$^{9\times9\times9}_{CSG}$-R/N/P$^{4\times4\times4}_{A}$-log_reg	88.9
HOG/HOF-KM-SVM	86.1	32convGRBM$^{16\times16}$-128F$^{9\times9\times9}_{CSG}$-R/N/P$^{4\times4\times4}_{A}$-mlp	**90.0**
HOG-KM-SVM	79.0	32F$^{16\times16\times2}_{CSG}$-R/N/P$^{4\times4\times4}_{A}$-128F$^{9\times9\times9}_{CSG}$-R/N/P$^{4\times4\times4}_{A}$-log_reg	79.4
HOF-KM-SVM	88.0	32F$^{16\times16\times2}_{CSG}$-R/N/P$^{4\times4\times4}_{A}$-128F$^{9\times9\times9}_{CSG}$-R/N/P$^{4\times4\times4}_{A}$-mlp	79.5

- Convolutional GBM on Hollywood2:

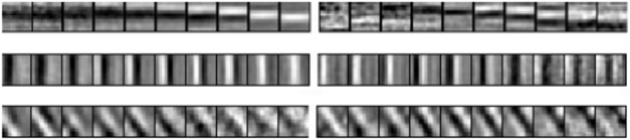
Method	AP
Prior Art [27]:	
HOG3D+KM+SVM	45.3
HOG/HOF+KM+SVM	**47.4**
HOG+KM+SVM	39.4
HOF+KM+SVM	45.5
convGRBM+SC+SVM	**46.6**

Stacked convolutional ISA

- (Le, et al., 2011)
- Velocity tuning of the higher-order features:



ISA applied to action recognition



- (Le, et al., 2011)

	KTH	Hollywood2	UCF	YouTube
until 2011	92.1	50.9	85.6	71.2
hierarchical ISA	93.9	53.3	86.5	75.8

Analogy making

$$A : A' \quad :: \quad B : ?$$

Analogy making

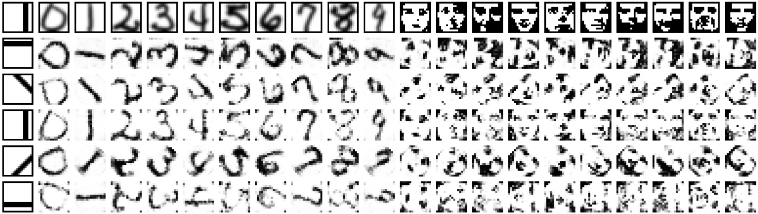
1. Infer transformation from *source* images $x_{\text{source}}, y_{\text{source}}$:

$$z(x_{\text{source}}, y_{\text{source}})$$

2. Apply the transformation to *target* image x_{target}:

$$y(z, x_{\text{target}})$$

Analogy making

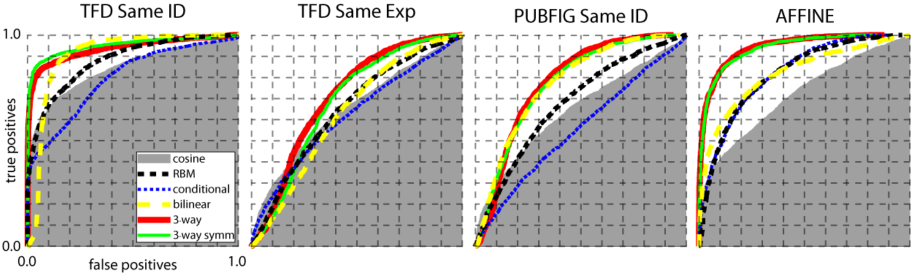


Filters learned from transforming faces

- Filters learned from faces:

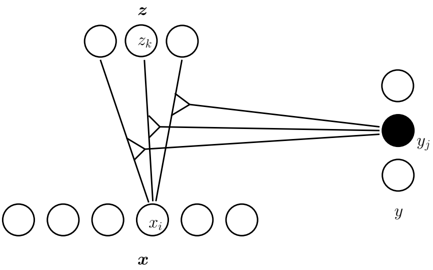
Metric learning and analogy making

- Learning a gated Boltzmann machine on changing facial expressions.
- (Susskind, et al., 2011)
- **Joint density training** allows for comparing compatibilities of pairs.



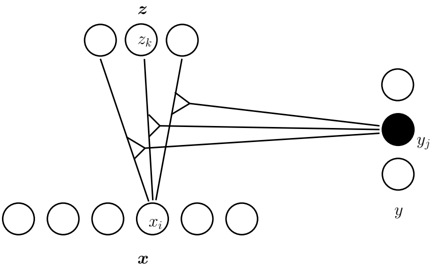
Model/Task	TFD ID	TFD Exp	PUBFIG ID	AFFINE
cosine	0.848	0.663	0.649	0.721
RBM	0.869	0.656	0.647	0.799
conditional	0.805	0.634	0.557	0.825
bilinear	0.905	0.637	**0.774**	0.812
3-way	0.932	**0.705**	0.771	0.930
3-way symm	**0.951**	0.695	0.762	**0.931**

Bi-linear classification



- Special case of a gated Boltzmann machine:
- Replace the output-“image” by a one-hot-encoded **class-label.**
- This is a classifier, where each *label can blend in it’s own model*!

Bi-linear classification



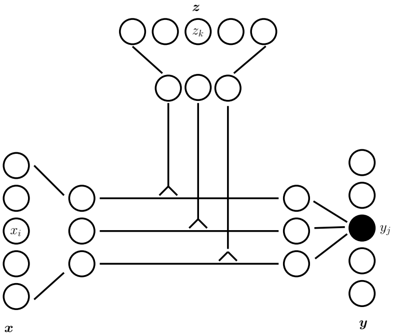
- Marginalization is tractable in closed form

$$p(y|\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(y, \boldsymbol{z}|\boldsymbol{x}) \quad \propto \quad \sum_{\boldsymbol{z}} \exp(\boldsymbol{x}^{\mathrm{t}} w_y \boldsymbol{z}) = \sum_{\boldsymbol{z}} \exp(\sum_{ik} w_{yik} x_i h_k)$$

$$= \quad \prod_k (1 + \exp(\sum_i w_{yik} x_i))$$

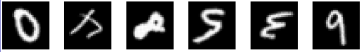
- It is also equivalent to a mixture of 2^K logistic regressors (Nair, 2008), (Memisevic, et al.; 2010), (Warrell et al.; 2010)

Bi-linear classification

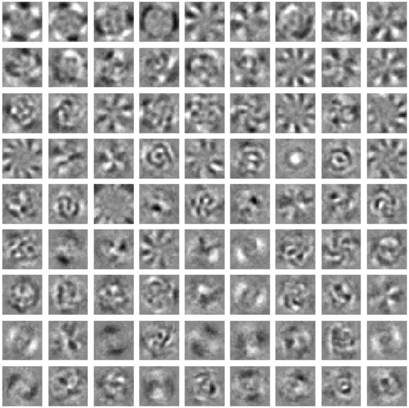


- We can factorize parameters like before.
- This allows classes to share features.
- The activity of a factor, f, given class j, is now exactly equal to the parameter value w^{y}_{jf}.
- Thus the weights can be thought of as the responses of **virtual class-templates**.

Rotated digit classification



- Data-set from the “deep learning-challenge” [Larochelle et al., 2007] like before.
- Learned rotation-invariant filters:

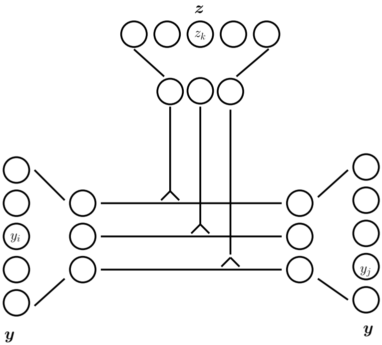


Bi-linear classification

- Deep Learning challenge (Larochelle et al., 2008).

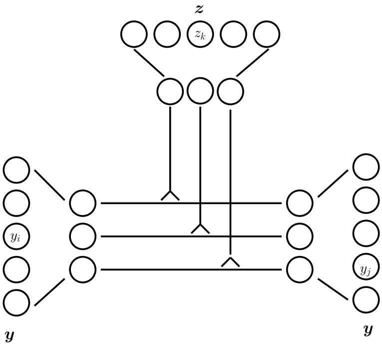
	SVMs		NNet	RBM	DEEP		GSM	
dataset/model:	SVMRBF	SVMPOL	NNet	DBN1	DBN3	SAA3	**GSM**	(unfact)
rectangles	2.15	2.15	7.16	4.71	2.60	2.41	0.83	(0.56)
rect.-images	24.04	24.05	33.20	23.69	22.50	24.05	22.51	(23.17)
mnistplain	3.03	3.69	4.69	3.94	3.11	3.46	3.70	(3.98)
convexshapes	19.13	19.82	32.25	19.92	18.63	18.41	17.08	(21.03)
mnistbackrand	14.58	16.62	20.04	9.80	6.73	11.28	10.48	(11.89)
mnistbackimg	22.61	24.01	27.41	16.15	16.31	23.00	23.65	(22.07)
mnistrotbackimg	55.18	56.41	62.16	52.21	47.39	51.93	55.82	(55.16)
mnistrot	11.11	15.42	18.11	14.69	10.30	10.30	11.75	(16.15)

Extension to *less* than two frames



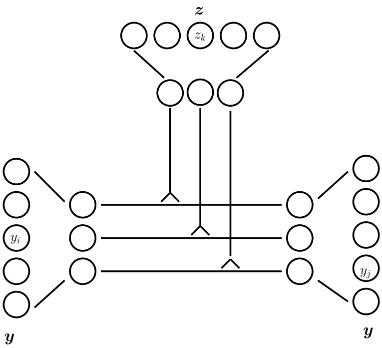
- To train energy models on *single images*:
- Plug in the same image left and right.
- Hiddens will model pixel covariance matrices.
- Eg., (Ranzato et al., 2010), (Karklin, Lewicki; 2008)
- Training can be finicky.

Extension to *less* than two frames



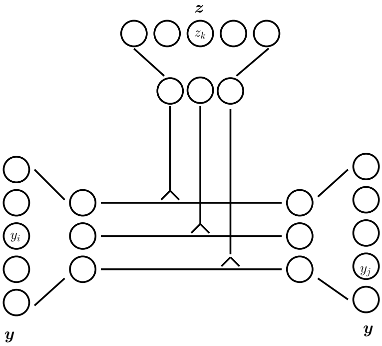
- To train energy models on *single images*:
- Plug in the same image left and right.
- Hiddens will model pixel covariance matrices.
- Eg., (Ranzato et al., 2010), (Karklin, Lewicki; 2008)
- ~~Training can be finicky.~~ Use a relational auto-encoder.

Extension to *less* than two frames



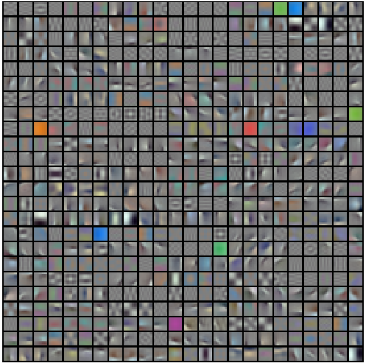
- We can combine this with standard hidden units in one model.
- The combination tends to work better recognition (Ranzato et al., 2010).
- The vanilla hidden units then plays the role of "higher-order-biases" (Memisevic, 2007).

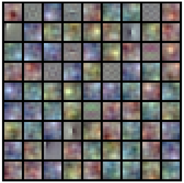
Extension to *less* than two frames



- Learning higher-order within-image structure has been suggested to address the fact that ICA does not really yield independent components...
- Add layers to model correlations of filter responses.
- Closely related to Deep Learning.

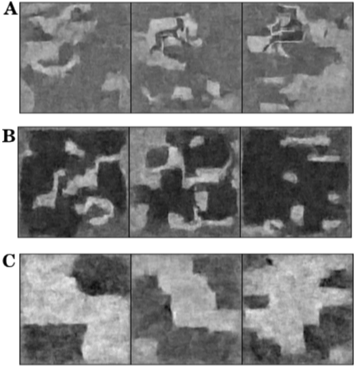
Some within image covariance and mean filters





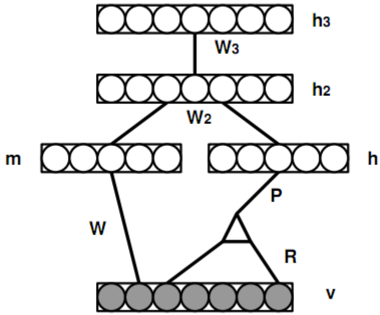
Within-image correlations

- (Karklin, Lewicki; 2008), (Osindero et al., 2006), ...
- ISA itself used mainly for modeling within-image structure.
- (Ranzato et al., 2010) suggest combining covariance features and traditional “mean” features, for example to generate images with an MRF:



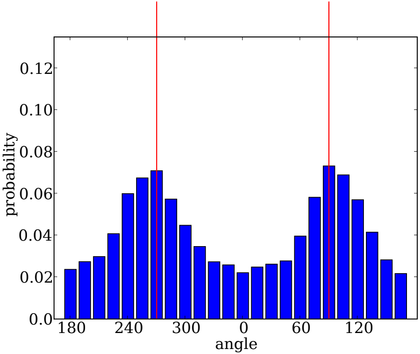
mcRBMs on TIMIT

- mcRBM applied to speech recognition (phones, speaker independent, TIMIT)
- (Dahl, et al.; 2010)



Method	PER
Stochastic Segmental Models [17]	36%
Conditional Random Field [18]	34.8%
Large-Margin GMM [19]	33%
CD-HMM [20]	27.3%
Augmented conditional Random Fields [20]	26.6%
Recurrent Neural Nets [21]	26.1%
Bayesian Triphone HMM [22]	25.6%
Monophone HTMs [23]	24.8%
Heterogeneous Classifiers [24]	24.4%
Deep Belief Networks(DBNs) [5]	23.0*%
Triphone HMMs discriminatively trained w/ BMMI [7]	22.7%
Deep Belief Networks with mcRBM feature extraction (this work)	**21.7*%**
Deep Belief Networks with mcRBM feature extraction (this work)	**20.5%**

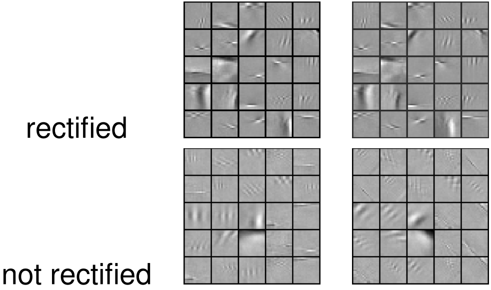
Transparent motion



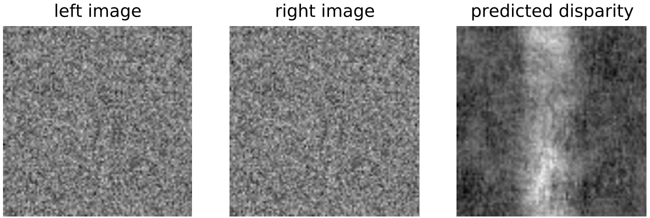
- Hidden variables make extracting multiple, simultaneous motions easy.
- When they fail they do so in a similar way as humans:
- Better disrimination at large angles, averaging at very small angles, “motion repulsion”.
- (eg., Treue et al., 2000)

Depth as a latent variable

- Learning a dictionary for stereo:
- Generate left-right camera pairs with known disparities.
- *Predict* disparity from the hidden units.
- This gives rise to a three-layer network, that may be trained with Hebbian-like learning.

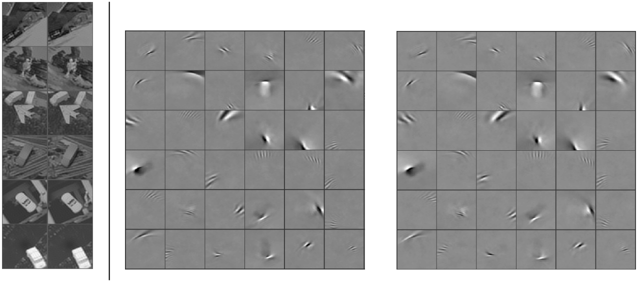


Hiddens learn to encode disparities



- Can use this to encode 3d-structure implicitly, for example, for multi-view recognition.

Norb stereo features

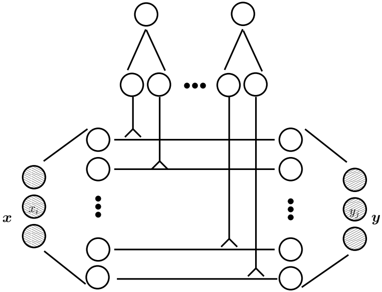


NORB training subset:				NORB testset:		
RBMmon	RBMbin	cc	cc+bin	RBMbin	cc	cc+bin
73.65	60.43	34.85	31.48	63.28	38.91	36.80

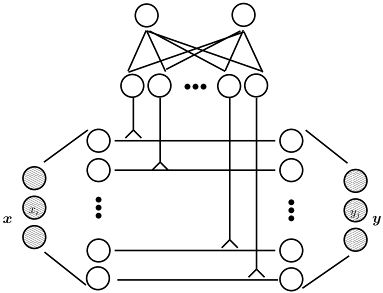
“Harnessing the aperture problem”

- **Transformations are transformation invariant.**
- The 2-D subspace projections, however, are at the same time affected by the aperture problem, so they are selective to other sources of variability, including object ID!
- We can use the aperture effect to build invariant features:

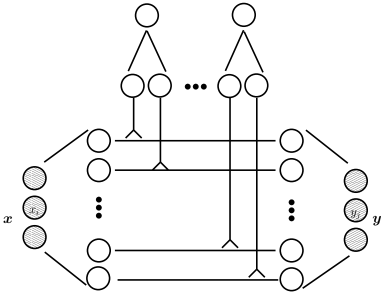
Harnessing the aperture problem



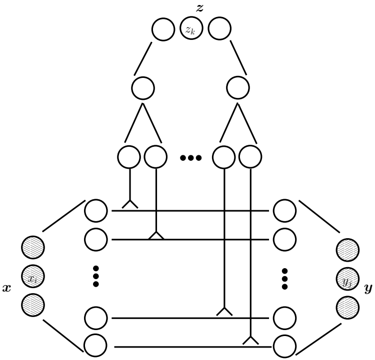
Harnessing the aperture problem



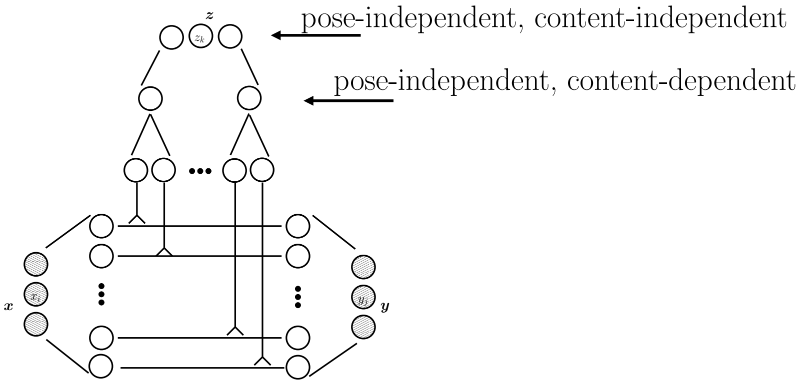
Harnessing the aperture problem



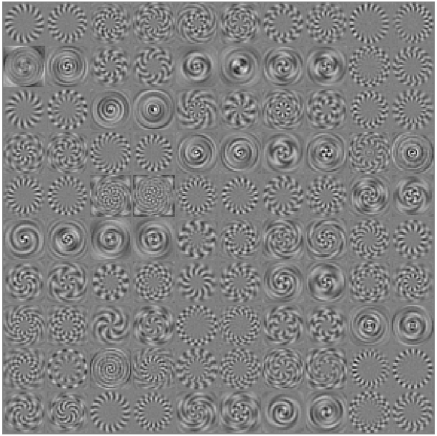
Harnessing the aperture problem



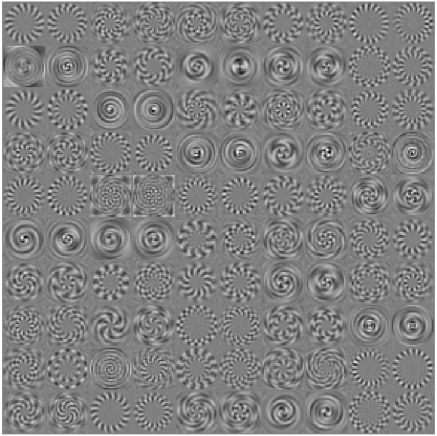
Harnessing the aperture problem



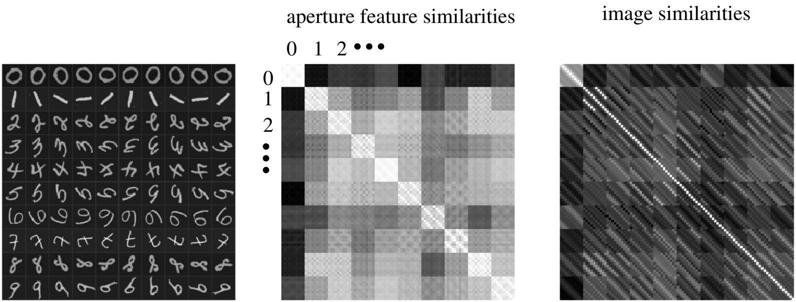
Rotation "quadrature" filters



Rotation “quadrature” filters

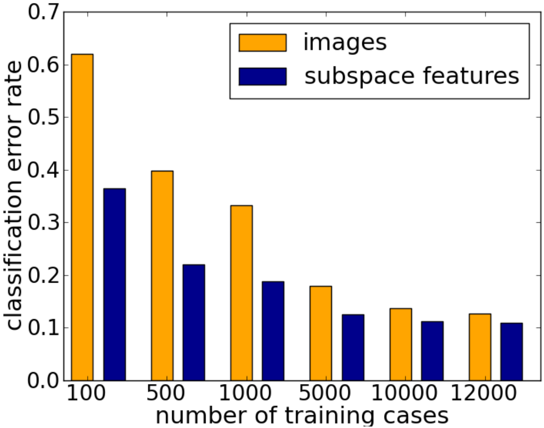


Representing digits using rotation aperture features



- Learn rotation features. Represent digits using aperture features.
- No video available? Fill video buffer with copies of the same image: Represent the non-transformation.

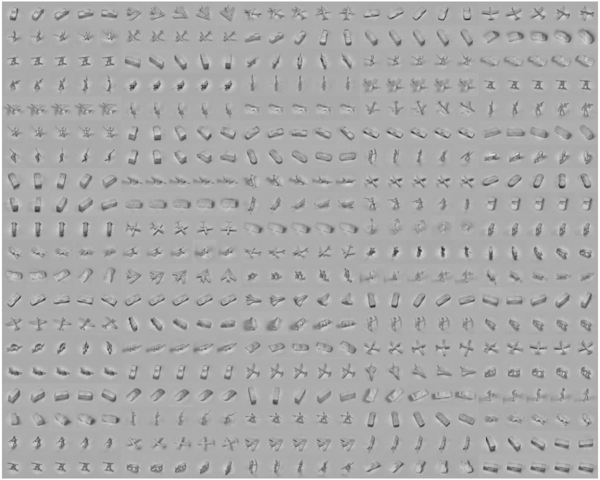
Rotated MNIST error rates



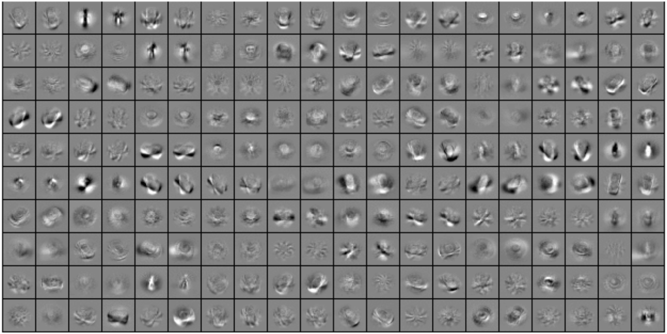
Video object features

- Humans do not recognize still images but videos of objects.
- The way in which an object changes can convey useful information about the object, including 3-D structure.
- $\rightarrow$ **Learn features from videos not still images.** For example, (Lee and Soatto, 2011).

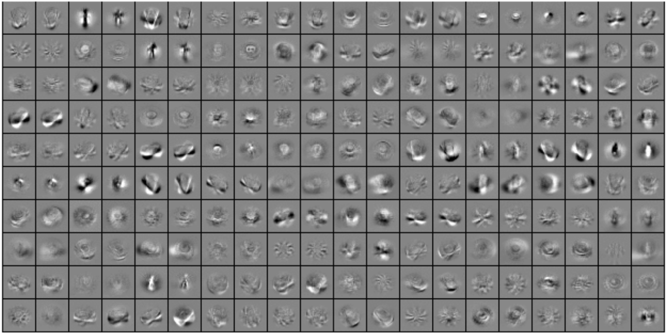
The “norbjects” video dataset



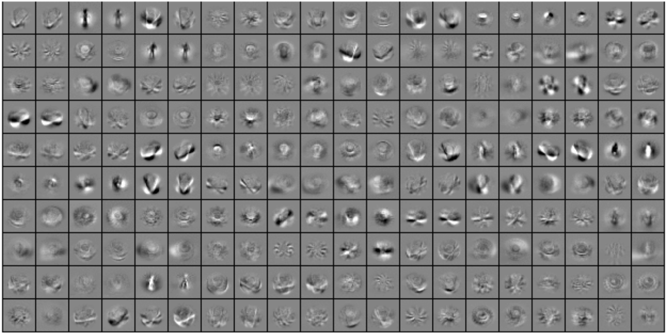
3-D rotation subspaces



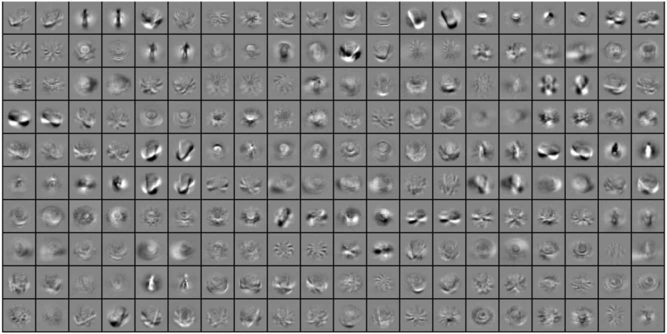
3-D rotation subspaces



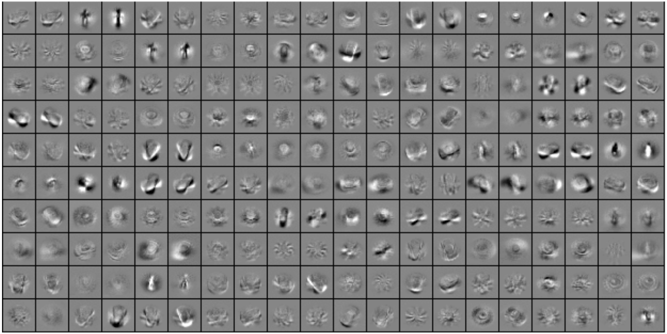
3-D rotation subspaces



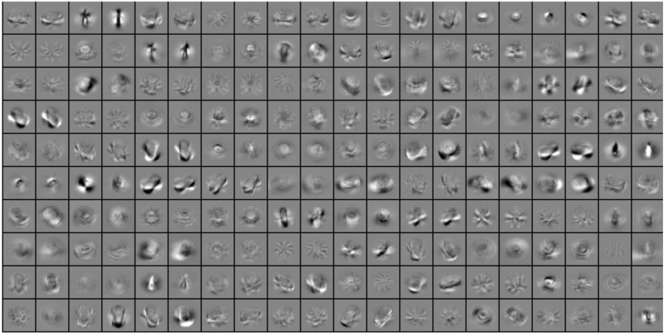
3-D rotation subspaces



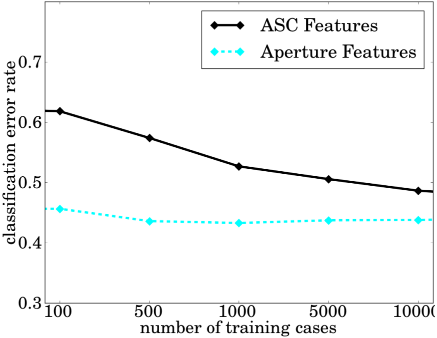
3-D rotation subspaces

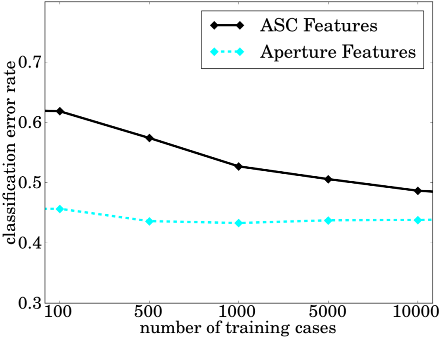


3-D rotation subspaces



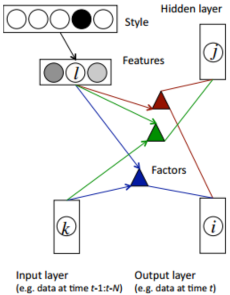
“Harnessing the aperture problem”

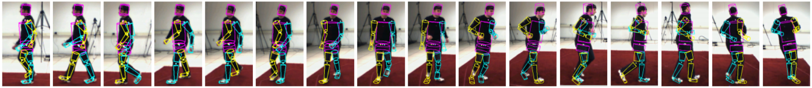




Mocap

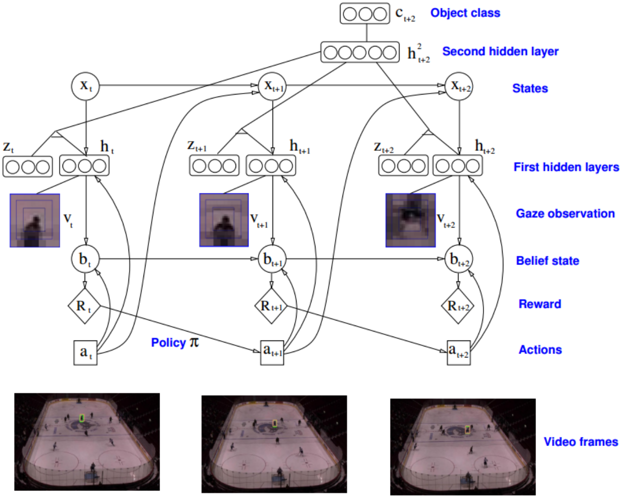
- (Taylor, Hinton; 2009), (Taylor, et al.; 2010)
- Learning models on mocap instead of images makes it possible to model motion style and to perform tracking.





Training	Test	Baseline	MoCorr [28]	GPLVM [13]	CMFA-VB [13]	CRBM	imCRBM-10
S1+S2+S3	S1	129.18±19.47	140.35	-	-	55.43±0.79	**54.27±0.49**
S1	S1		-	-	-	**48.75±3.72**	58.62±3.87
S1+S2+S3	S2	162.75±15.36	149.37	-	-	99.13±22.98	**69.28±3.30**
S2	S2		-	88.35±25.66	68.67±24.66	**47.43±2.86**	67.02±0.70
S1+S2+S3	S3	180.11±24.02	156.30	-	-	70.89±2.10	**43.40±4.12**
S3	S3		-	87.39±21.69	69.59±22.22	**49.81±2.19**	51.43±0.92

More Tracking



- (Bazzani et al.), (Larochelle, Hinton, 2011)

Outline

Conclusions

- *Learning* is a way to support simplicity and homogeneity of complex, intelligent systems.
- *Feature* learning even more so.
- *Relational* feature learning even more:

- Learning “verbs”, not just “nouns”, can help address more tasks with a single kind of model.
- *This seems like a very good reason to have complex cells.*
- One reason, why looking for *correspondences* – across frames, across views, across modalities, etc. – is a common operation, is that mappings between modalities are often *one-to-many*.
- The theory provides a strong inductive bias for products and/or squaring non-linearities when building deep learning models.

Thank you

- More info, code, links, etc. at

  `http://www.cs.toronto.edu/~rfm/multiview-feature-learning-cvpr/index.html`