Multiview Feature Learning

Roland Memisevic

Uni Frankfurt

Tutorial at CVPR 2012

Higher-order Feature Learning

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

Outline

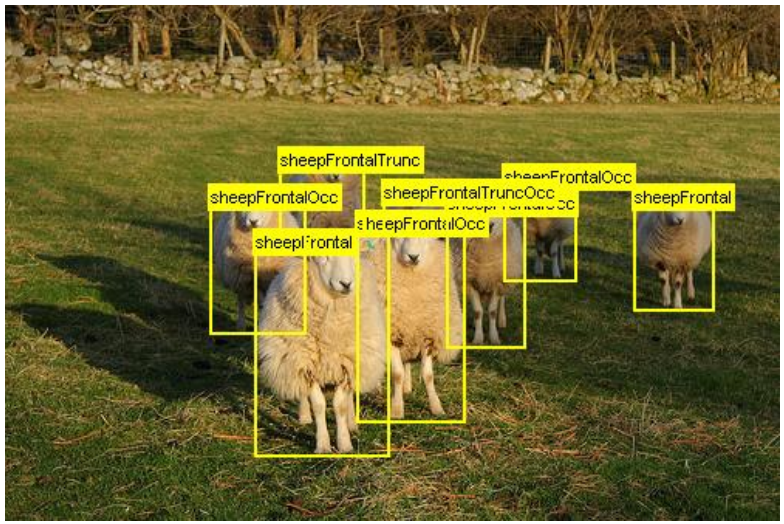
Outline

What this is about

- Extending feature learning to model *relations*.
- “Bi-linear models”, “energy-models”, “complex cells”, “spatio-temporal features”, “covariance features”, “bi-linear classification”, “mapping units”, “quadrature features”, “gated Boltzmann machine”, “mcrbm”, ...

Recognition tasks

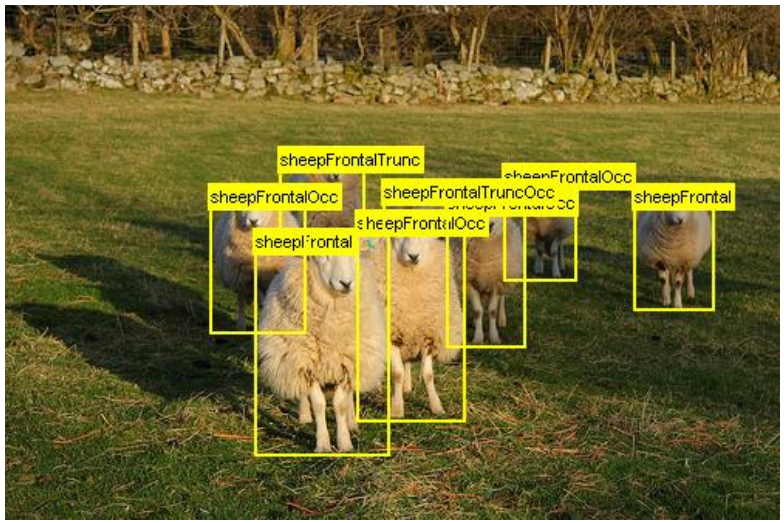
- Recognition has become a focus of interest in Computer Vision.
- Recognition of static objects started to work *very* well. In fact –
- Recognition is getting quite serious.



(PASCAL challenge)

“It’s the feature, stupid!”

- A main reason is the use of the *right representation*:
- Recognition started to work after the community converged on **local features**, like SIFT.
- With the right representation, the choice of top level classifier (SVM, logreg, NN) doesn’t matter all that much.



(PASCAL challenge)

Recognition with local features

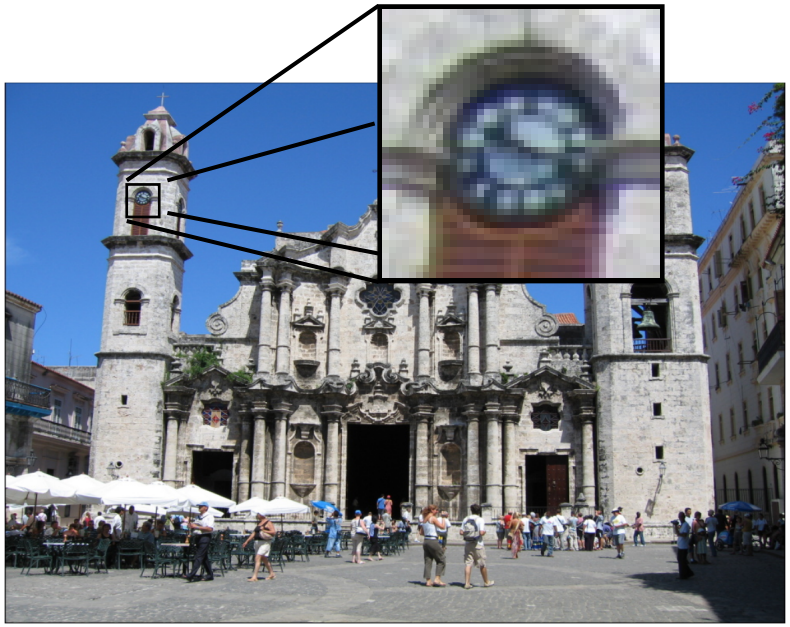
- Task: Recognize the building.
- Two approaches:
  1. Bag-Of-Features
  2. Convolution

Bag-Of-Features

Bag-Of-Features

1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** (“features”).
4. **Add** all local descriptors to obtain a global descriptor for the image.

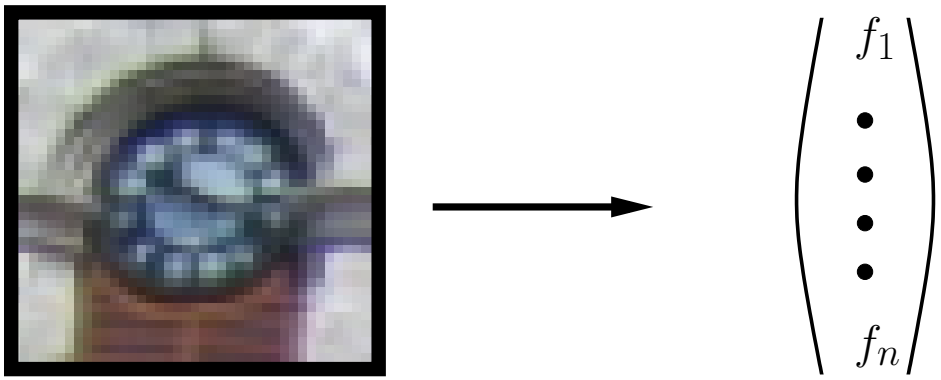
Bag-Of-Features



Bag-Of-Features

1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** (“features”).
4. **Add** all local descriptors to obtain a global descriptor for the image.

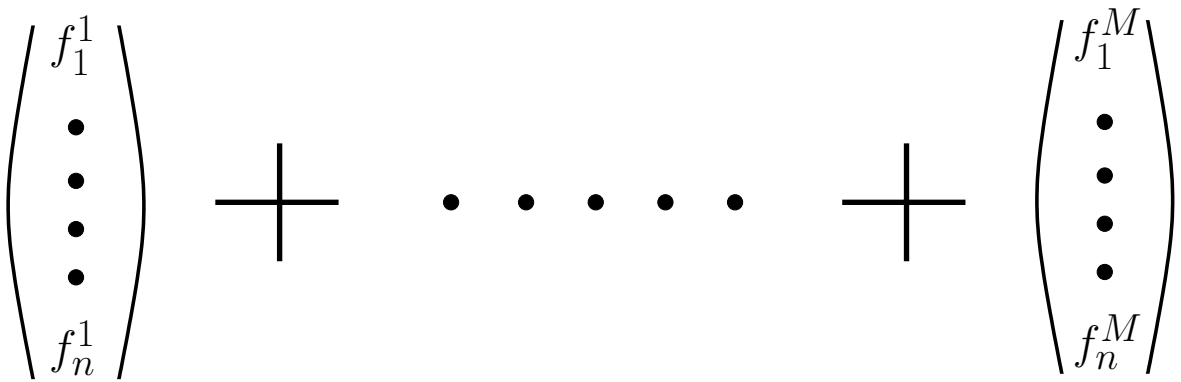
Bag-Of-Features



Bag-Of-Features

1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** (“features”).
4. **Add** all local descriptors to obtain a global descriptor for the image.

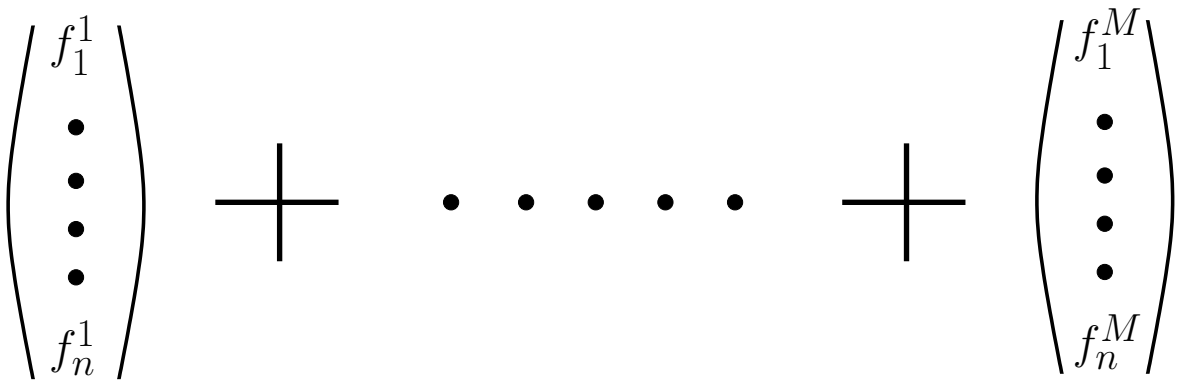
Bag-Of-Features



Bag-Of-Features

1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** (“features”).
4. **Add** all local descriptors to obtain a global descriptor for the image.

Bag-Of-Features



Bag-Of-Features

1. Find **interest points** (AKA keypoints).
2. Crop patches around interest points.
3. Represent each patch with a **sparse local descriptor** (“features”).
4. **Add** all local descriptors to obtain a global descriptor for the image.

Convolutional

Convolutional

1. Crop patches along a regular grid (dense or not).
2. Represent each patch with a local descriptor.
3. **Concatenate** all descriptors into a very large vector.

Convolutional

Convolutional

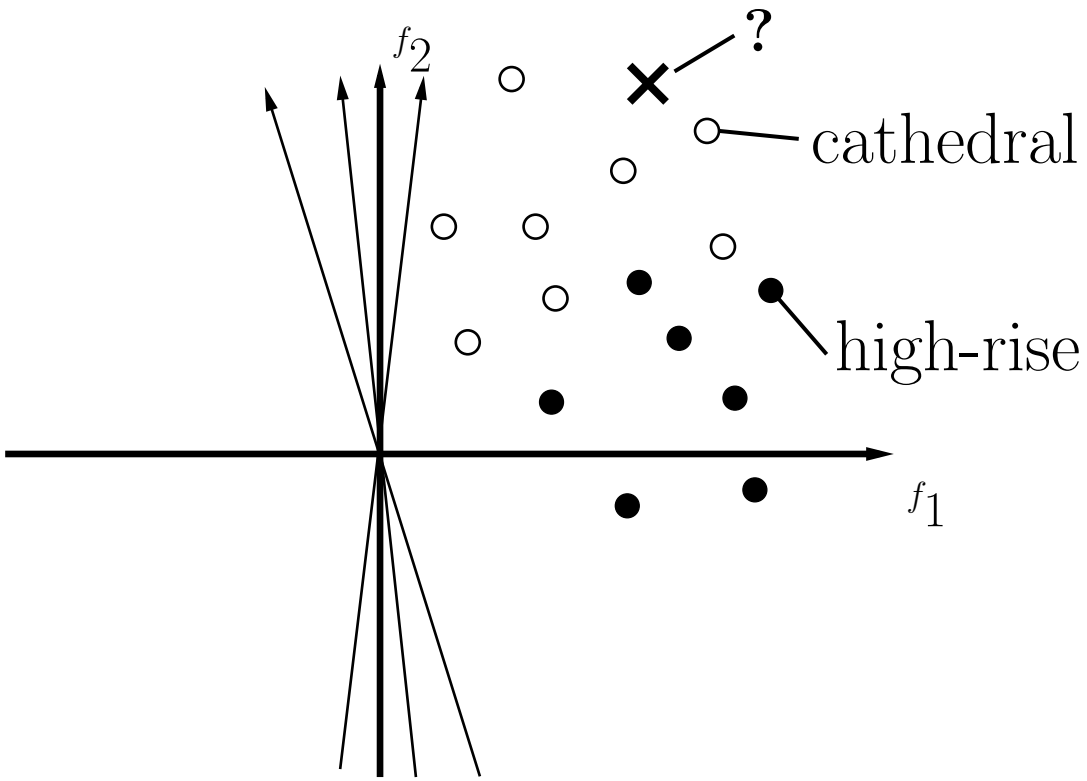
1. Crop patches along a regular grid (dense or not).
2. Represent each patch with a local descriptor.
3. **Concatenate** all descriptors into a very large vector.

Convolutional

Convolutional

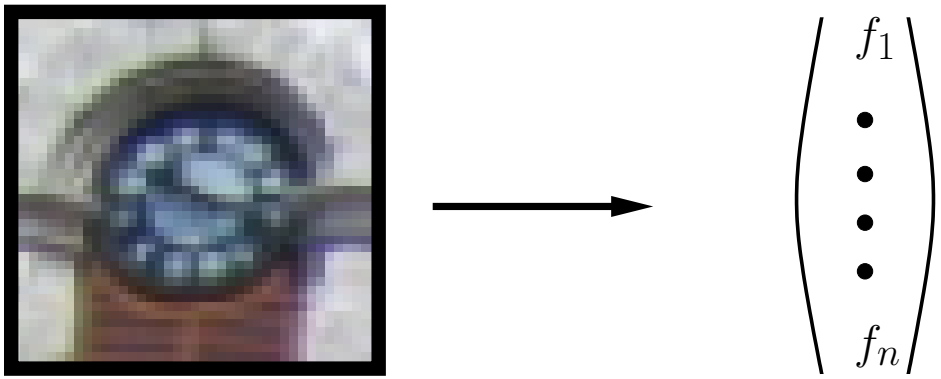
1. Crop patches along a regular grid (dense or not).
2. Represent each patch with a local descriptor.
3. **Concatenate** all descriptors into a very large vector.

Classification



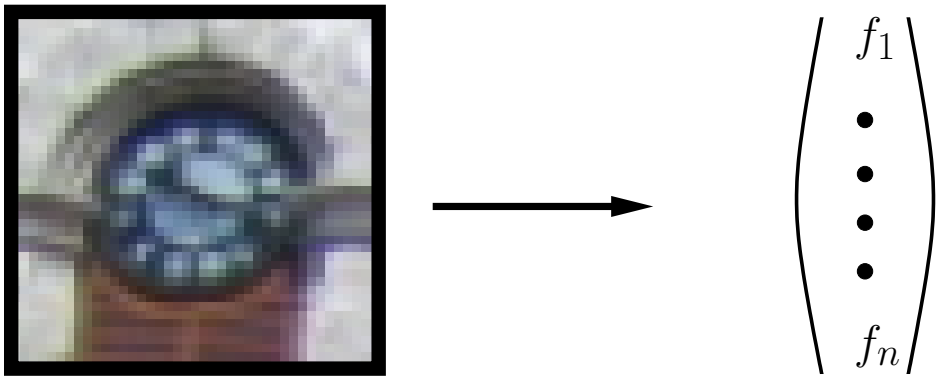
- When images are represented as points in $\mathbb{R}^n$, we can use a simple classifier to do recognition.
- Eg., Logistic regression, SVM, NN, ...
- (There are various extensions, like fancy pooling, etc.)

Feature Learning



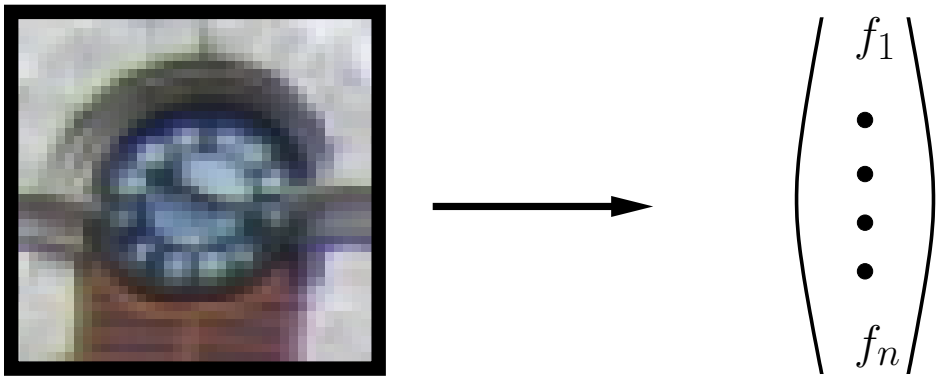
- How do we get good features?
- Option B: Engineer them. SIFT, HOG, LBP, etc.
- Natural Images are not nandom.
- Option A: *Learn* the representation from image data.

Feature Learning



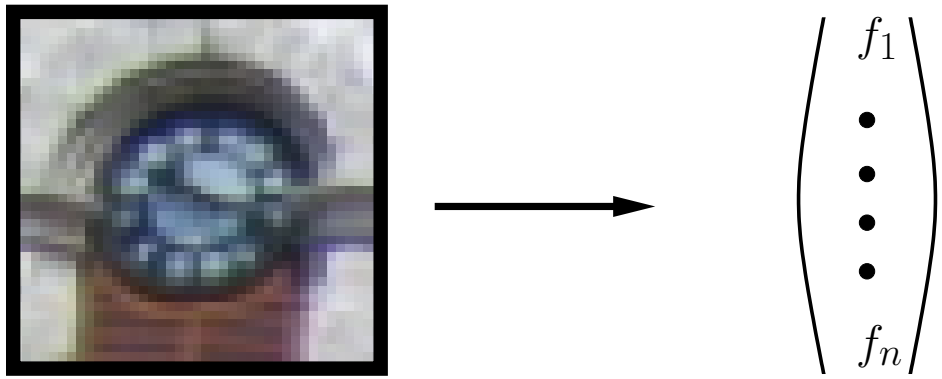
- How do we get good features?
- Option B: Engineer them. SIFT, HOG, LBP, etc.
- Natural Images are not nandom.
- Option A: *Learn* the representation from image data.

Feature Learning



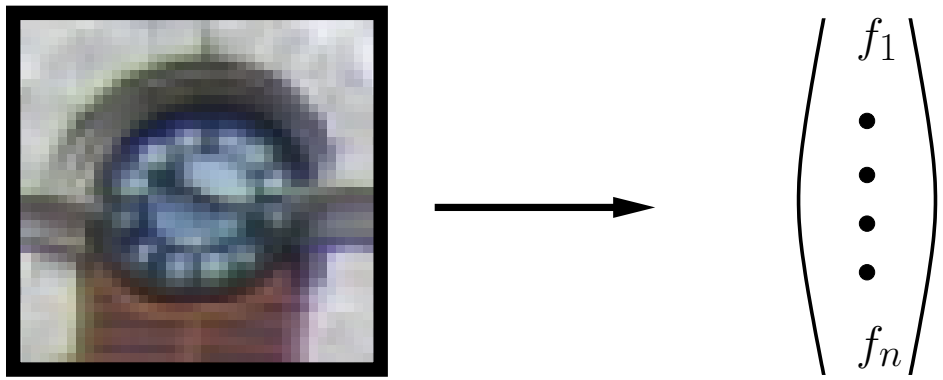
- How do we get good features?
- Option B: Engineer them. SIFT, HOG, LBP, etc.
- Natural Images are not nandom.
- Option A: *Learn* the representation from image data.

Feature Learning



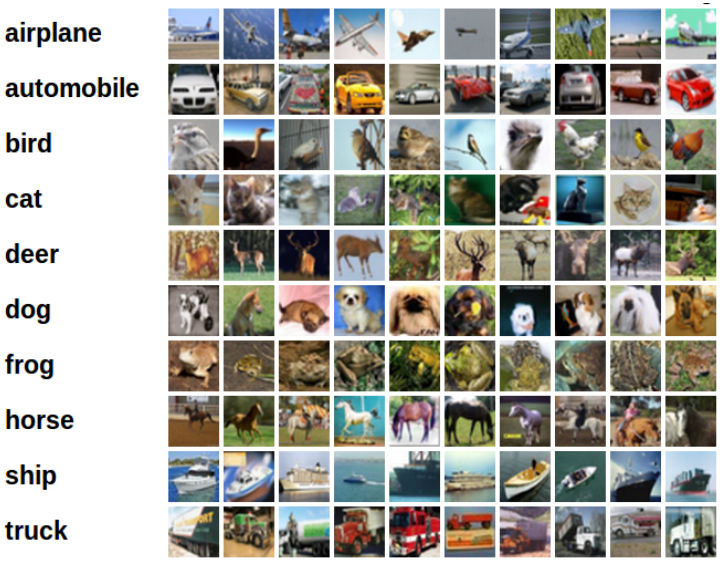
- How do we get good features?
- Option B: Engineer them. SIFT, HOG, LBP, etc.
- Natural Images are not nandom.
- Option A: *Learn* the representation from image data.

Why Feature Learning

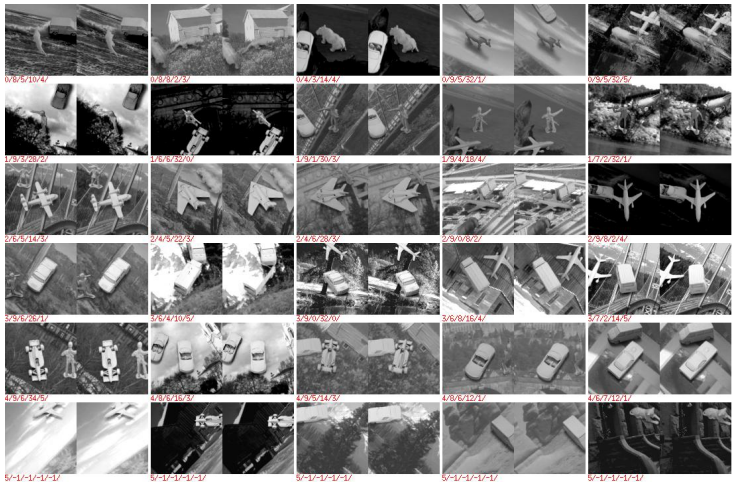


- **Feature Learning**, **Dictionary Learning**, **Receptive Field Learning**, **Sparse Coding**, etc.
  - Helps overcome tedious engineering.
  - Helps adapt models to different data domains (including a model's own representations! – **"deep learning"**)
  - Biologically consistent.
  - Brings us closer to *end-to-end learning* of vision systems.

Feature Learning Works



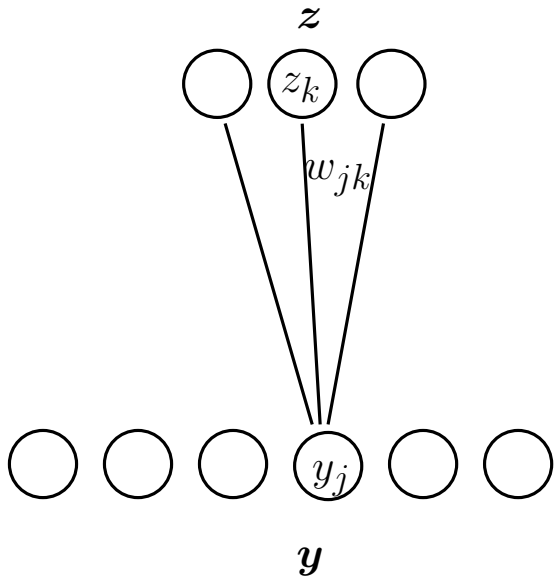
(CIFAR)



(NORB)

- More importantly... it works well
- See, eg., (Coates, et al., 2011)

Feature Learning

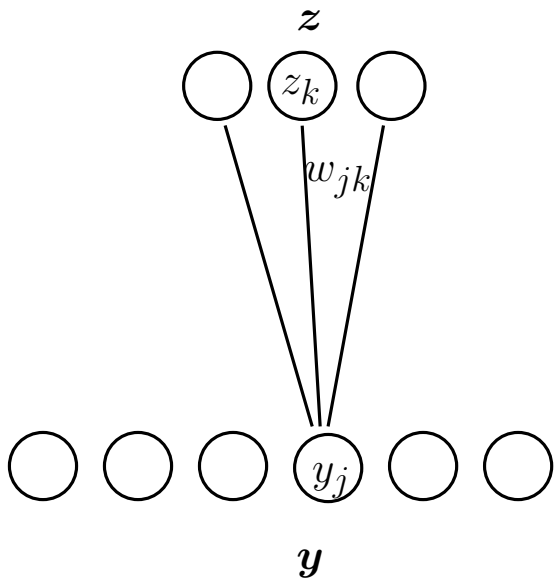


Feature Learning

- Encode patch y using latent variables z.
- Learn weights W from training data of image patches.

Feature Learning works well for recognition, so...

Feature Learning



Feature Learning

- Encode patch y using latent variables z.
- Learn weights W from training data of image patches.

Feature Learning works well for recognition, so...

Outline

Beyond object recognition

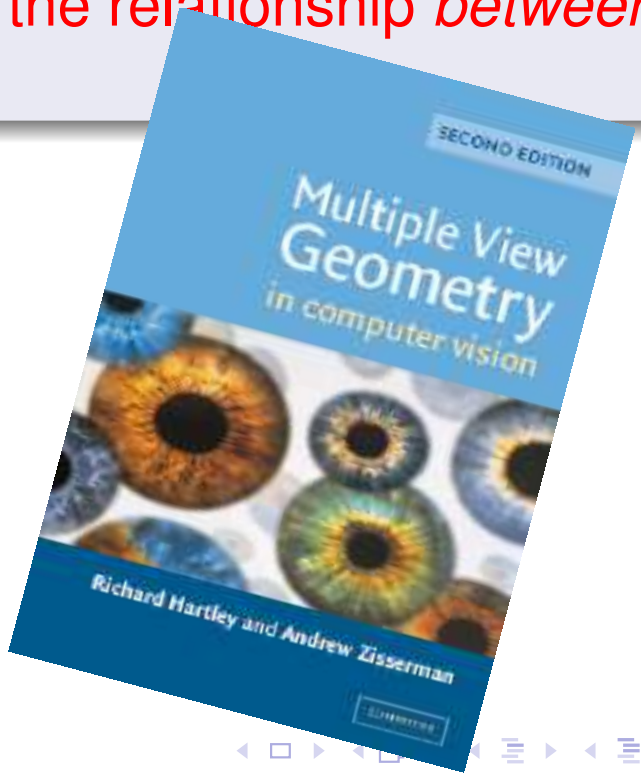
Can we do more with Feature Learning than recognize *things*?

- Good features work well for object recognition.
- But brains can do much more than recognize objects.
- A large number of vision tasks goes beyond object recognition.
- In surprisingly many vision tasks, the relationship *between* images carries the relevant information

Beyond object recognition

Can we do more with Feature Learning than recognize *things*?

- Good features work well for object recognition.
- But brains can do much more than recognize objects.
- A large number of vision tasks goes beyond object recognition.
- In surprisingly many vision tasks, the relationship *between* images carries the relevant information



Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

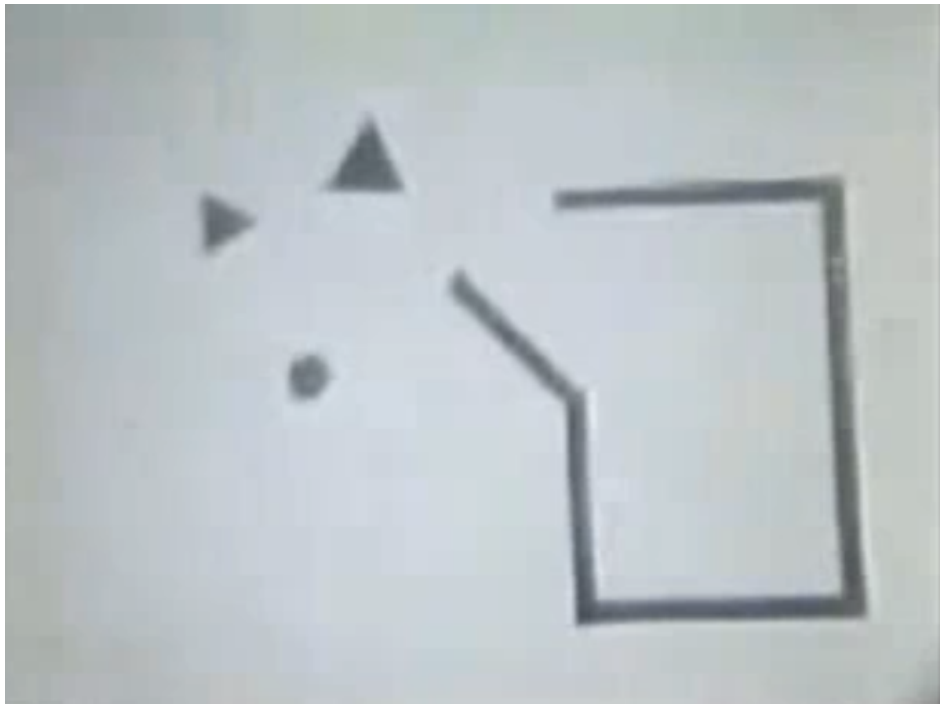
Correspondences in Computer Vision

- **Correspondence** is one of the most ubiquitous problems in Computer Vision.

Some correspondence tasks in Vision

- Tracking
- Stereo
- Geometry
- Optical Flow
- Invariant Recognition
- Odometry
- Action Recognition
- Contours, Within-image structure

Heider and Simmel

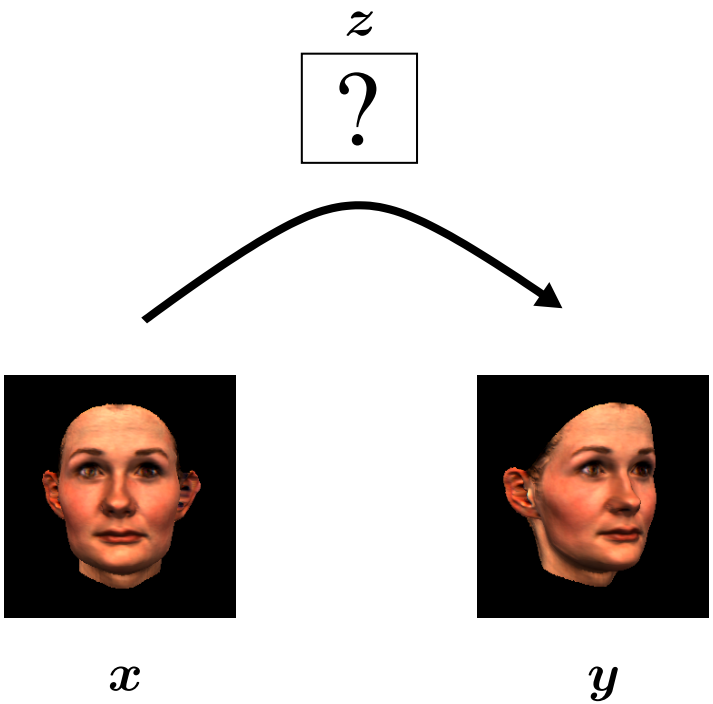


- Adding frames is not just about adding proportionally more information.
- The relationships between frames contain additional information, that is not present in any single frame.
- See *Heider and Simmel, 1944:* Any single frame shows a bunch of geometric figures. The motions reveal the story.

Outline

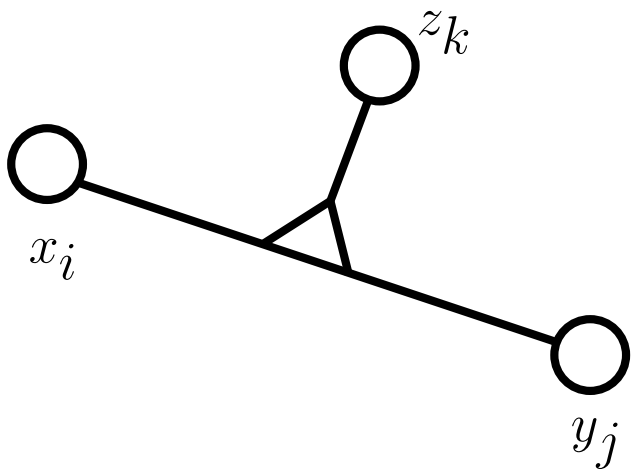
Learning features to model correspondences

- If *correspondences* matter in vision, **can we learn them**?



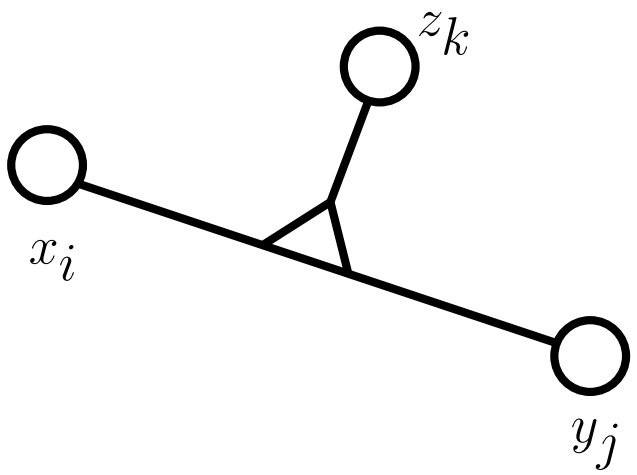
Learning features to model correspondences

- It turns out that this requires latent variables to act like *gates*, that dynamically change the connections between fellow variables.



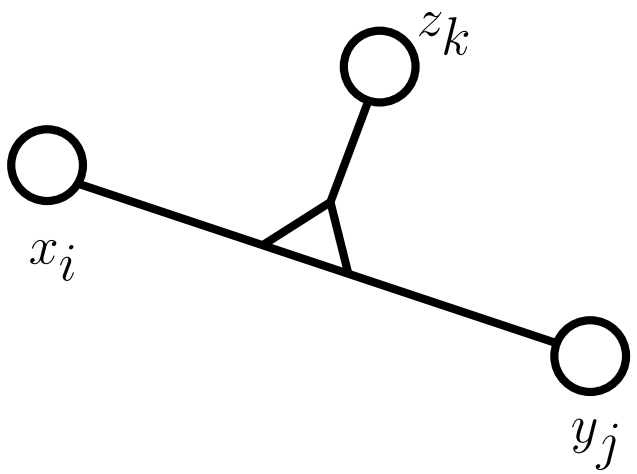
Learning features to model correspondences

- This amounts to letting variables multiply connections between other variables.
- And it is equivalent to having *three-way multiplicative interactions*.



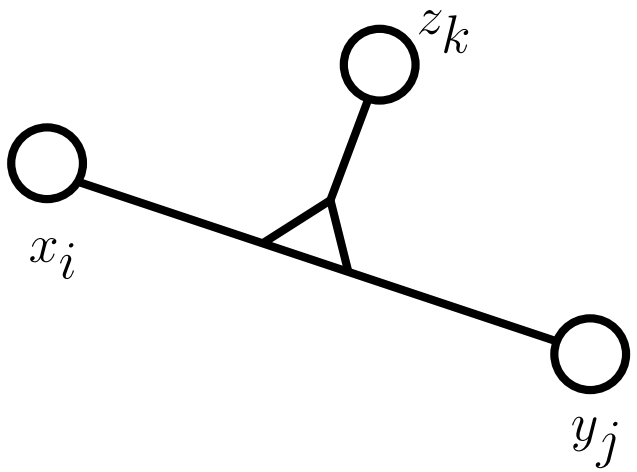
Learning features to model correspondences

- Learning and inference in the presence of gating variables is (slightly) different from learning without.
- We can set things up, such that inference is almost unchanged. Yet the *meaning* of the latent variables will be entirely different.



Learning features to model correspondences

- Multiplicative interactions allow hidden variables to *blend in a whole "sub"-network*.
- This leads to a qualitatively quite different behaviour from all common bi-partite feature learning models.



Multiplicative interactions

Brief history of multiplication

- "Mapping units" (Hinton; 1981), "dynamic mappings" (v.d. Malsburg; 1981)
- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, "Sigma-Pi-units"
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...

Multiplicative interactions

Brief history of multiplication

- "Mapping units" (Hinton; 1981), "dynamic mappings" (v.d. Malsburg; 1981)
- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, "Sigma-Pi-units"
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...

Multiplicative interactions

Brief history of multiplication

- "Mapping units" (Hinton; 1981), "dynamic mappings" (v.d. Malsburg; 1981)
- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, "Sigma-Pi-units"
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...

Multiplicative interactions

Brief history of multiplication

- "Mapping units" (Hinton; 1981), "dynamic mappings" (v.d. Malsburg; 1981)
- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, "Sigma-Pi-units"
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...

Multiplicative interactions

Brief history of multiplication

- "Mapping units" (Hinton; 1981), "dynamic mappings" (v.d. Malsburg; 1981)
- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, "Sigma-Pi-units"
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...

Multiplicative interactions

Brief history of multiplication

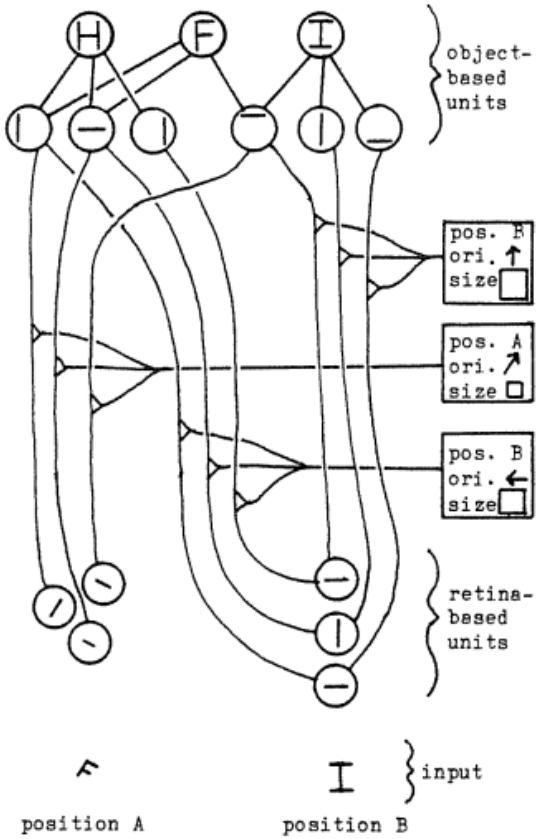
- "Mapping units" (Hinton; 1981), "dynamic mappings" (v.d. Malsburg; 1981)
- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, "Sigma-Pi-units"
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...

Multiplicative interactions

Brief history of multiplication

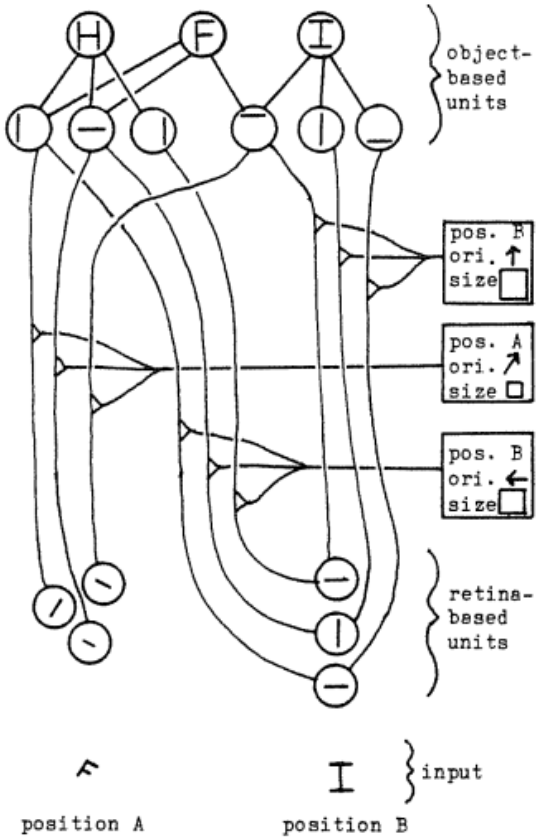
- “Mapping units” (Hinton; 1981), “dynamic mappings” (v.d. Malsburg; 1981)
- Binocular+Motion Energy models (Adelson, Bergen; 1985), (Ozhawa, DeAngelis, Freeman; 1990), (Fleet et al., 1994)
- Higher-order neural nets, “Sigma-Pi-units”
- Bi-linear models (Tenenbaum, Freeman; 2000), (Ohlshausen; 1994), (Grimes, Rao; 2005)
- Subspace SOM (Kohonen, 1996)
- ISA, topographic ICA (Hyvarinen, Hoyer; 2000), (Karklin, Lewicki; 2003): Higher-order within image structure
- (2006 –) GBM, mcRBM, RAE, convISA, applications...

Mapping units 1981



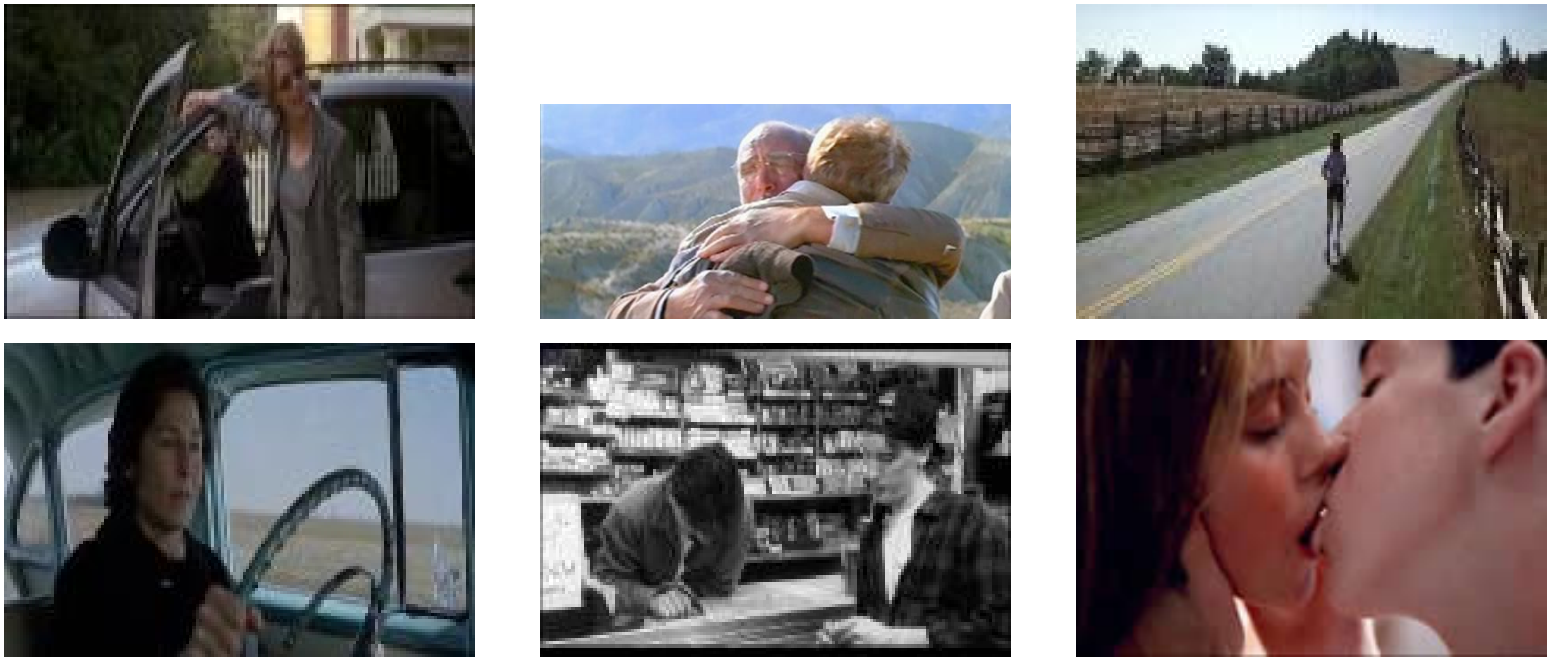
(Hinton, 1981)

Mapping units 1981



(Hinton, 1981)

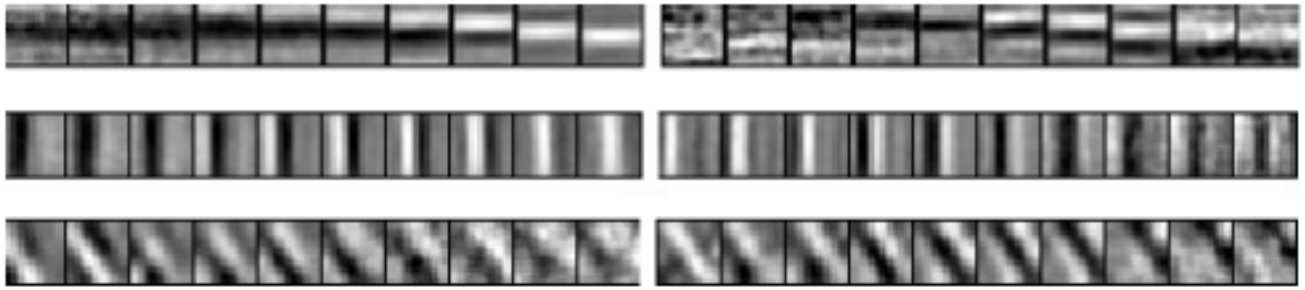
Example application: Action recognition



(Hollywood 2)

(Marszałek et al., 2009)

ISA applied to action recognition

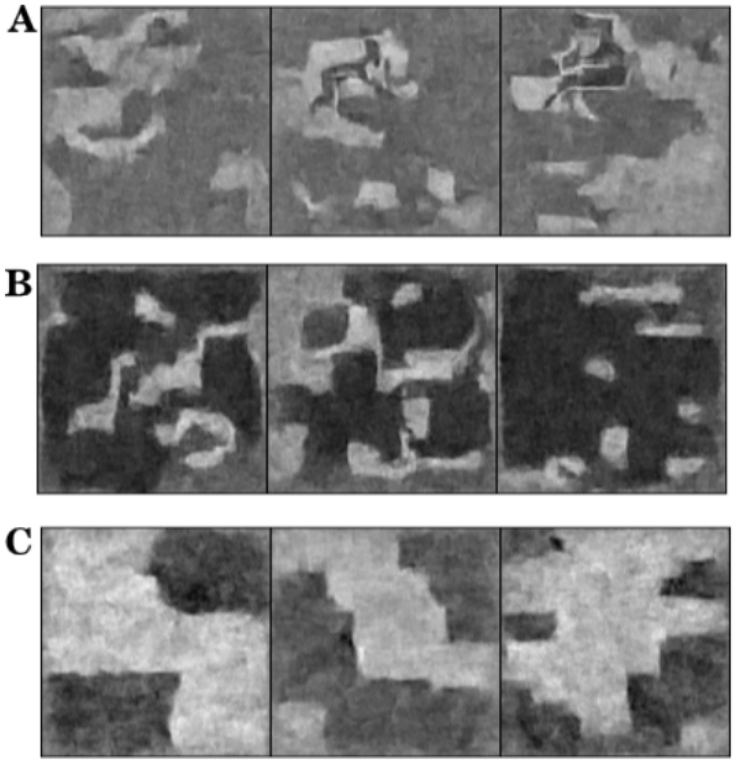


- (Le, et al., 2011)

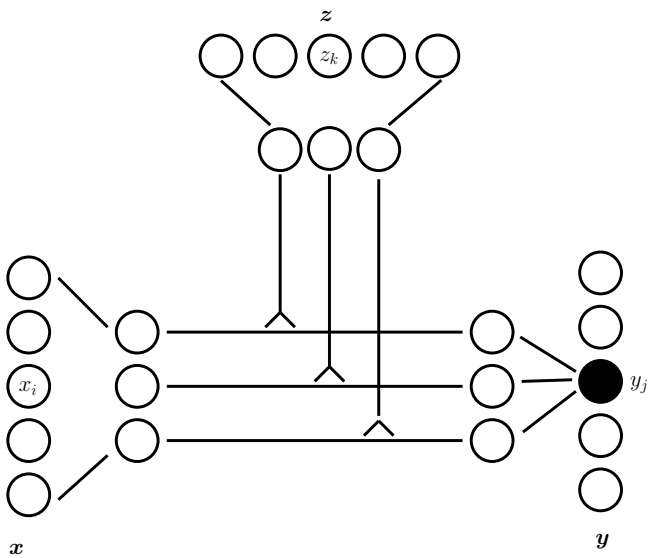
	KTH	Hollywood2	UCF	YouTube
until 2011	92.1	50.9	85.6	71.2
hierarchical ISA	93.9	53.3	86.5	75.8

Learning higher-order features

- (Ranzato et al., 2010)



Bi-linear classification



- Let *labels act like gates.* (Nair et al., 2009; Memisevic et al., 2010)

	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)

Tracking

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