FEATURE LEARNING FOR COMPARING EXAMPLES

IPAM GRADUATE SUMMER SCHOOL ON DEEP LEARNING

GRAHAM TAYLOR

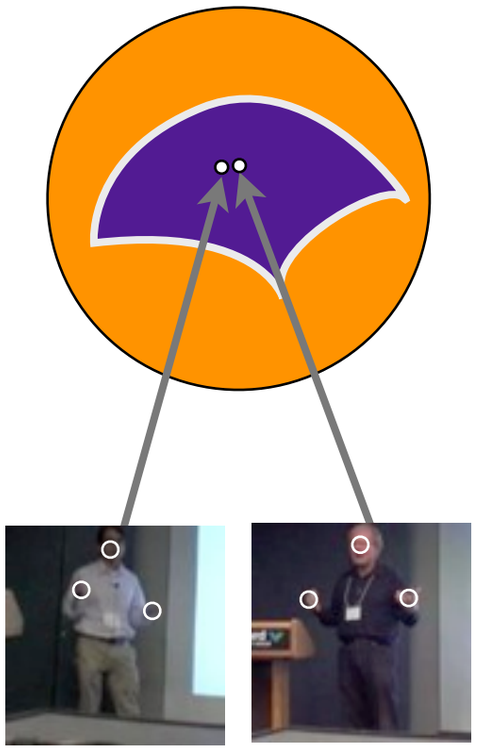
SCHOOL OF ENGINEERING
UNIVERSITY OF GUELPH

Papers and software available at: http://www.uoguelph.ca/~gwtaylor

OVERVIEW: THIS TALK

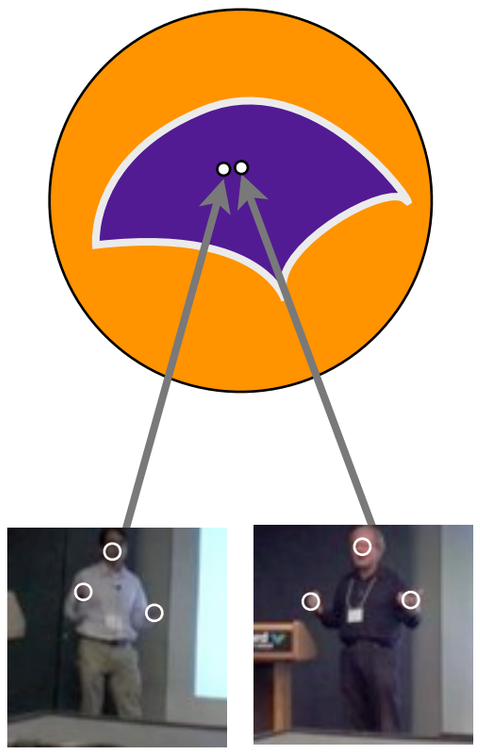
OVERVIEW: THIS TALK

- Learning to compare examples
  - it's a big field!
  - we will focus on methods inspired by deep learning and representation learning



OVERVIEW: THIS TALK

- Learning to compare examples
  - it's a big field!
  - we will focus on methods inspired by deep learning and representation learning

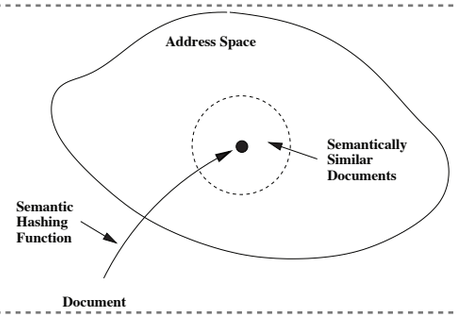


- Applications: finding similar documents, human pose estimation, pose-sensitive retrieval
  ... and a Dutch progressive-electro band called C-Mon & Kypski

OUTLINE

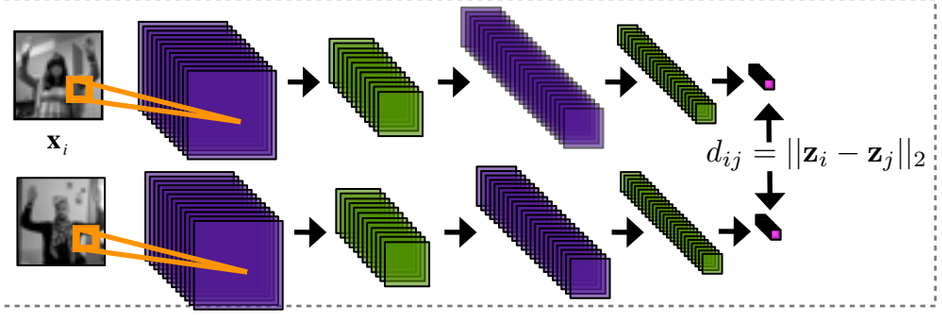
Unsupervised

LSA, Semantic Hashing



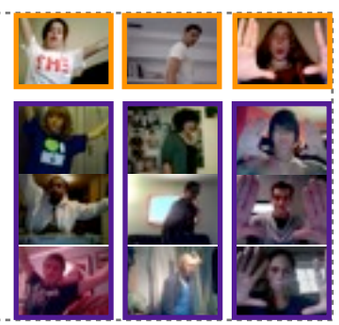
Supervised

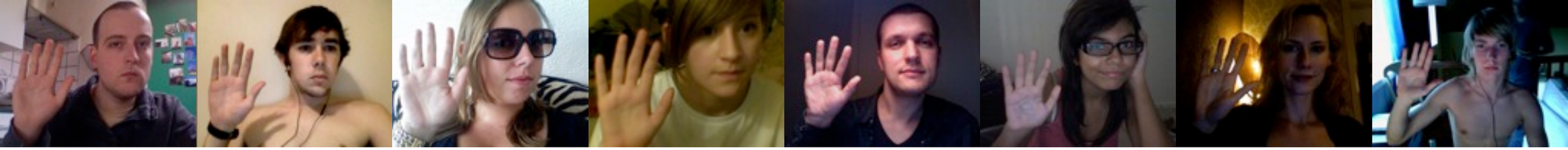
NCA, Nonlinear NCA, DrLIM



Weakly supervised

Applications to pose-sensitive retrieval





LEARNING SIMILARITY

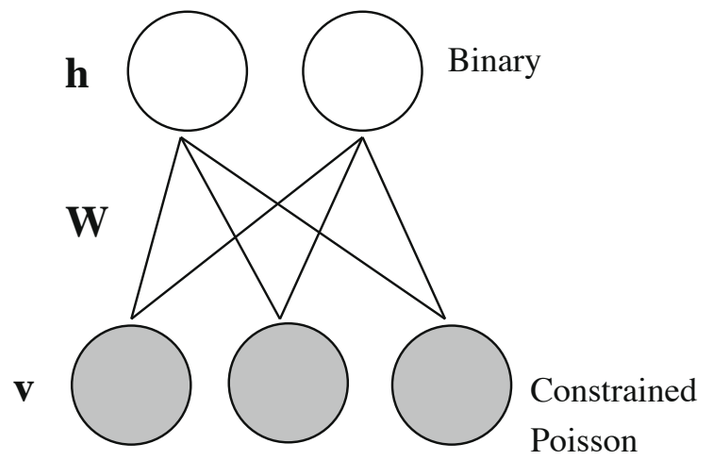
- Pixel distance ≠ semantic similarity
- Computing distances in pixel space is also computationally expensive
- Learning parametric embeddings that are *invariant* to certain input variability
- Today: focus on representations that capture human pose

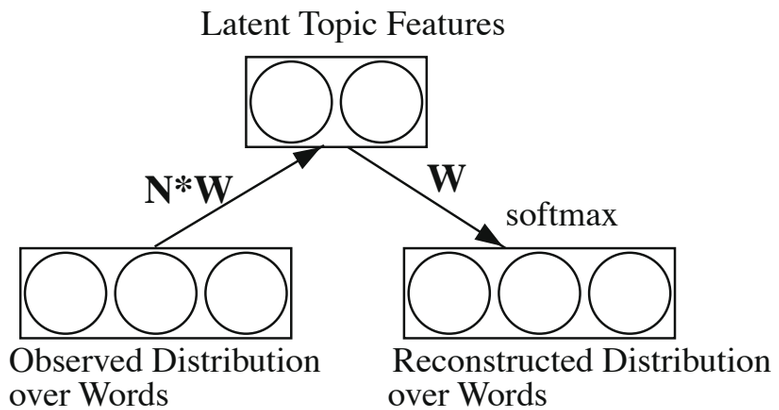
THE UNSUPERVISED APPROACH

- Learn (possibly deep) representations completely unsupervised
  - compute distances between top-level representations
  - representations are usually low-dimensional
- Classical methods: Latent Semantic Analysis (based on SVD), pLSA, LDA
- But directed models don't seem like a natural fit
  - fast inference is important for information retrieval
- Use undirected models in which exact inference is fast
  - Single layer approach by generalizing RBMs: Welling et al. 2005
  - Multi-layer approach: Salakhutdinov and Hinton 2007 "Semantic Hashing"

SEMANTIC HASHING

- Visible layer represents word-count vector of a document
  - "Constrained Poisson Model"
- Learn Constrained Poisson ➛ Binary first layer
- Learn one or more binary RBMs in a "greedy" fashion
- Unroll to a deep autoencoder and "fine-tune" w/ backprop
  - During fine-tuning add Gaussian noise to code layer
  - This forces the codes to be close to binary

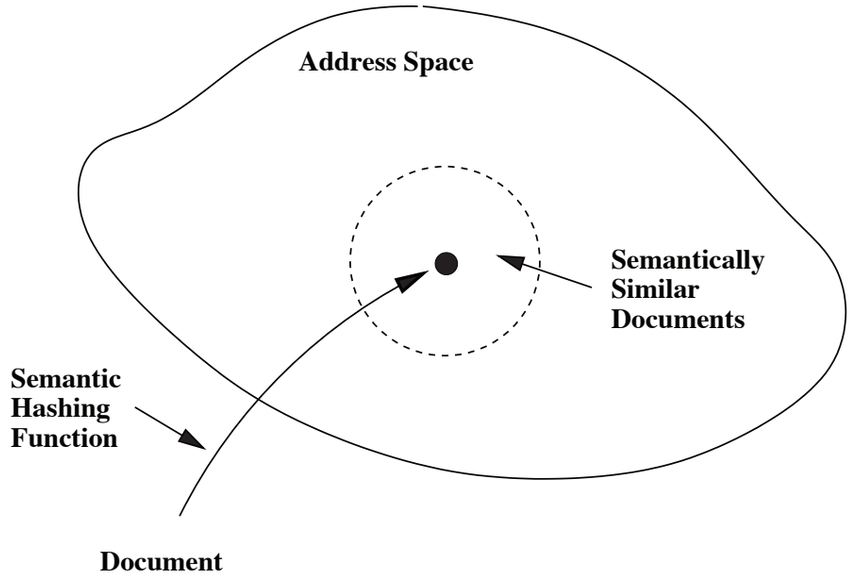


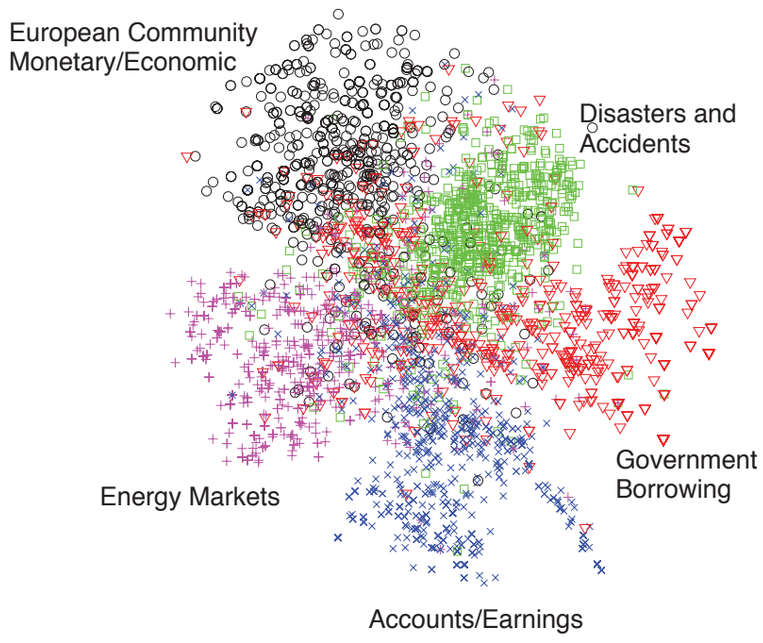


(Figures from R. Salakhutdinov and G. Hinton)

EXTREMELY FAST RETRIEVAL

- Documents are mapped to 20-D binary codes
- Can retrieve similar documents stored at nearby addresses with no search
- Binary LSA significantly reduces performance
  - Not surprising: it has not been optimized to make binary codes perform well
- One weakness: documents with similar addresses have similar content but the converse is not necessarily true
  - Can we use external information (e.g. labels) to pull together codes of similar documents?

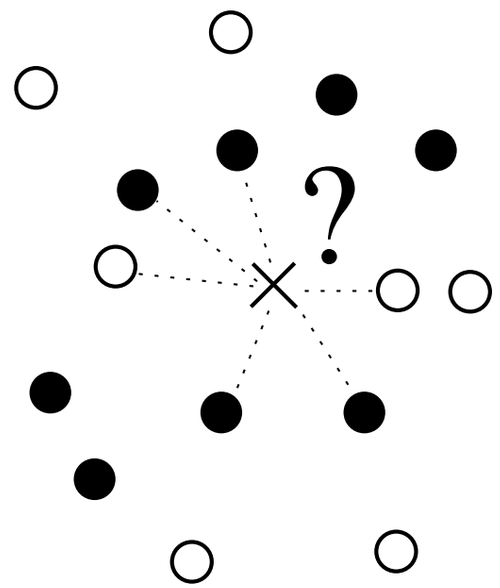




(Figures from Russ Salakhutdinov)

MOTIVATION: KNN CLASSIFICATION

- What is the right distance for KNN classification?
  - the one that optimizes test error!
- Think about approximating this by training error, defined by leave-one-out cross-validation
- Two problems:
  - LOO error is a highly discontinuous function of the distance metric
  - We still need to choose K
- Look for a smoother (or at least continuous) cost function



(Slide from Sam Roweis)

STOCHASTIC NEAREST NEIGHBOUR

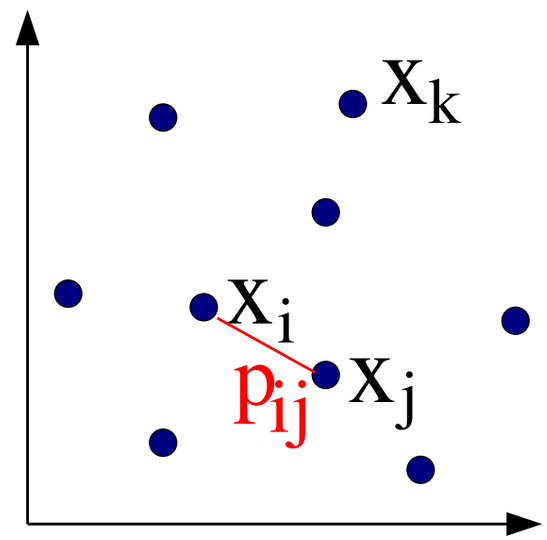
- Instead picking from a fixed set of K nearest neighbours, select a single neighbour stochastically
- Let each point i select other points j as its neighbour with probability p_{ij} based on the softmax of the distance d_{ij}:

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)}$$

where:

$$d_{ij} = ||\mathbf{z}_i - \mathbf{z}_j||_2$$

$$\mathbf{z}_i = f(\mathbf{x}_i|\theta)$$



(Figure from Sam Roweis)

NCA: LOSS

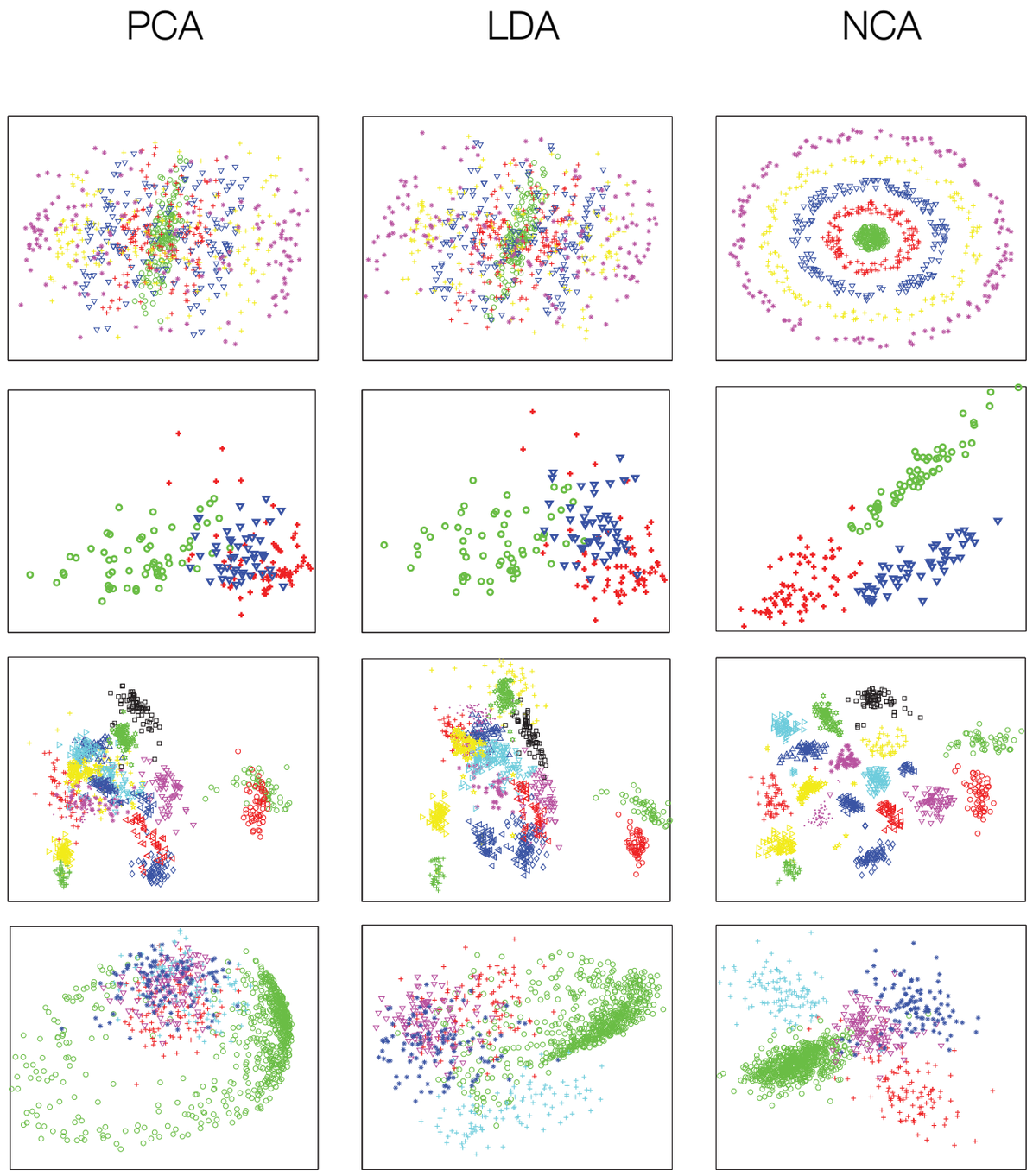
- Maximize the expected number of points correctly classified under this scheme
- This is much smoother than the actual leave-one-out cross-validation error!
- In fact, it is differentiable w.r.t. parameters of mapping
  - can use SGD or other gradient-based optimizer
- And there is no explicit parameter K

$$L_{\mathrm{NCA}} = -\sum_{i=1}^{N} \sum_{j:y_i=y_j} p_{ij}$$

Minimize loss w.r.t. θ

NCA: EMBEDDINGS

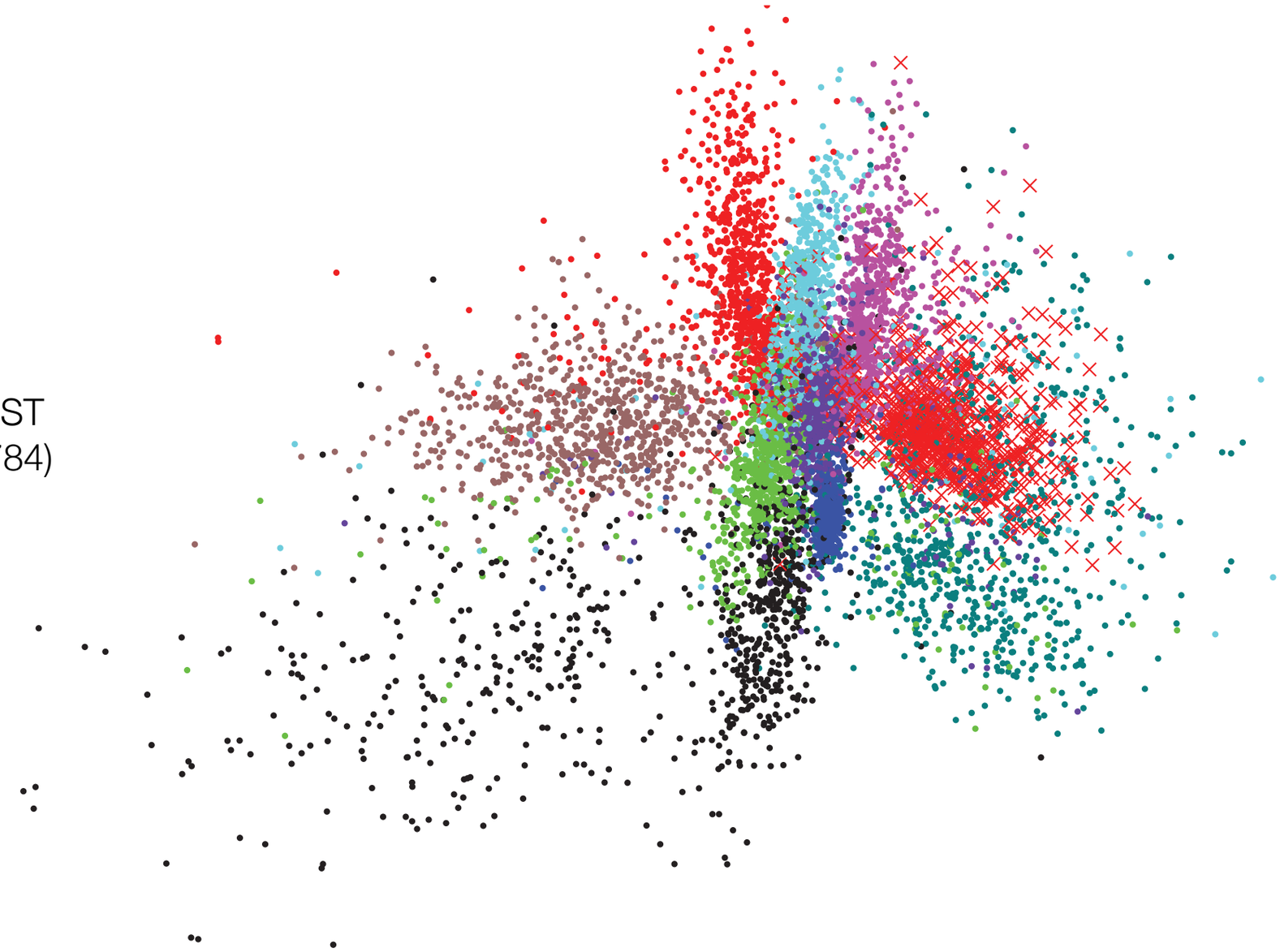
$$f(\mathbf{x}|\theta = A) = A\mathbf{x}$$



(Figures from Goldberger et al.)

NCA: MNIST

MNIST
(D=784)



NONLINEAR NCA

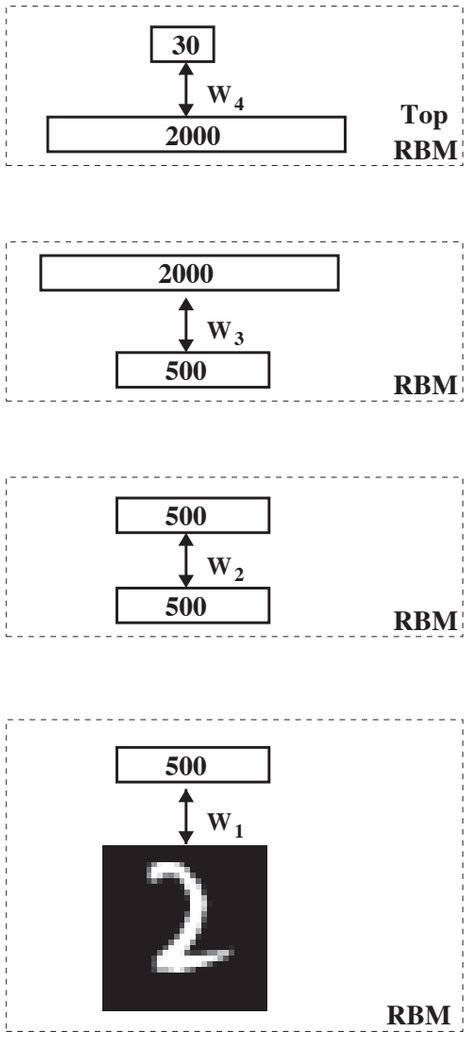
- The original NCA paper (Goldberger et al. 2004) points out that $f\left(\mathbf{x}_i \middle| \theta\right)$ need not be a linear mapping
- Salakhutdinov and Hinton (2007) pre-train with an RBM, then fine-tune with the NCA objective
- Can combine the NCA objective with an Autoencoder objective to regularize:

$$C = \lambda L_{\mathrm{NCA}} + (1-\lambda) L_{AE}$$

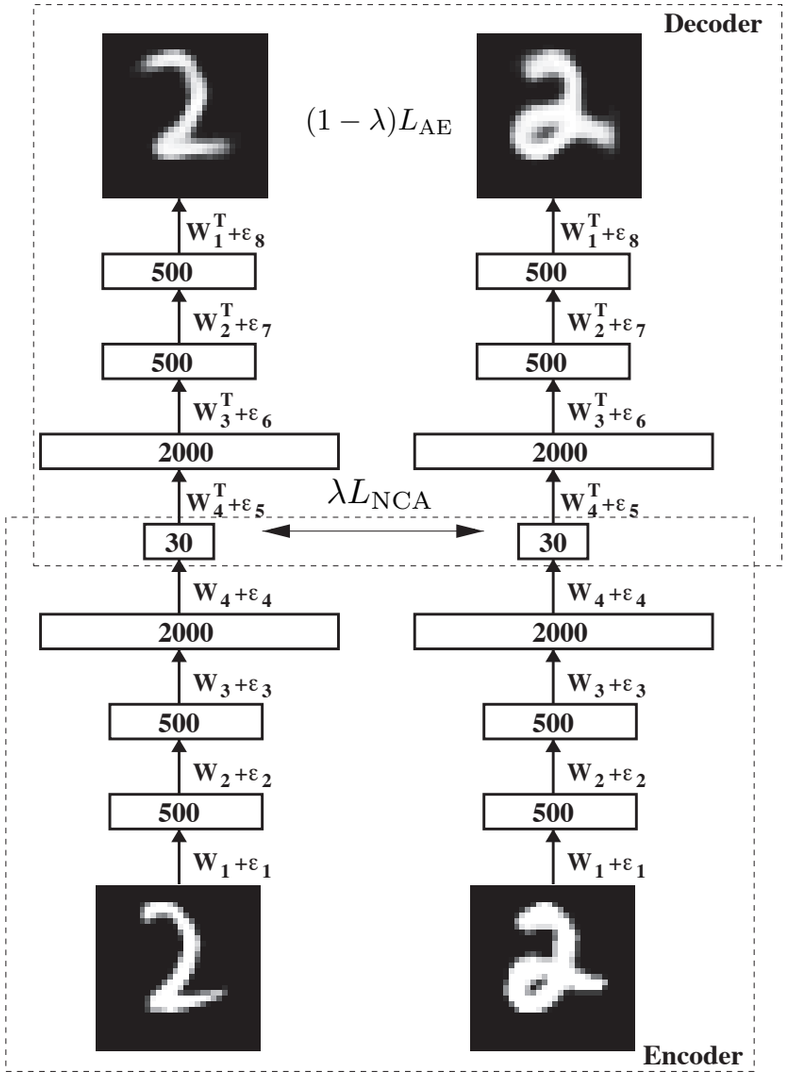
- Can take advantage of unlabeled data!

LEARNING NONLINEAR NCA

Pre-training



Mixed-objective fine-tuning

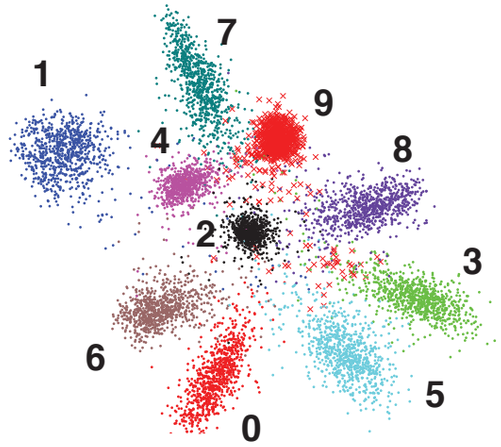


(Figures from R. Salakhutdinov and G. Hinton)

LIMITATIONS OF NCA

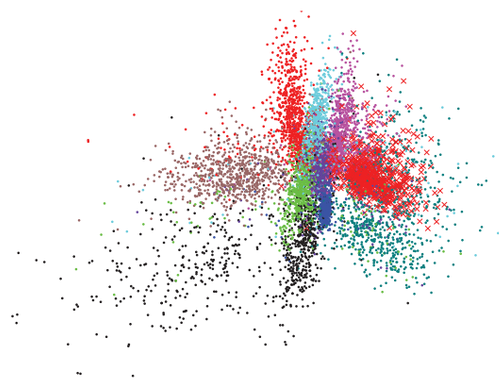
- Despite very nice embeddings (see right) NCA has a quadratic normalization term (must consider all pairs)
  - mini-batch training (approximate)
  - objectives that don't require normalization

Noninear NCA (MNIST)



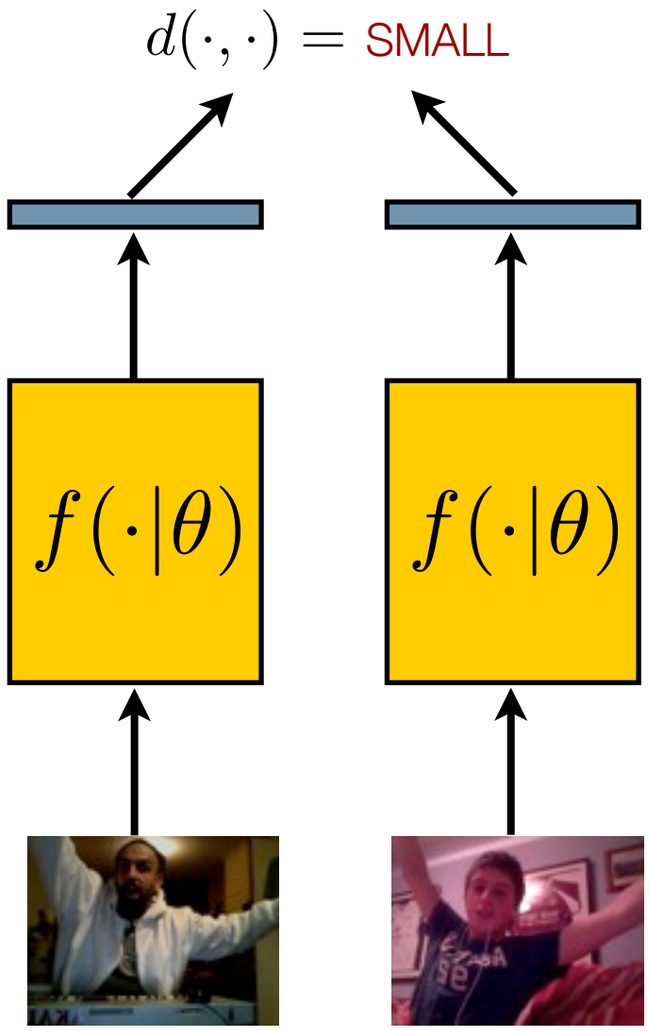
- What about continuous labels?
  - (Goldberger et al. 2004) describe a "soft" form of NCA that can use continuous labels

Linear NCA (MNIST)

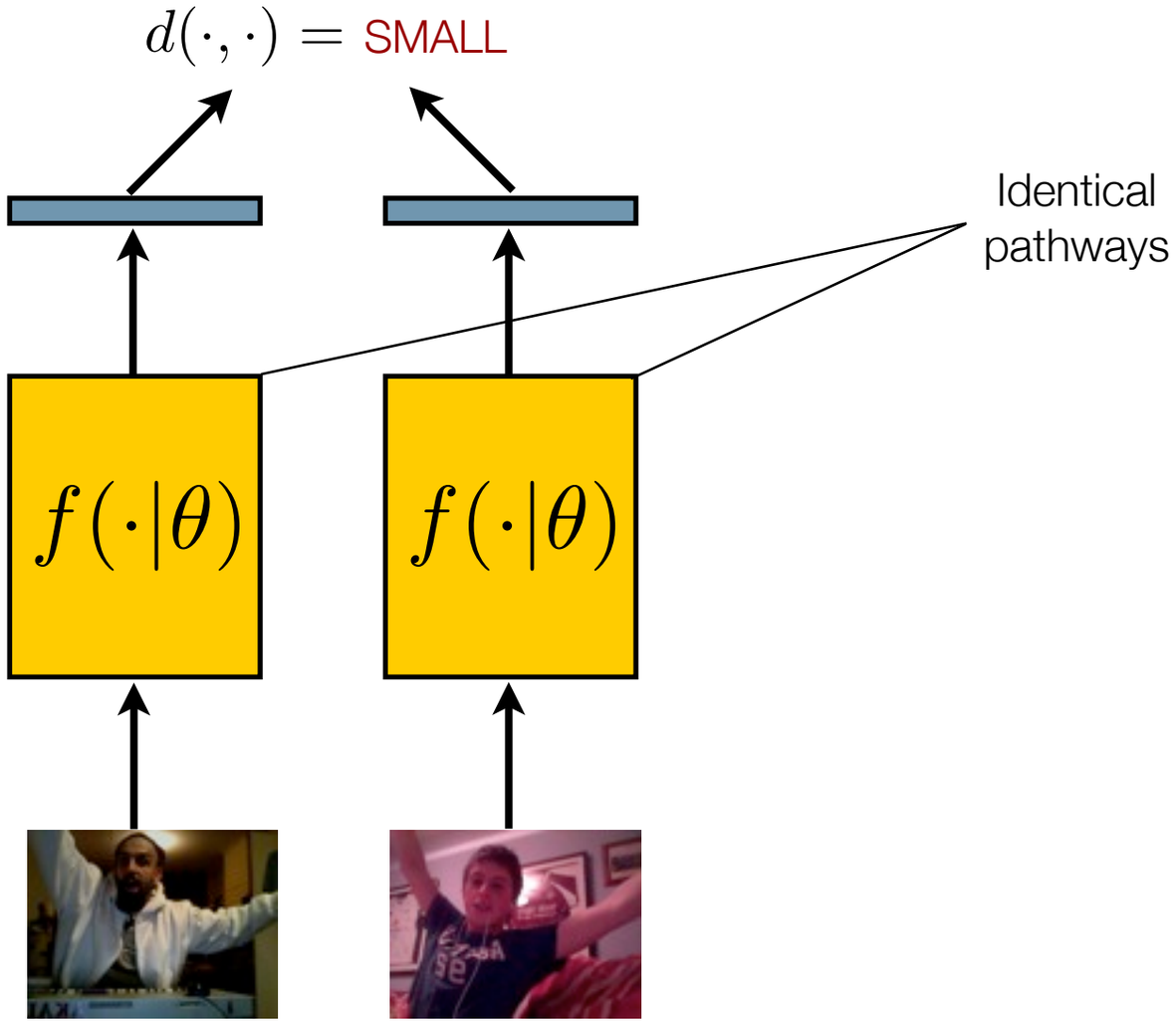


(Figures from R. Salakhutdinov and G. Hinton)

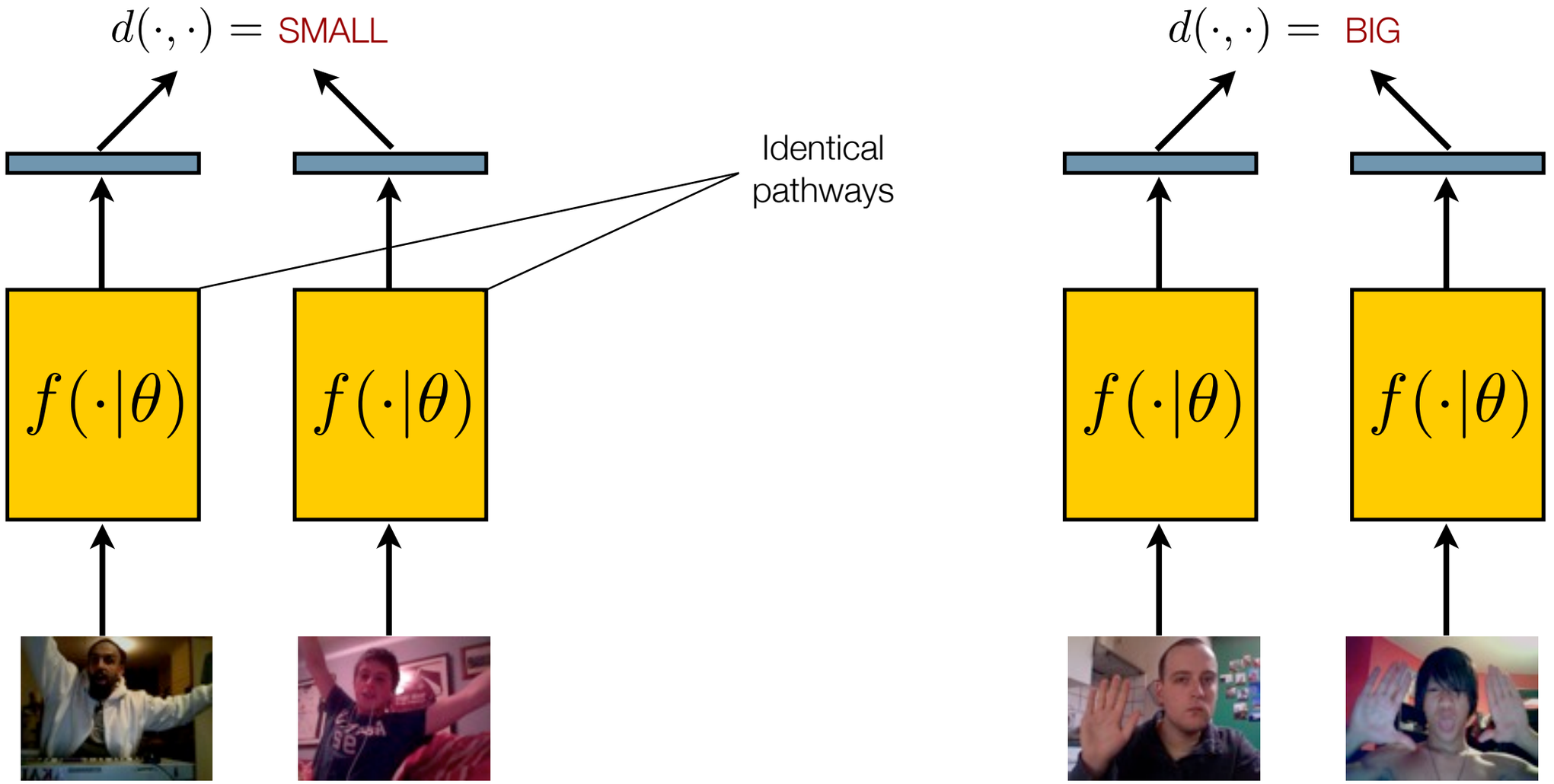
LEARNING EMBEDDINGS WITH A SIAMESE NETWORK



LEARNING EMBEDDINGS WITH A SIAMESE NETWORK

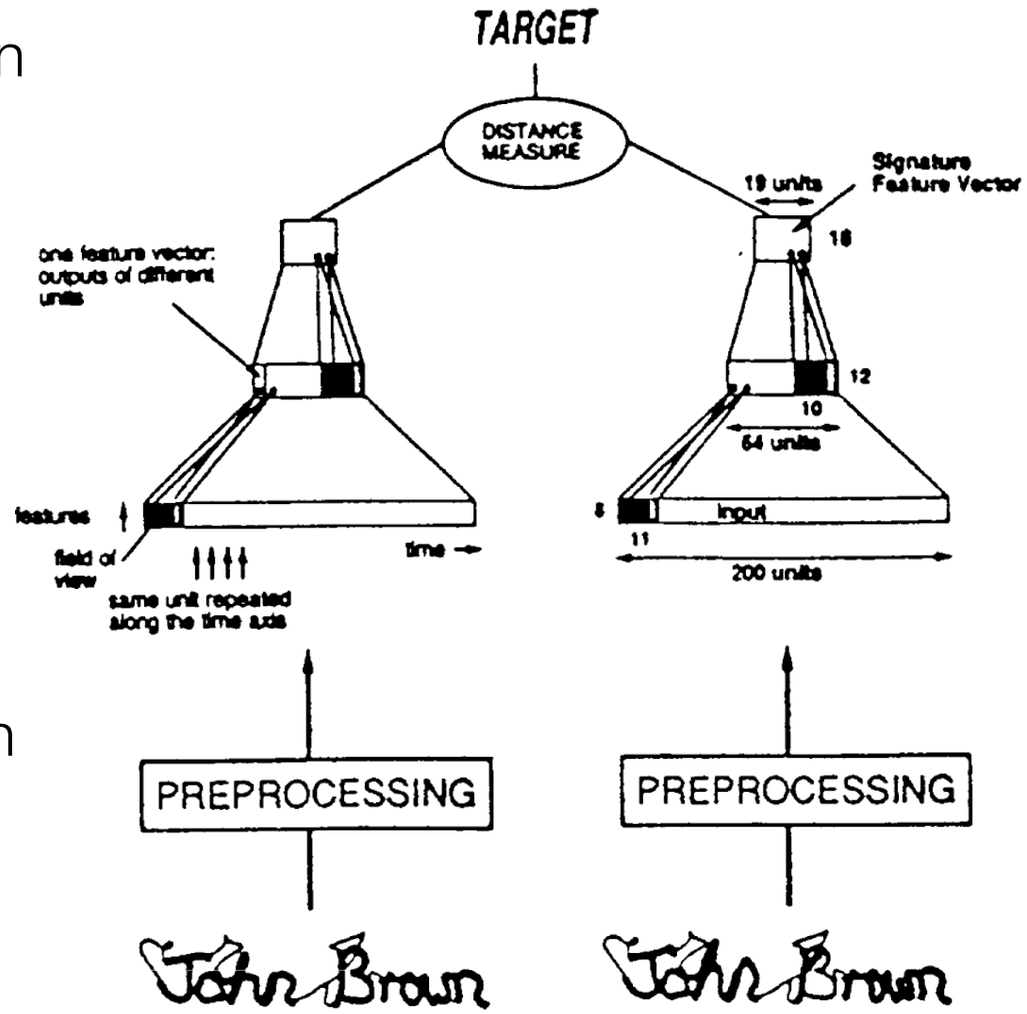


LEARNING EMBEDDINGS WITH A SIAMESE NETWORK



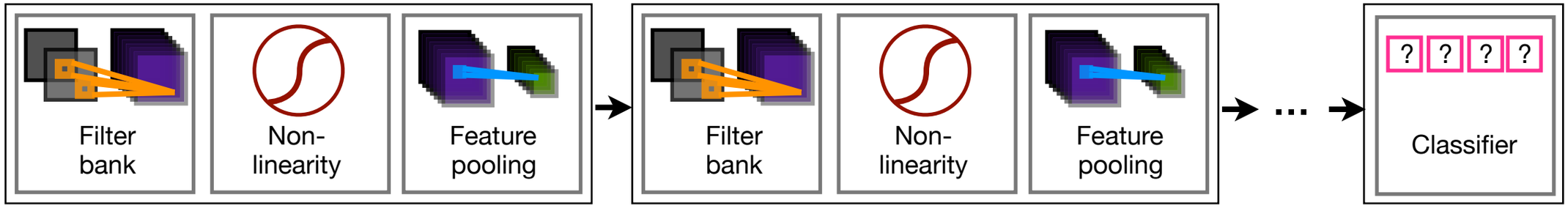
NOT A NEW IDEA (Bromley, Guyon, LeCun, Sackinger, and Shah 1994)

- Architecture proposed for signature verification
  - didn't really get the distance function right
  - learning unstable
  - small (by today's standards) training set
- 1D convolution (TDNN)
- Developed independently elsewhere:
  - Baldi and Chauvin, 1992: fingerprint verification
  - Becker and Hinton, 1992 - discovering depth in random-dot stereograms

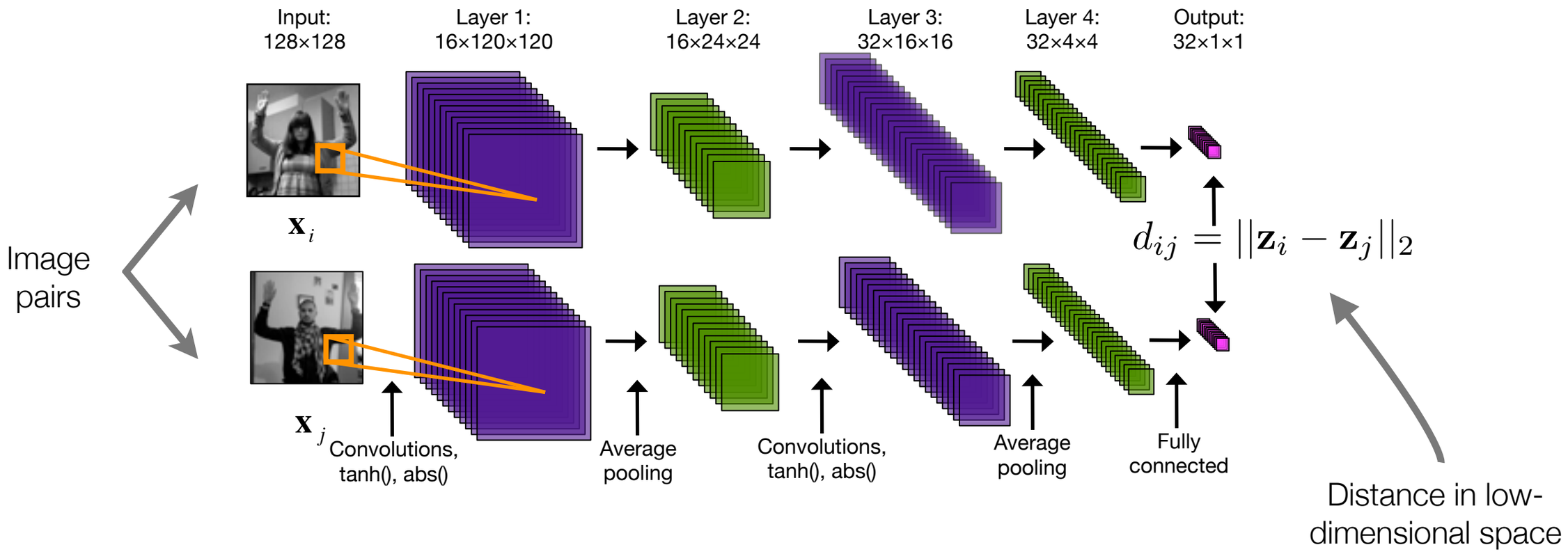


THE EMBEDDING: CONVOLUTIONAL NETWORKS

- Stacking multiple stages of Filter Bank + Non-Linearity + Pooling
- Shared with other approaches (SIFT, GIST, HOG)
- Main difference: Learn the filter banks at every layer



EMBEDDING WITH A SIAMESE CONVOLUTIONAL NET



What's the objective function?

-needs to pull together semantically similar pairs

-needs to push apart semantically dissimilar pairs

(Hadsell, Chopra and LeCun 2006)

DIMENSIONALITY REDUCTION BY LEARNING AN INVARIANT MAPPING (DRLIM)

s_{ij} is a binary indicator

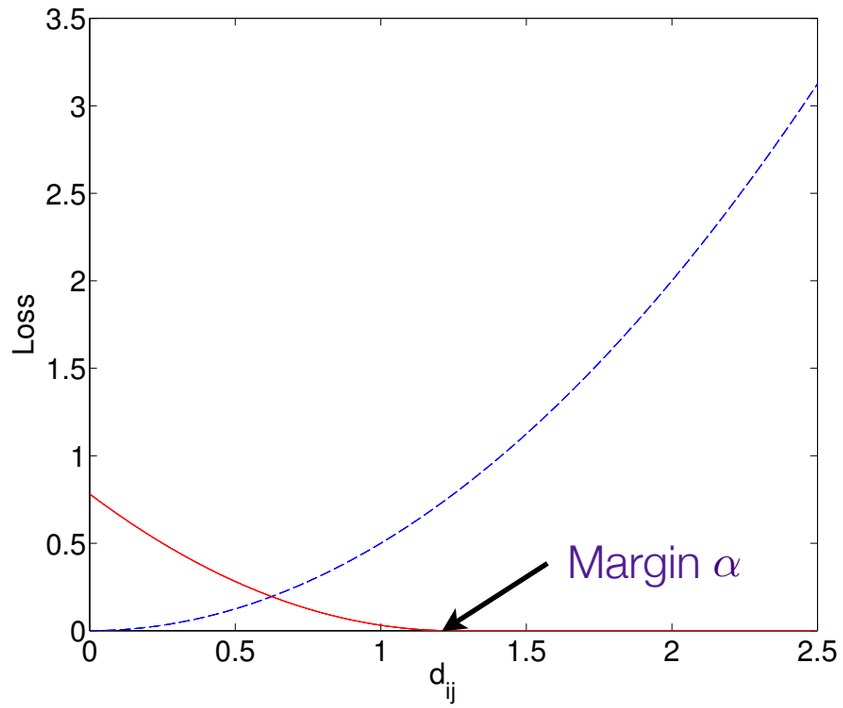
$$L = s_{ij} L_S(\mathbf{x}_i, \mathbf{x}_j) + (1 - s_{ij}) L_D(\mathbf{x}_i, \mathbf{x}_j)$$

Similarity loss

$$L_S(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(d_{ij})^2$$

Dissimilarity loss

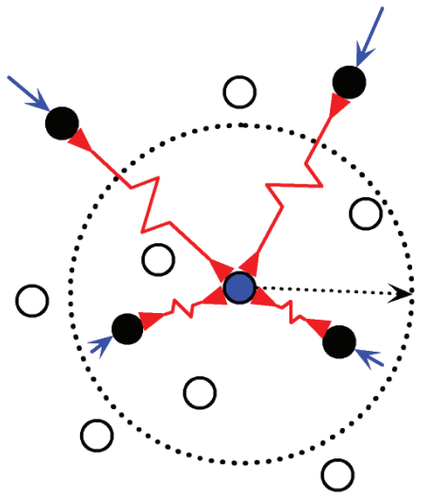
$$L_D(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\left[\max(0, \alpha - d_{ij})\right]^2$$

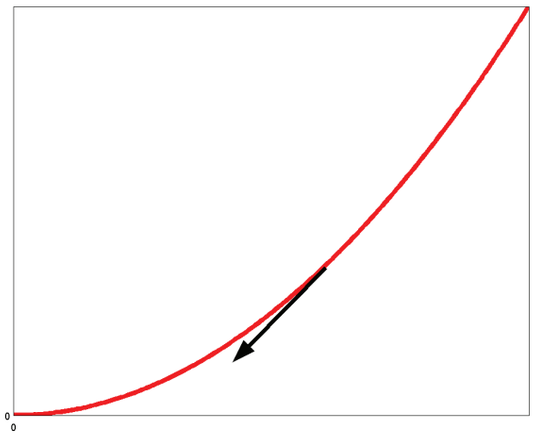


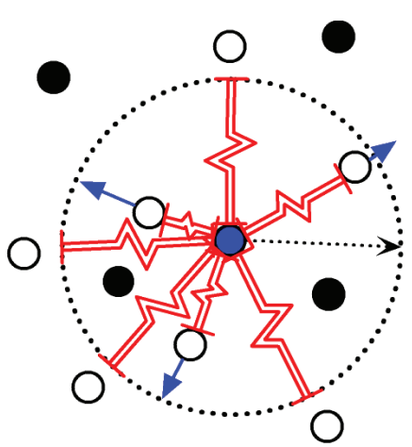
- The similarity loss "pushes together" similar points
- The dissimilarity loss "pulls apart" dissimilar points
  - but only if their distance is within some margin, α

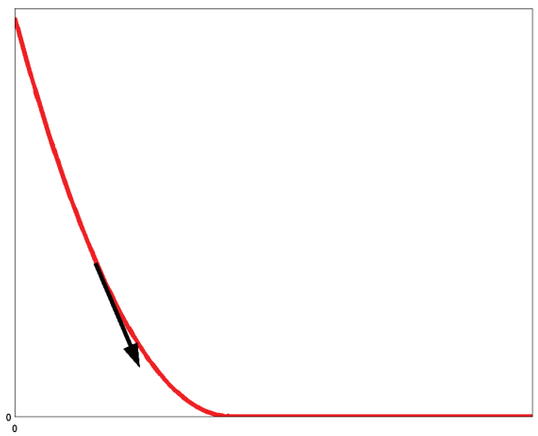
SPRING ANALOGY

- Solid dots are points that are similar to the point in the centre
- Hollow dots are points that are dissimilar to the point in the centre
- Forces acting on the points are shown in blue
  - The length of the arrow represents the strength of the force
- Radius represents the margin, α

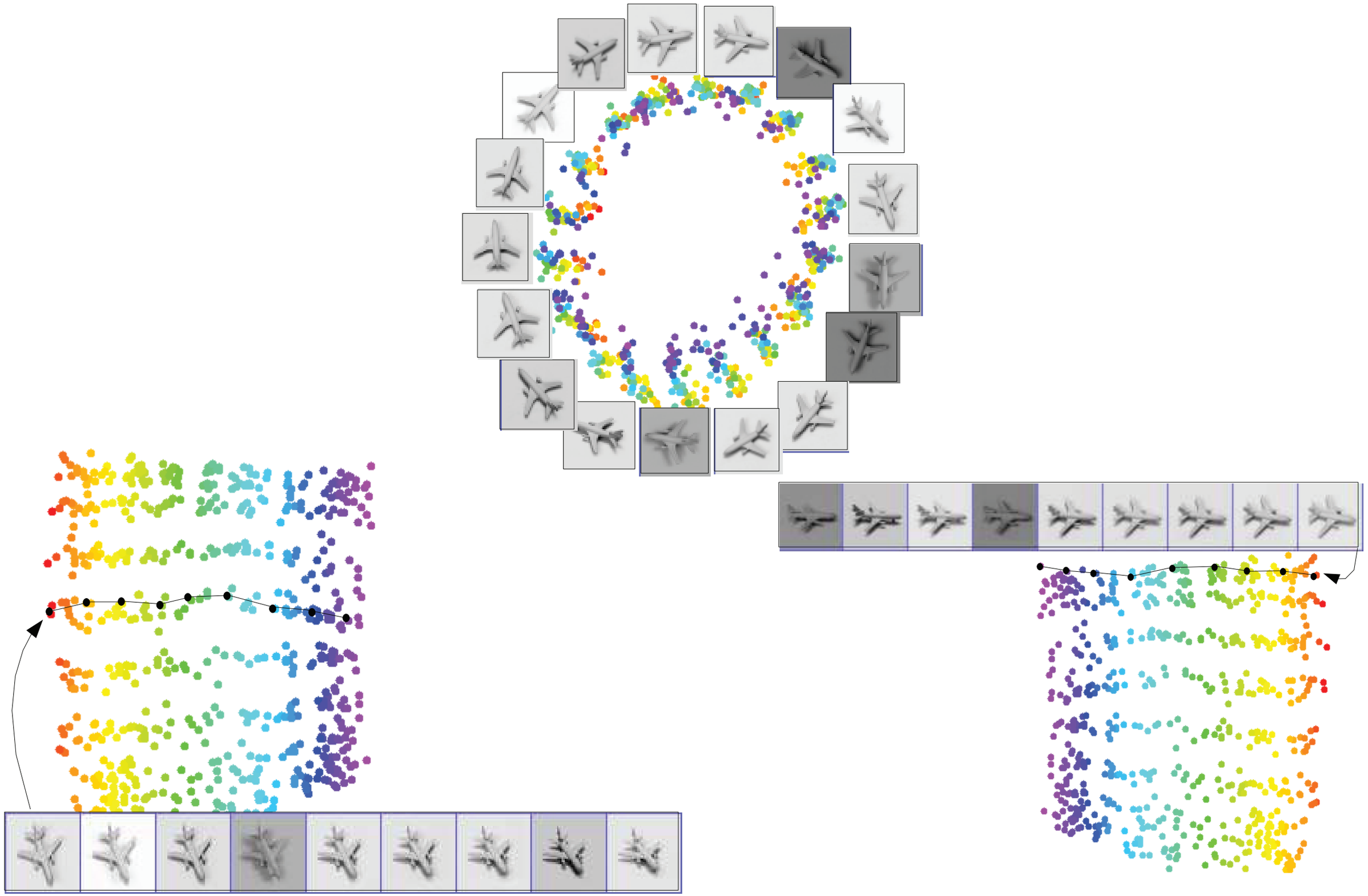








(Figures from Hadsell et al.)



(Figures from Hadsell et al.)

EXISTING PARADIGM: PAIRWISE SIMILARITY

EXISTING PARADIGM: PAIRWISE SIMILARITY

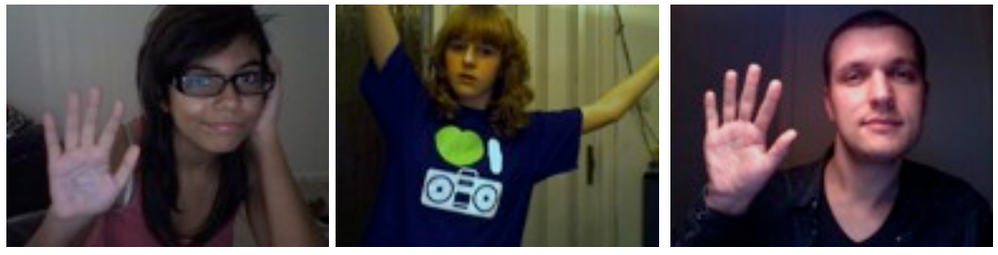
- NCA, DrLIM: binary notion of similarity typically defined by class membership or explicitly constructed neighbourhood graph

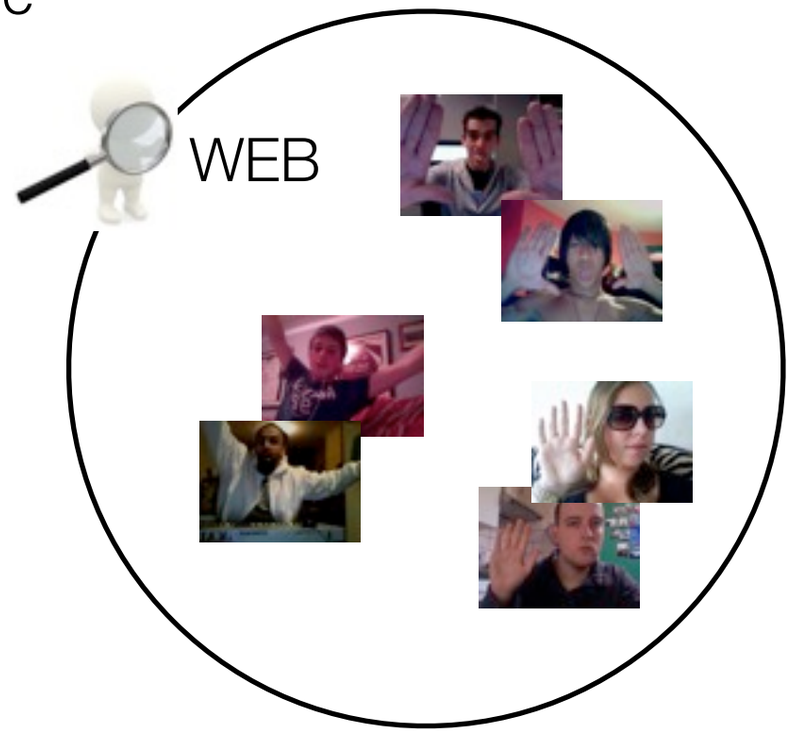
EXISTING PARADIGM: PAIRWISE SIMILARITY

- NCA, DrLIM: binary notion of similarity typically defined by class membership or explicitly constructed neighbourhood graph
- Defining pairwise similarity is difficult and inconsistent across observers

EXISTING PARADIGM: PAIRWISE SIMILARITY

- NCA, DrLIM: binary notion of similarity typically defined by class membership or explicitly constructed neighbourhood graph
- Defining pairwise similarity is difficult and inconsistent across observers
- Despite crowd-sourcing platforms (e.g. Amazon Mechanical Turk) gathering semantically similar pairs of images is expensive





HANDS BY HAND

- One solution is to turn to synthetic data (e.g. Shakhnarovich et al. 2003, Jain et al. 2008)

- Difficult to generalize to real (e.g. “YouTube” settings)

- Another solution: ask people to label heads and hands (Spiro et al. 2010) or superimpose articulated skeletons (Bourdev et al. 2009)

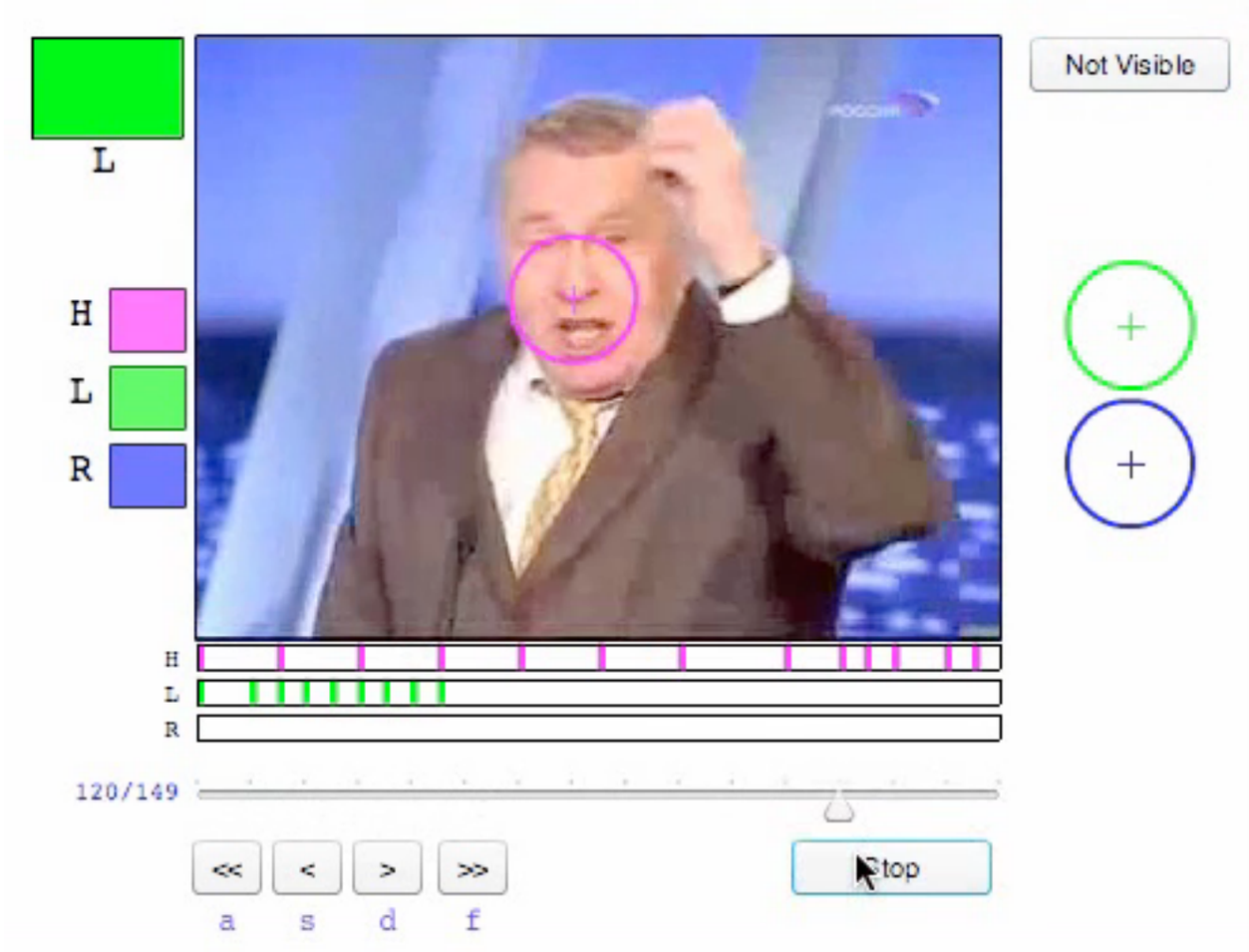
(Spiro, Taylor, Williams and Bregler ACVHL 2010)

HANDS BY HAND

- One solution is to turn to synthetic data (e.g. Shakhnarovich et al. 2003, Jain et al. 2008)

- Difficult to generalize to real (e.g. “YouTube” settings)

- Another solution: ask people to label heads and hands (Spiro et al. 2010) or superimpose articulated skeletons (Bourdev et al. 2009)



(Spiro, Taylor, Williams and Bregler ACVHL 2010)

(Taylor, Spiro, Williams, Fergus and Bregler NIPS 2010)

NONPARAMETRIC POSE ESTIMATION

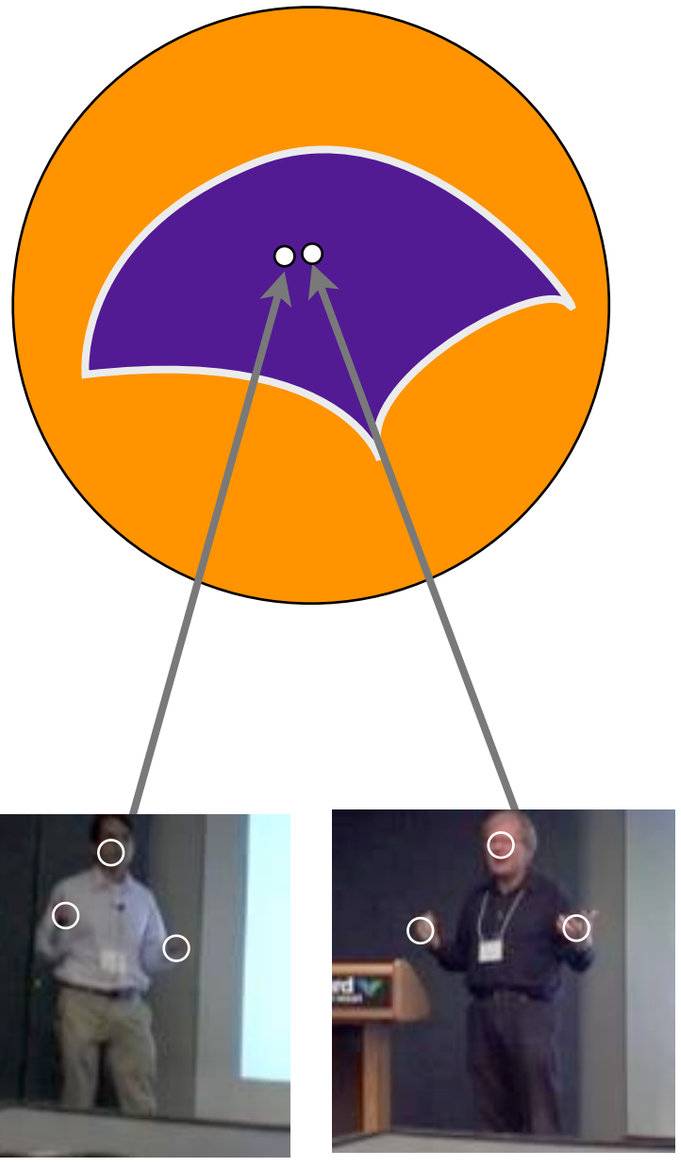
(Taylor, Spiro, Williams, Fergus and Bregler NIPS 2010)

NONPARAMETRIC POSE ESTIMATION

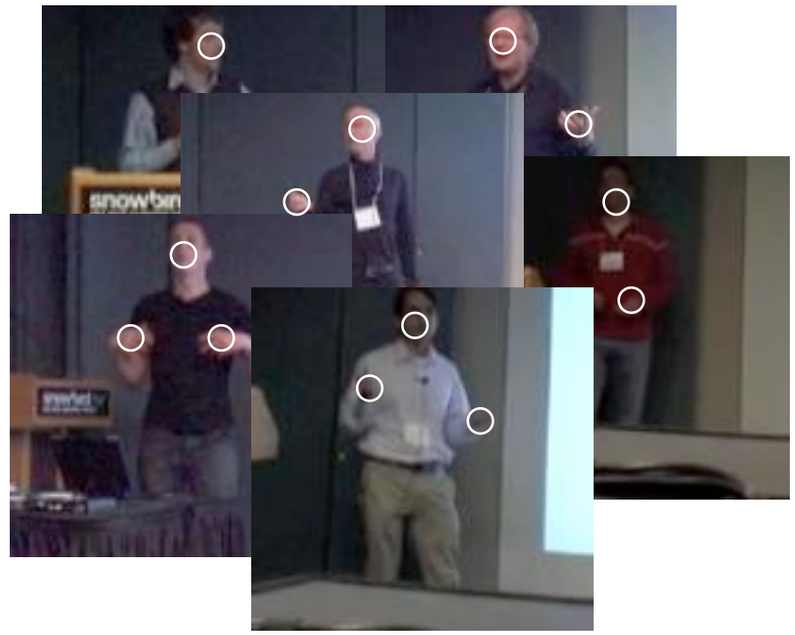
Database

(Taylor, Spiro, Williams, Fergus and Bregler NIPS 2010)

NONPARAMETRIC POSE ESTIMATION



Database



(Taylor, Spiro, Williams, Fergus and Bregler NIPS 2010)

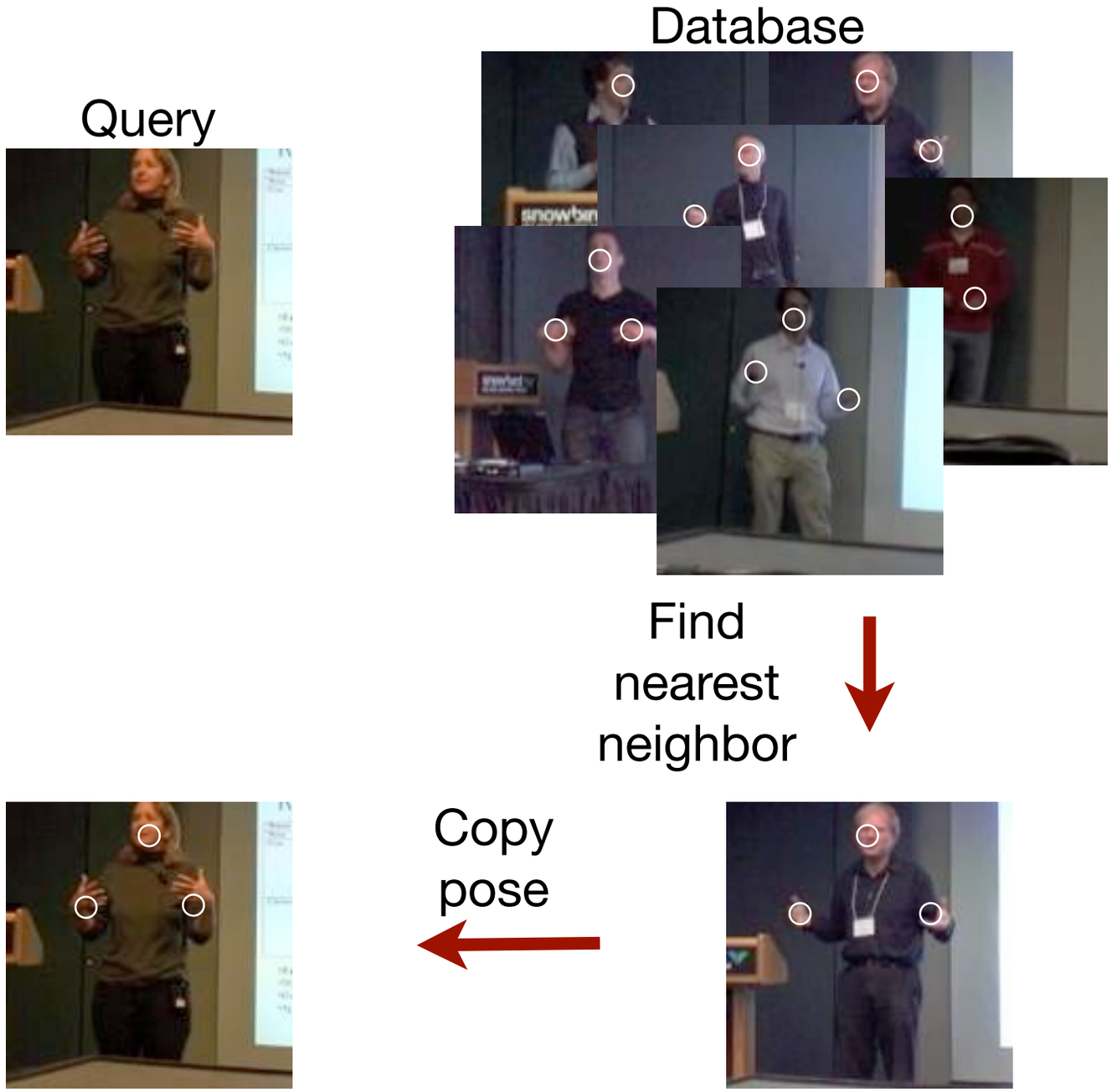
NONPARAMETRIC POSE ESTIMATION

- If we have a database of images labeled with 2D or 3D pose information - we can do non-parametric pose estimation

(Taylor, Spiro, Williams, Fergus and Bregler NIPS 2010)

NONPARAMETRIC POSE ESTIMATION

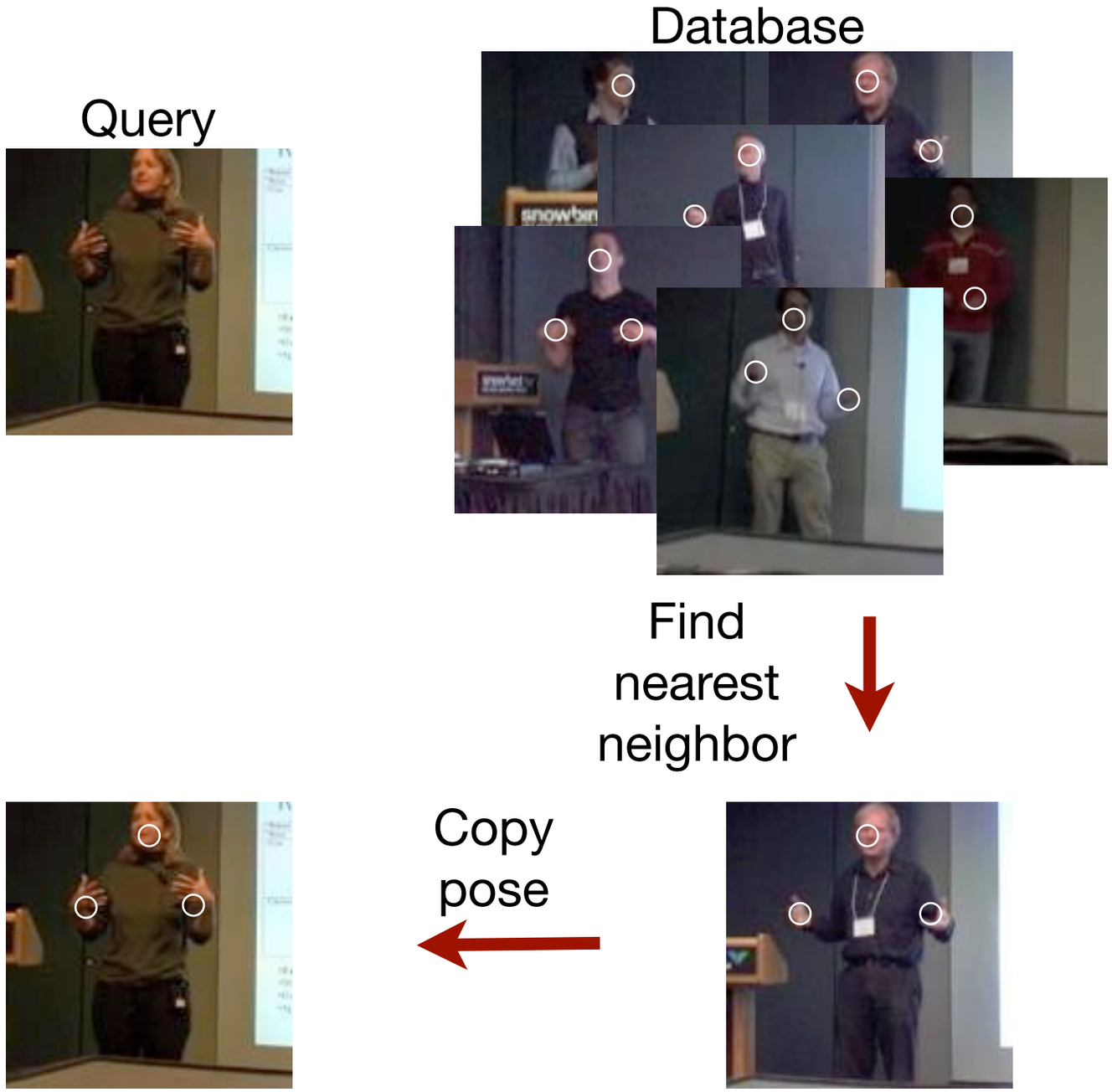
- If we have a database of images labeled with 2D or 3D pose information - we can do non-parametric pose estimation



(Taylor, Spiro, Williams, Fergus and Bregler NIPS 2010)

NONPARAMETRIC POSE ESTIMATION

- If we have a database of images labeled with 2D or 3D pose information - we can do non-parametric pose estimation
- Nearest neighbor lookup must be quick (e.g. performed in a low-dimensional space)

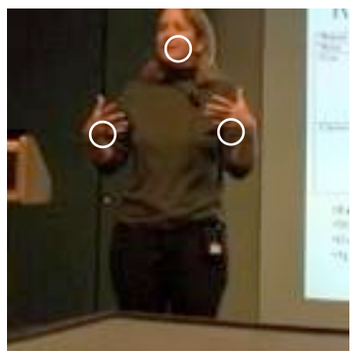


(Taylor, Spiro, Williams, Fergus and Bregler NIPS 2010)

NONPARAMETRIC POSE ESTIMATION

- If we have a database of images labeled with 2D or 3D pose information - we can do non-parametric pose estimation
- Nearest neighbor lookup must be quick (e.g. performed in a low-dimensional space)
- It also must be informative of pose and invariant to clothing, lighting, scale, and other appearance changes

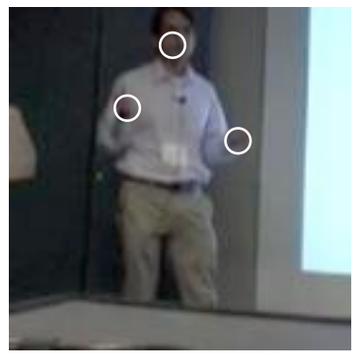
Find nearest neighbor



Copy pose

NCA REGRESSION

$$L_{\text{NCAR}} = \sum_{i=1}^{N} \sum_{j} p_{ij} ||\mathbf{y}_i - \mathbf{y}_j||_2^2$$

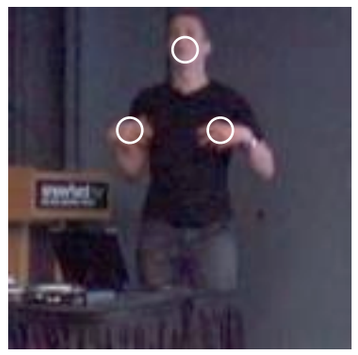


$\mathbf{x}_i$

$$\mathbf{y}_i = [48.2, 46.3, \ldots, 63.3]^T$$

Minimize loss w.r.t. θ

Pay a high cost for “neighbours” in feature space that are far away in pose space



$\mathbf{x}_j$

$$\mathbf{y}_i = [54.4, 45.8, \ldots, 64.1]^T$$

SNOWBIRD DATASET

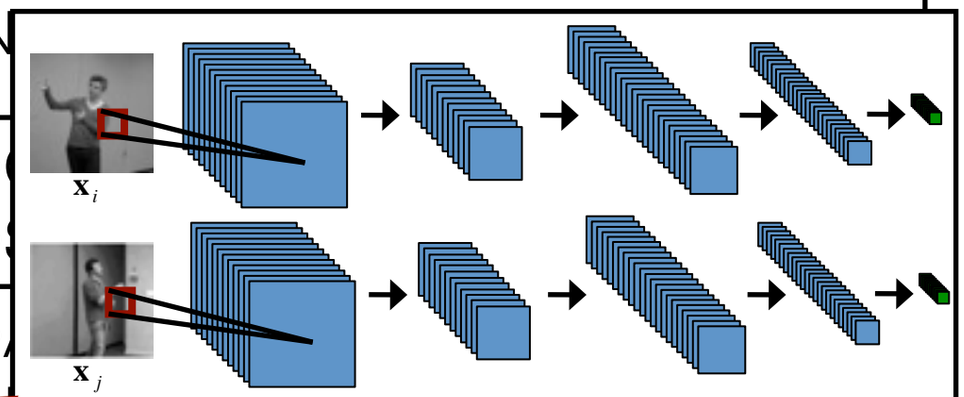
- We digitally recorded all contributing and invited speakers at the 2010 Snowbird workshop
- After each session of talks, blocks of 150 frames were distributed as Human Intelligence Tasks (HITs) on Amazon Mechanical Turk
- Split speakers into 39k training examples, 37k test examples (no overlap in identity)

COMPARISON OF APPROACHES

Pixel distance	Not practical
GIST	•Global representation of image •Still not practical
Linear NCA regression (NCAR)	•Applied to pre-computed GIST •Fit by conjugate gradient
Convolutional NCAR (C-NCAR)	•Convolutions applied to pixels •Tanh(),Abs(),Average
DrLIM Regression (DrLIMR)	•Similar to NCAR but adds an explicit contrastive loss
Convolutional DrLIMR (C-DrLIMR)	•Similar to C-NCAR but adds an explicit contrastive loss

COMPARISON OF APPROACHES

Pixel distance	N
GIST	• •
Linear NCA regression (NCAR)	• A Fit by conjugate gradient
Convolutional NCAR (C-NCAR)	•Convolutions applied to pixels •Tanh(),Abs(),Average
DrLIM Regression (DrLIMR)	•Similar to NCAR but adds an explicit contrastive loss
Convolutional DrLIMR (C-DrLIMR)	•Similar to C-NCAR but adds an explicit contrastive loss

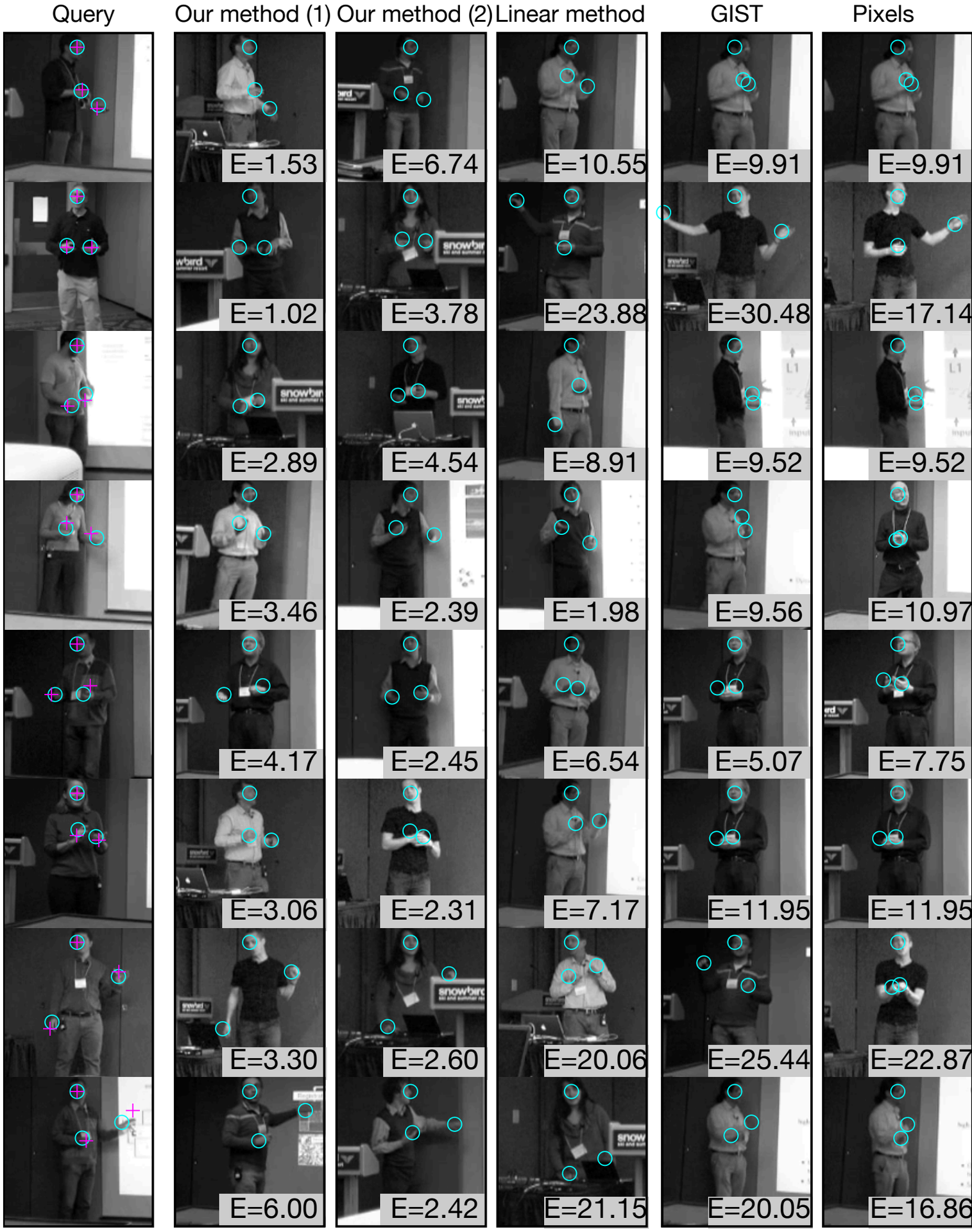


COMPARISON OF APPROACHES

Pixel distance	Not practical
GIST	•Global representation of image •Still not practical
Linear NCA regression (NCAR)	•Applied to pre-computed GIST •Fit by conjugate gradient
Convolutional NCAR (C-NCAR)	•Convolutions applied to pixels •Tanh(),Abs(),Average
DrLIM Regression (DrLIMR)	•Similar to NCAR but adds an explicit contrastive loss
Convolutional DrLIMR (C-DrLIMR)	•Similar to C-NCAR but adds an explicit contrastive loss

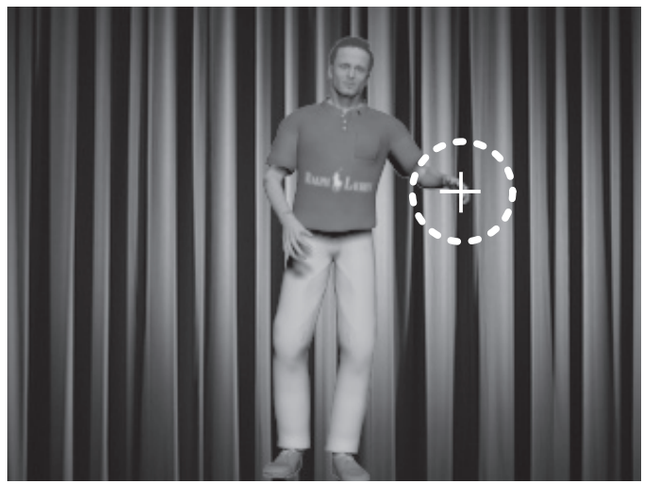
LABELING POSE

- Both Pixel-based matching and GIST focus on scene content, lighting
- Our method learns invariance to background, focuses on pose
- Though trained on hands relative to head, seems to capture something more substantial about body pose



RESULTS

Embedding	Input	Code size	Err-SY	Err-RE
None	Pixels	16384	32.86	25.12
None	GIST	512	47.41	25.30
PCA	GIST	128	47.17	24.85
PCA	GIST	32	48.99	25.74
NCAR	GIST	32	34.21	24.93
NCAR	LCN	32	32.90	23.15
S-DrLIM	GIST	32	37.80	25.19
Boost-SSC	LCN	32	34.80	22.65
PSE(b)	LCN	32	28.95	**16.41**
PSE(o)	LCN	32	**25.40**	19.61



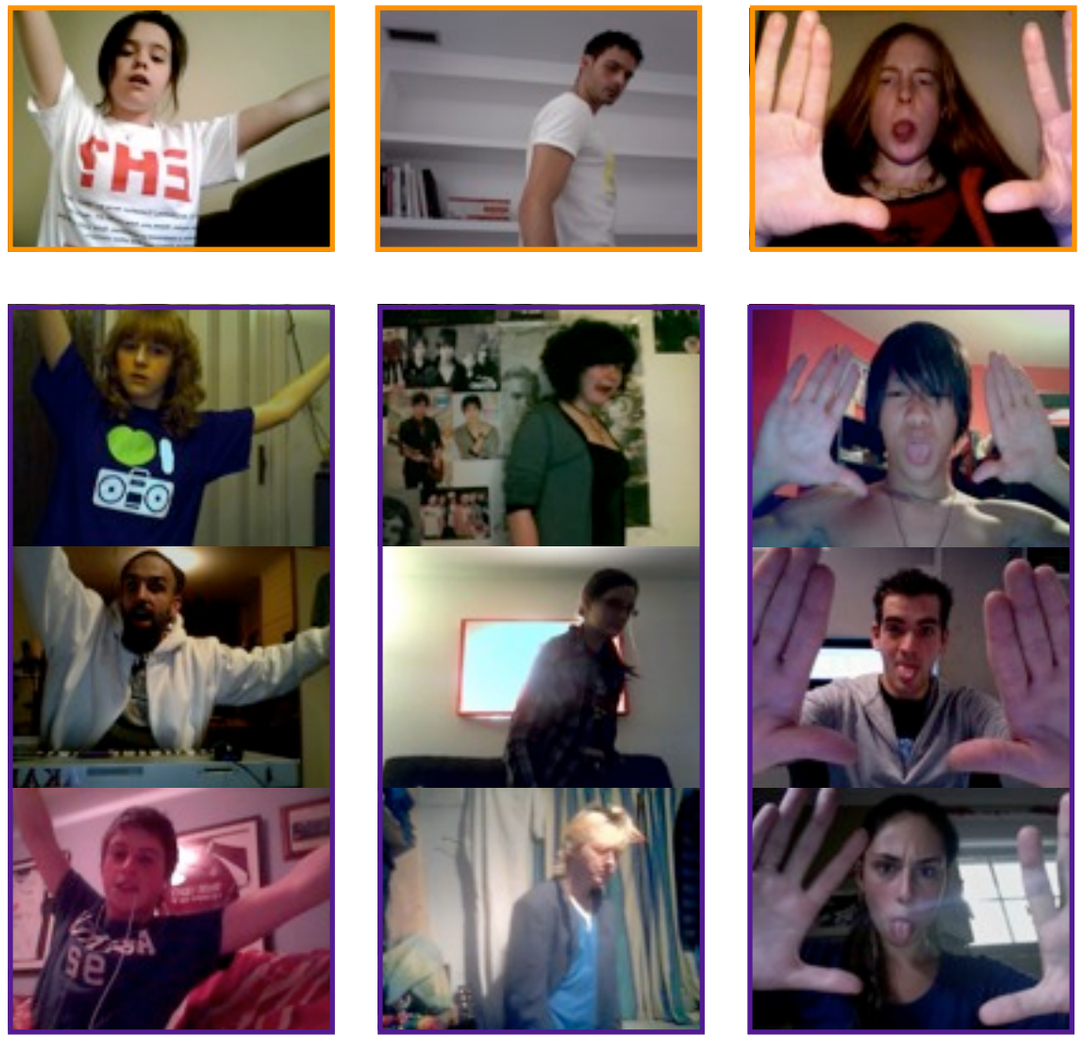
25.4 pixel error

16.4 pixel error

Can we get away with not asking people to provide explicit labels of body parts?

LEARNING INVARIANCE THROUGH IMITATION

- A new paradigm for learning invariant mappings: *imitation*
- People have a remarkable ability to mimic image content
- Exploit the abundance of webcams to quickly crowd-source a massive dataset of people in similar pose
- *Active* crowd-sourcing



A NEW TAKE ON TEMPORAL COHERENCE

A NEW TAKE ON TEMPORAL COHERENCE

- How do we select the images people are asked to imitate?

A NEW TAKE ON TEMPORAL COHERENCE

- How do we select the images people are asked to imitate?
- Temporal coherence in video can increase the number of pairwise similarities and add graded similarity

A NEW TAKE ON TEMPORAL COHERENCE

- How do we select the images people are asked to imitate?
- Temporal coherence in video can increase the number of pairwise similarities and add graded similarity

A NEW TAKE ON TEMPORAL COHERENCE

- How do we select the images people are asked to imitate?
- Temporal coherence in video can increase the number of pairwise similarities and add graded similarity

A NEW TAKE ON TEMPORAL COHERENCE

- How do we select the images people are asked to imitate?
- Temporal coherence in video can increase the number of pairwise similarities and add graded similarity

A NEW TAKE ON TEMPORAL COHERENCE

- How do we select the images people are asked to imitate?
- Temporal coherence in video can increase the number of pairwise similarities and add graded similarity

A NEW TAKE ON TEMPORAL COHERENCE

- How do we select the images people are asked to imitate?
- Temporal coherence in video can increase the number of pairwise similarities and add graded similarity
- Video is used only as a source of *seed* images
- Our model learns only from user-contributed imitations

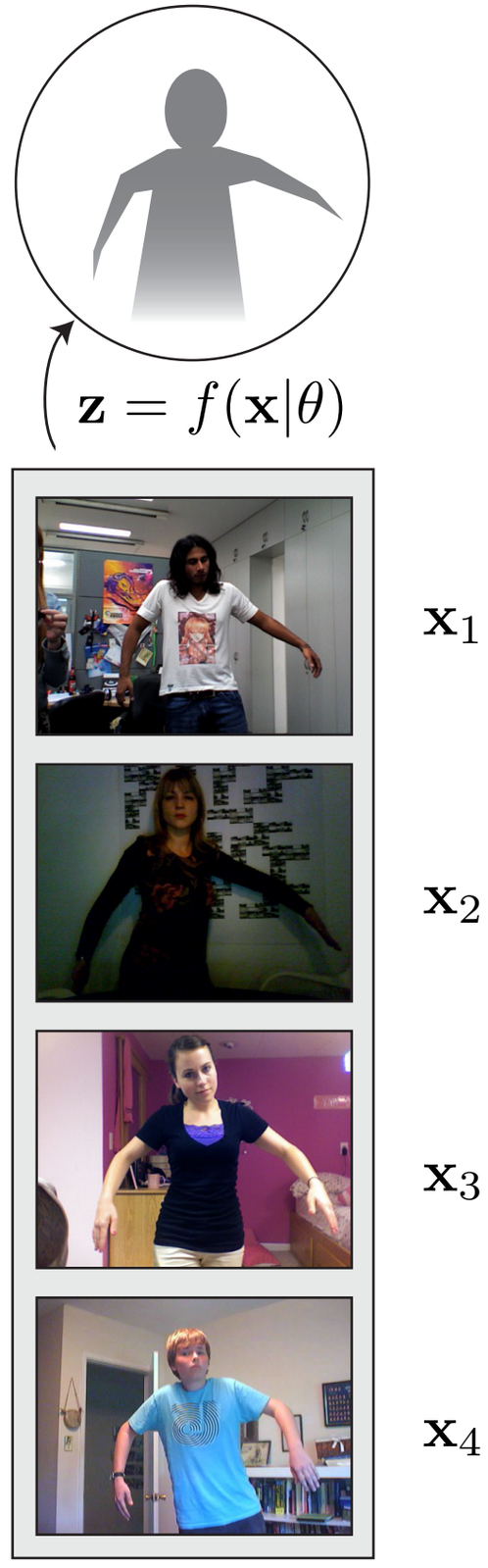
FORMALIZING THE PROBLEM

- Each image, $\mathbf{x}_i$, has an associated *seed* label, y_i
- We seek to learn a mapping:

$$\mathbf{z}_i = f(\mathbf{x}_i|\theta)$$ Can be linear or nonlinear

such that if $\mathbf{x}_i$ and $\mathbf{x}_j$ come from nearby seed images, then

$$d_{ij} = ||\mathbf{z}_i - \mathbf{z}_j||_2$$ will be small.



DIMENSIONALITY REDUCTION BY LEARNING AN INVARIANT MAPPING (DRLIM)

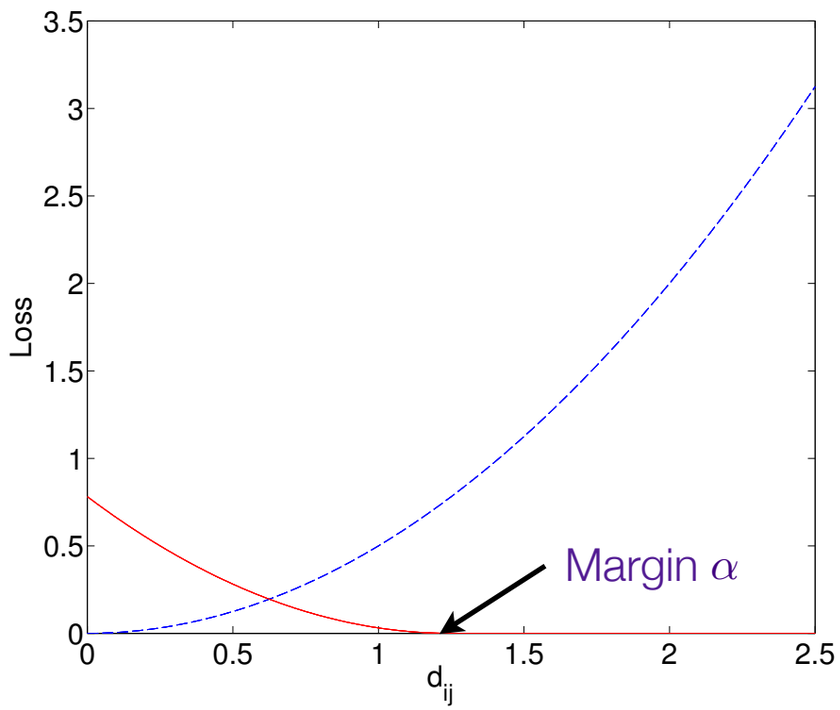
$$L = s_{ij} L_S(\mathbf{x}_i, \mathbf{x}_j) + (1 - s_{ij}) L_D(\mathbf{x}_i, \mathbf{x}_j)$$

Similarity loss

Dissimilarity loss

$$L_S(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(d_{ij})^2$$

$$L_D(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\left[\max(0, \alpha - d_{ij})\right]^2$$



DIMENSIONALITY REDUCTION BY LEARNING AN INVARIANT MAPPING (DRLIM)

$$L = s_{ij} L_S(\mathbf{x}_i, \mathbf{x}_j) + \delta(s_{ij}, 0) L_D(\mathbf{x}_i, \mathbf{x}_j)$$

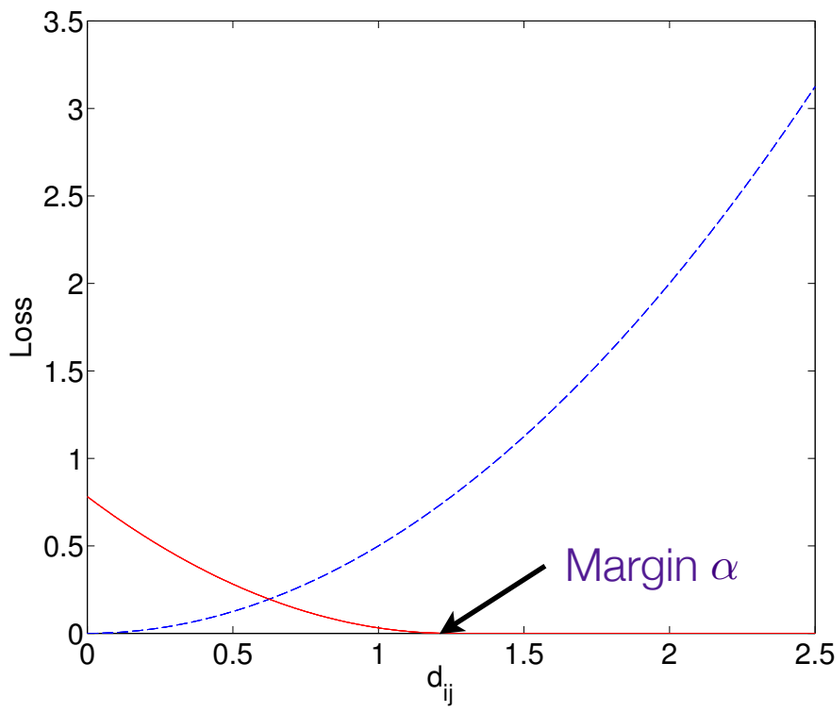
s_{ij} no longer binary

Similarity loss

Dissimilarity loss

$$L_S(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(d_{ij})^2$$

$$L_D(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\left[\max(0, \alpha - d_{ij})\right]^2$$



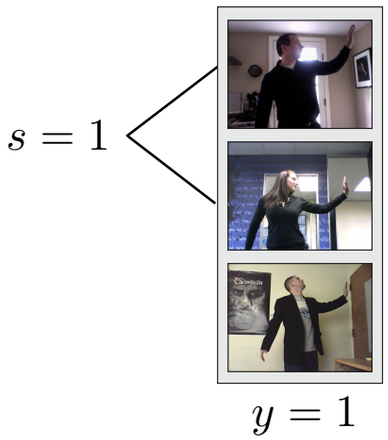
FROM DISCRETE LABELS TO SIMILARITY

- In addition to the learned mapping, f, we require a mapping from discrete seed identity to a real-valued similarity score
- Simplest example: $s_{ij} = (1 + |y_i - y_j|)^{-1}$

Skip convnet intro

FROM DISCRETE LABELS TO SIMILARITY

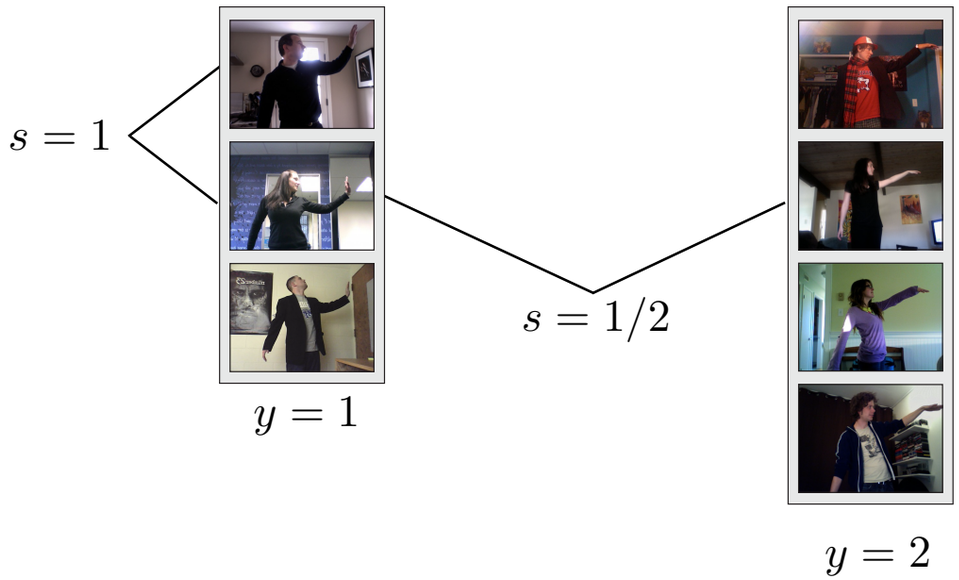
- In addition to the learned mapping, f, we require a mapping from discrete seed identity to a real-valued similarity score
- Simplest example: $s_{ij} = (1 + |y_i - y_j|)^{-1}$



Skip convnet intro

FROM DISCRETE LABELS TO SIMILARITY

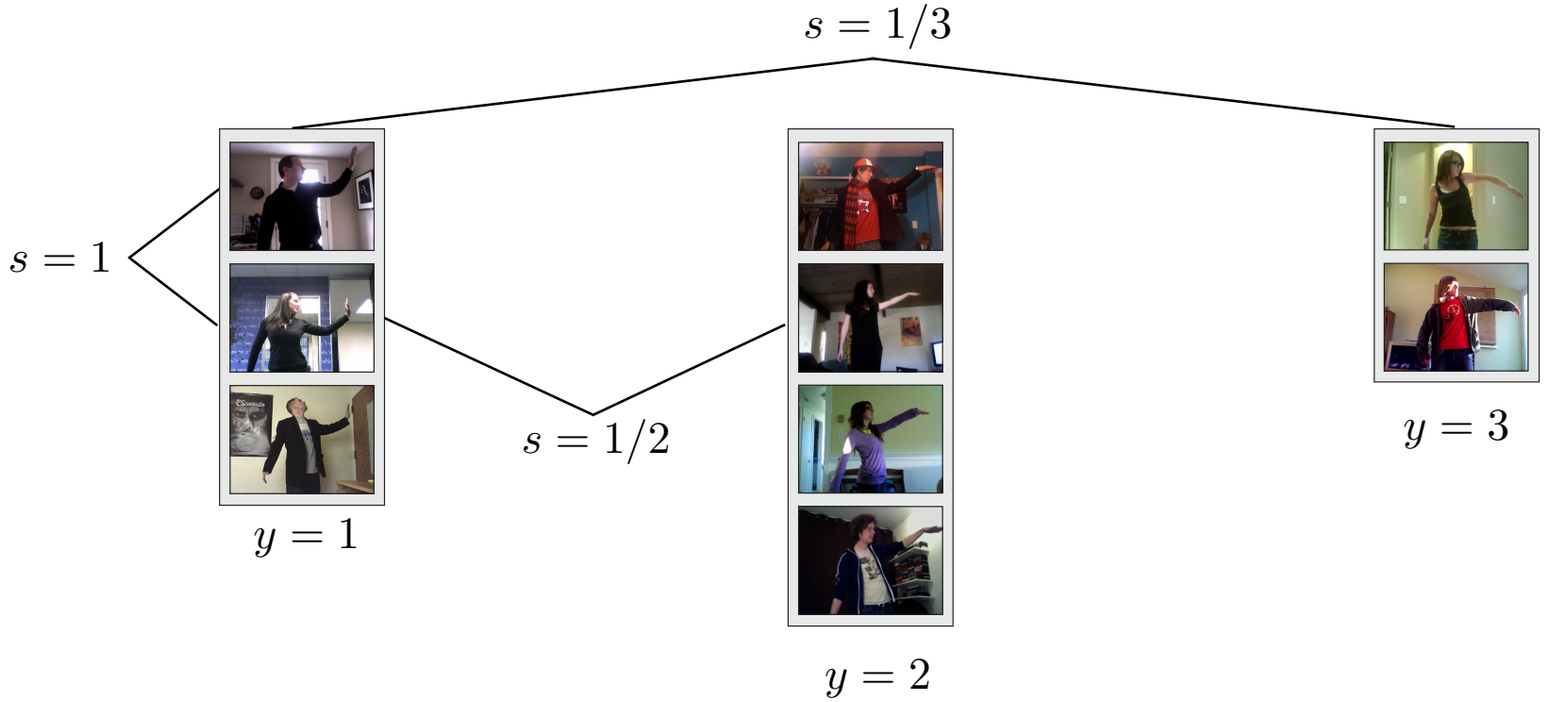
- In addition to the learned mapping, f, we require a mapping from discrete seed identity to a real-valued similarity score
- Simplest example: $s_{ij} = (1 + |y_i - y_j|)^{-1}$



Skip convnet intro

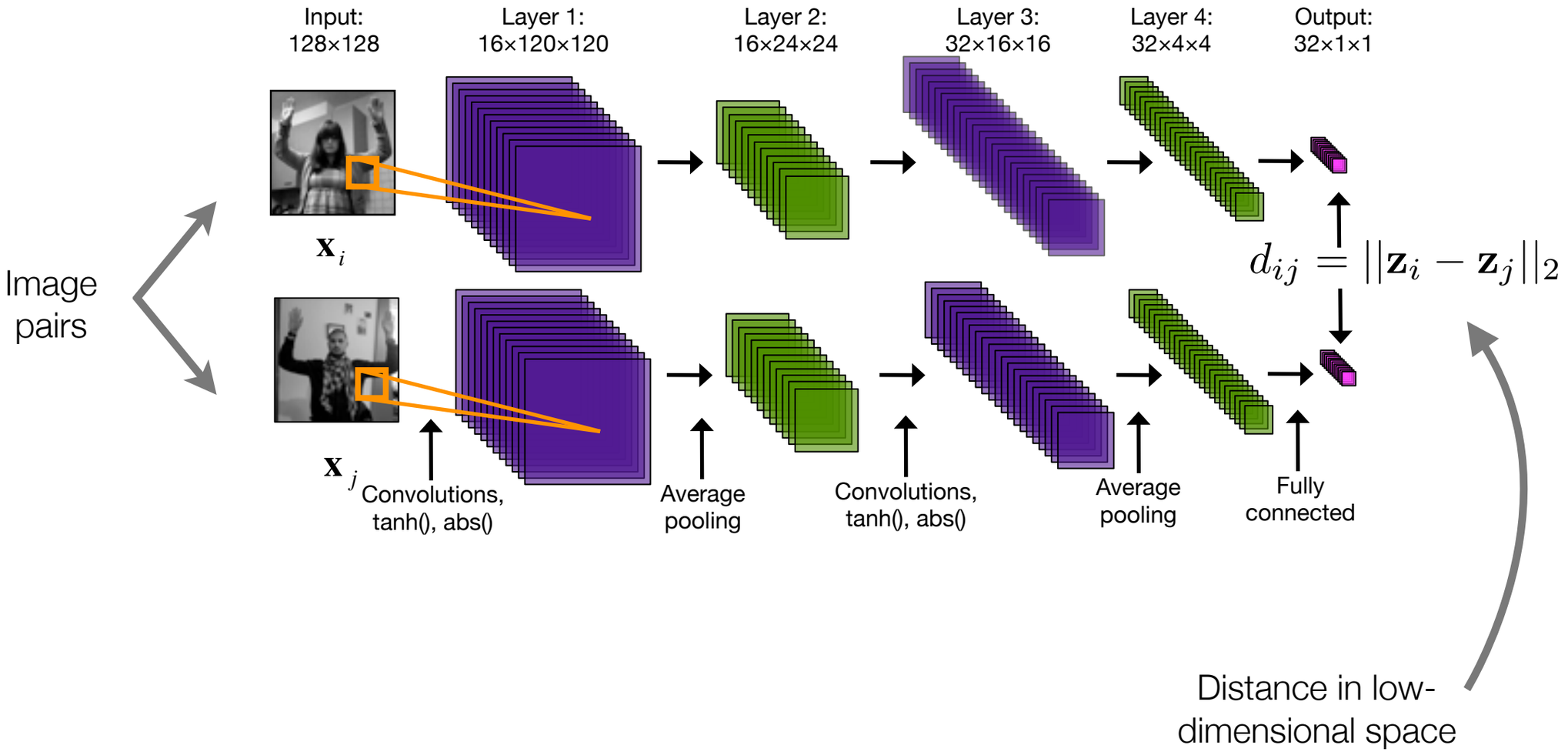
FROM DISCRETE LABELS TO SIMILARITY

- In addition to the learned mapping, f, we require a mapping from discrete seed identity to a real-valued similarity score
- Simplest example: $s_{ij} = (1 + |y_i - y_j|)^{-1}$



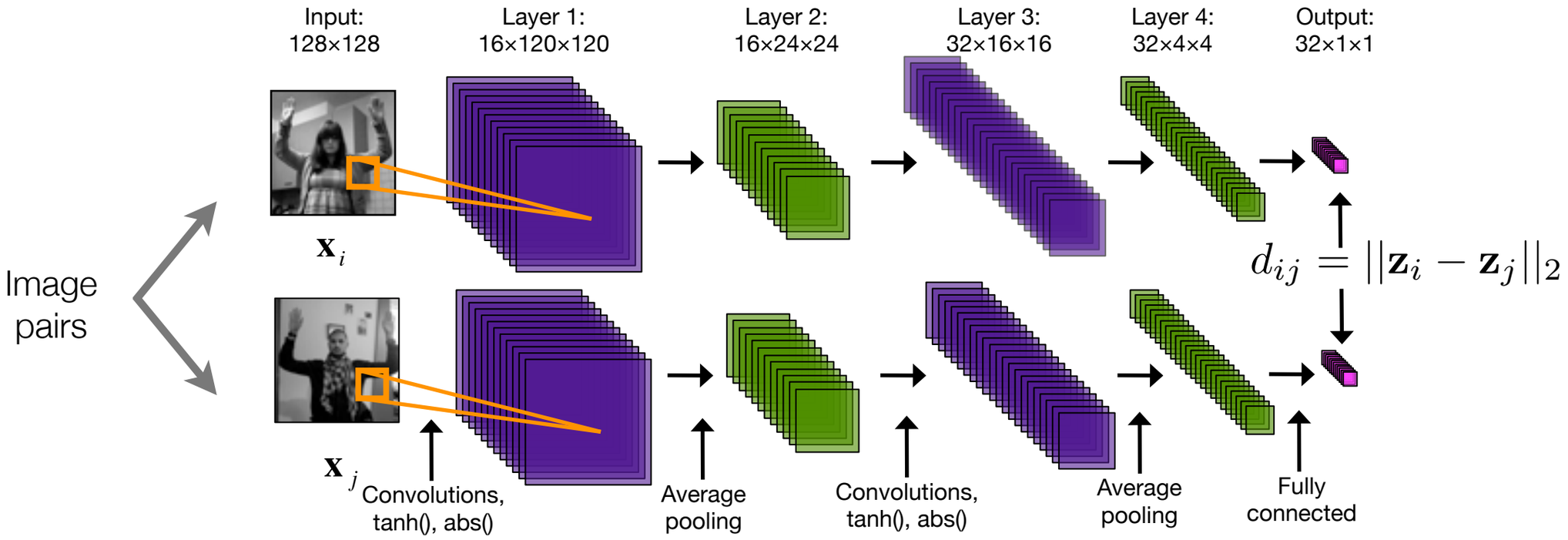
Skip convnet intro

EMBEDDING WITH A SIAMESE CONVOLUTIONAL NET



Skip toy experiments

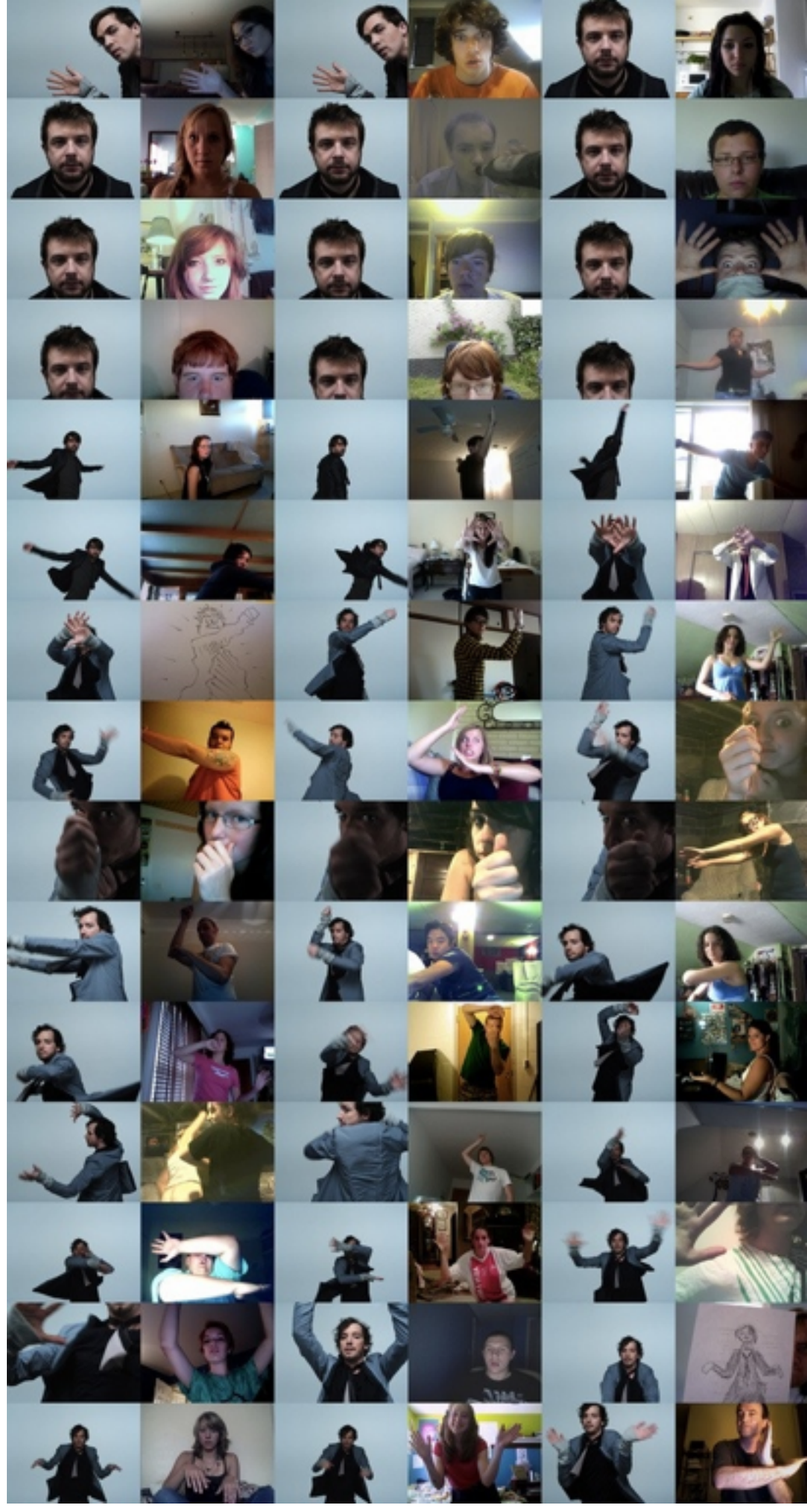
EMBEDDING WITH A SIAMESE CONVOLUTIONAL NET



Skip toy experiments

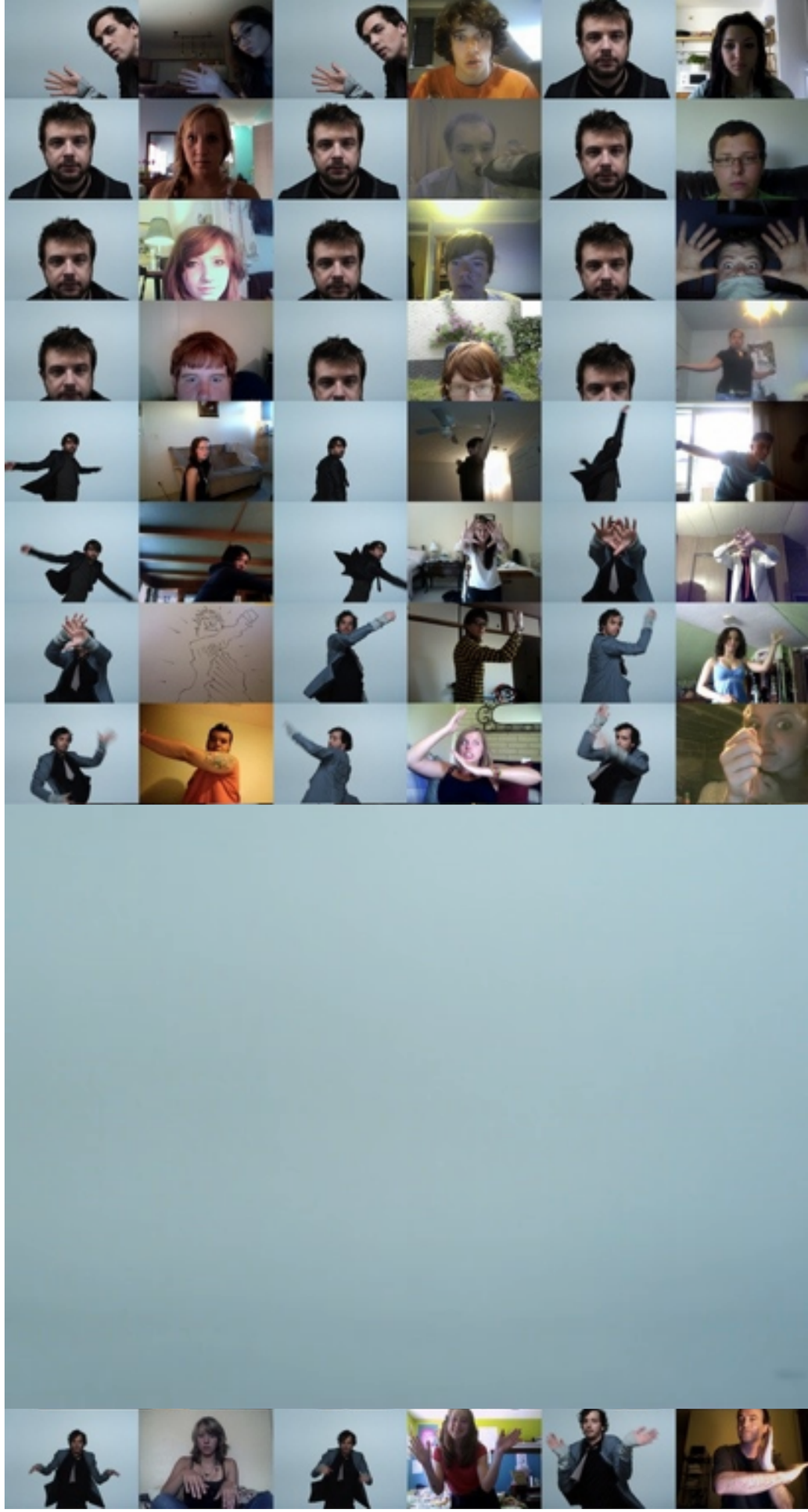
ONE FRAME OF FAME

- No need to collect data ourselves: we leverage an existing project in an unintended way
- *One Frame of Fame* is a music video by the Dutch band C-Mon & Kypski
- The band aims to replace selected frames with audience imitations
- To date, the band has >35k contributions
- Only manual intervention is in determining scene cuts

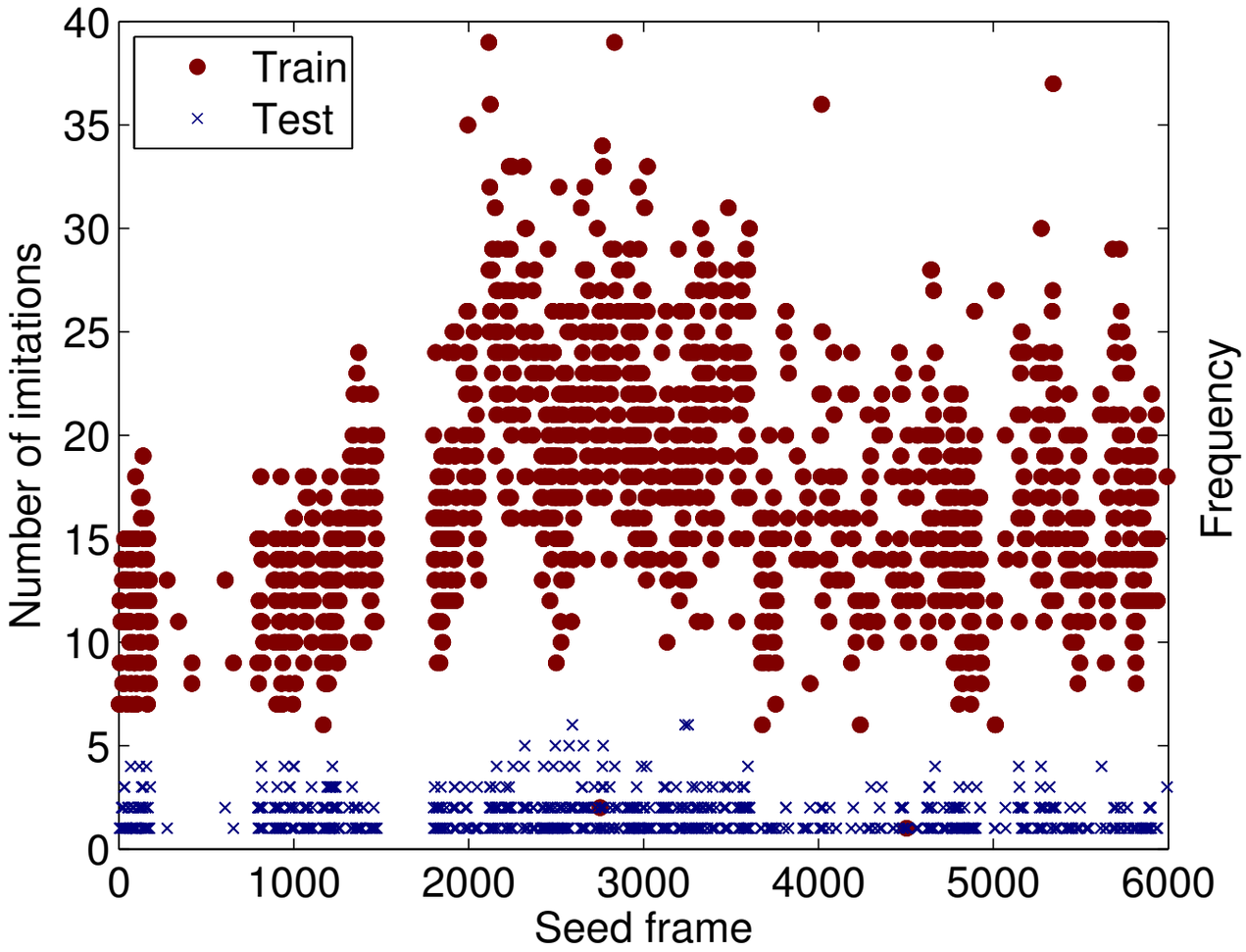


ONE FRAME OF FAME

- No need to collect data ourselves: we leverage an existing project in an unintended way
- *One Frame of Fame* is a music video by the Dutch band C-Mon & Kypski
- The band aims to replace selected frames with audience imitations
- To date, the band has >35k contributions
- Only manual intervention is in determining scene cuts



ONE FRAME OF FAME DATASET



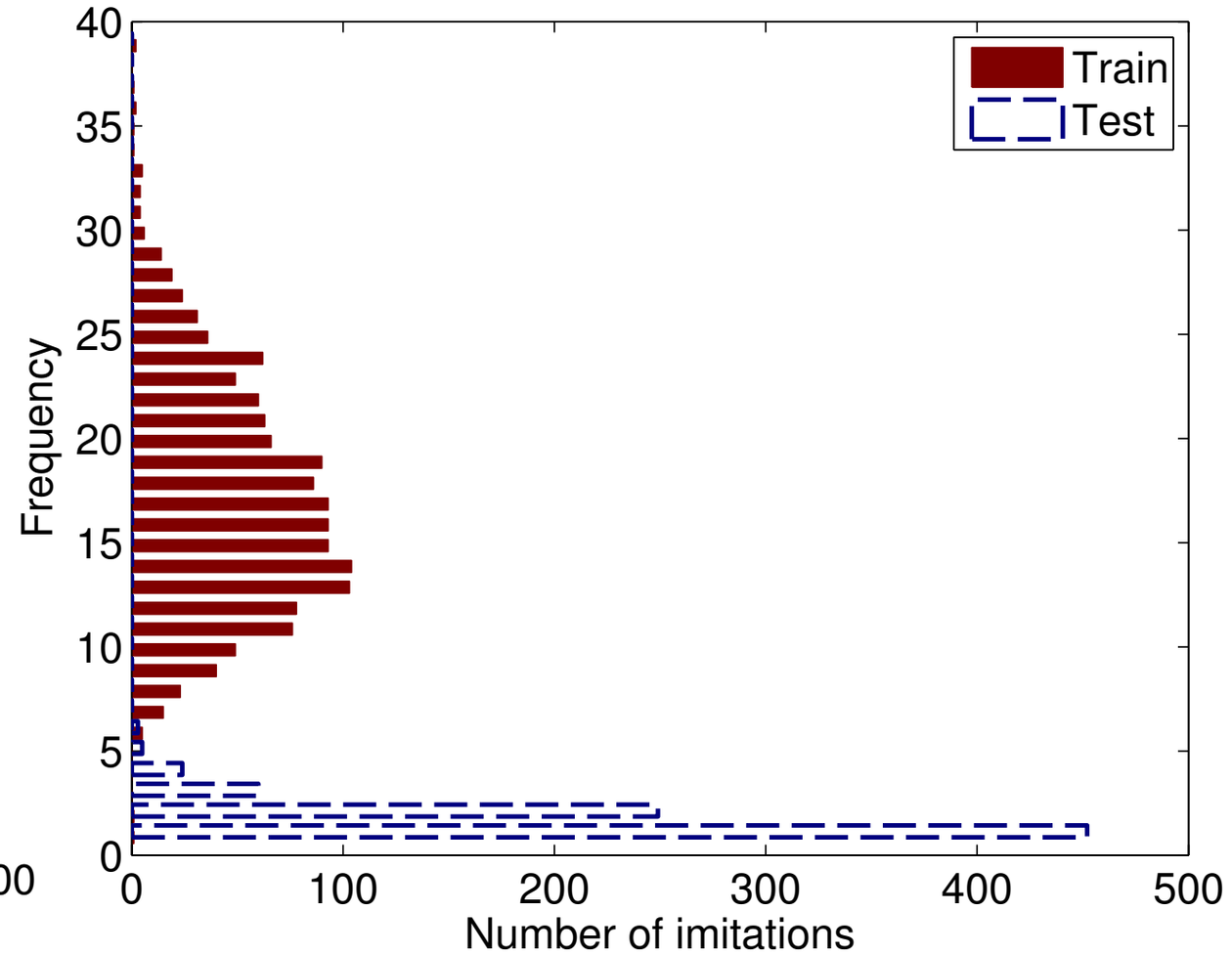


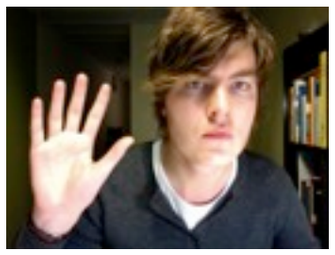
IMAGE RETRIEVAL

- The most common evaluation metric used by the IR community is Discounted Cumulative Gain (DCG)
- Typically used to measure search engine performance
- User submits query, presented with a ranked list of results

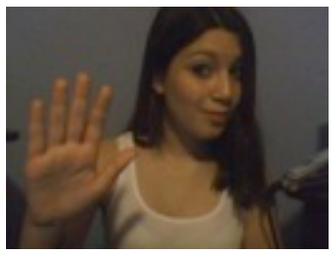
$$\mathrm{DCG@}K = \sum_{j=1}^{K} \frac{2^{g_j} - 1}{\log(j+1)}$$

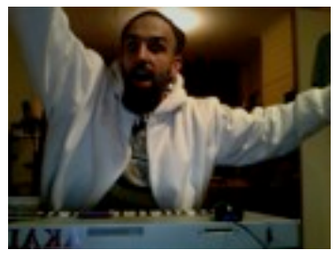
- We only consider the first *K* results
- In our experiments, we let $g_j = (1 + |y_i - y_j|)^{-1}$

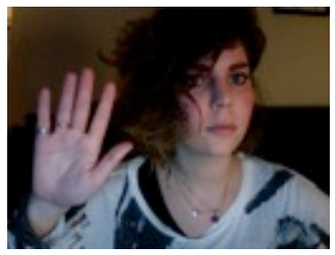
Query (test set)



Ranked list (training set)

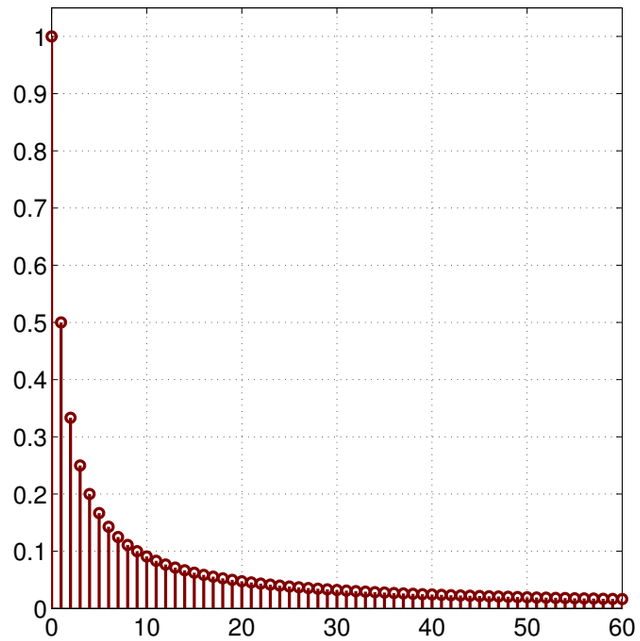




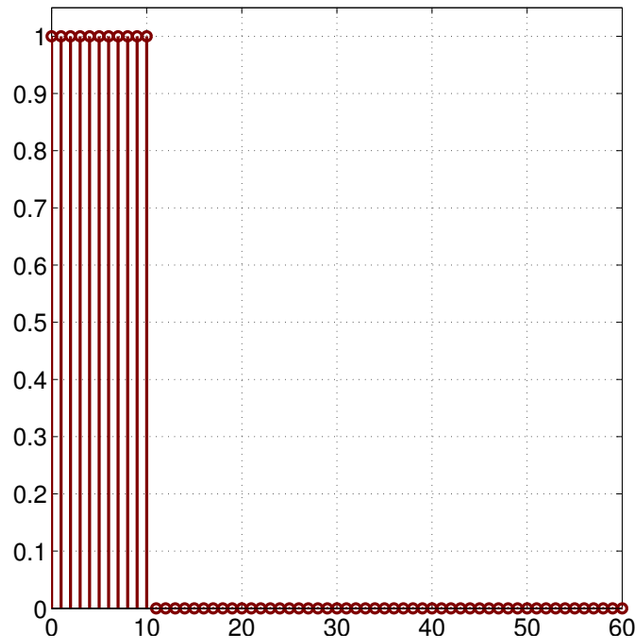


METHODS FOR SIMILARITY SCORE

- Simple: $$s_{ij}^m = (1 + |y_i - y_j|)^{-1}$$

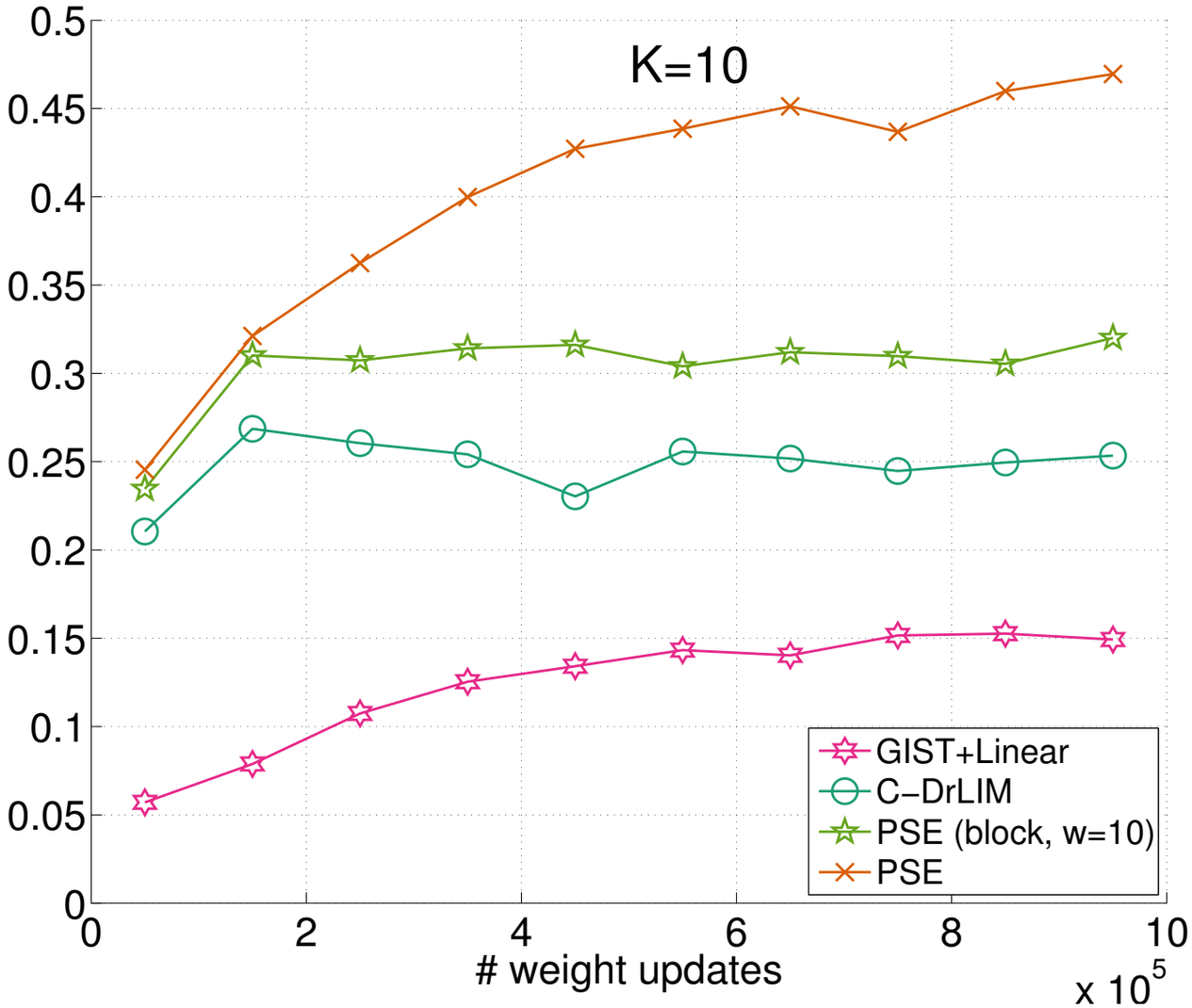


- Block: $$s_{ij}^m = 1 \quad \text{if} \quad |y_i - y_j| <= w$$



RETRIEVAL PERFORMANCE: QUANTITATIVE

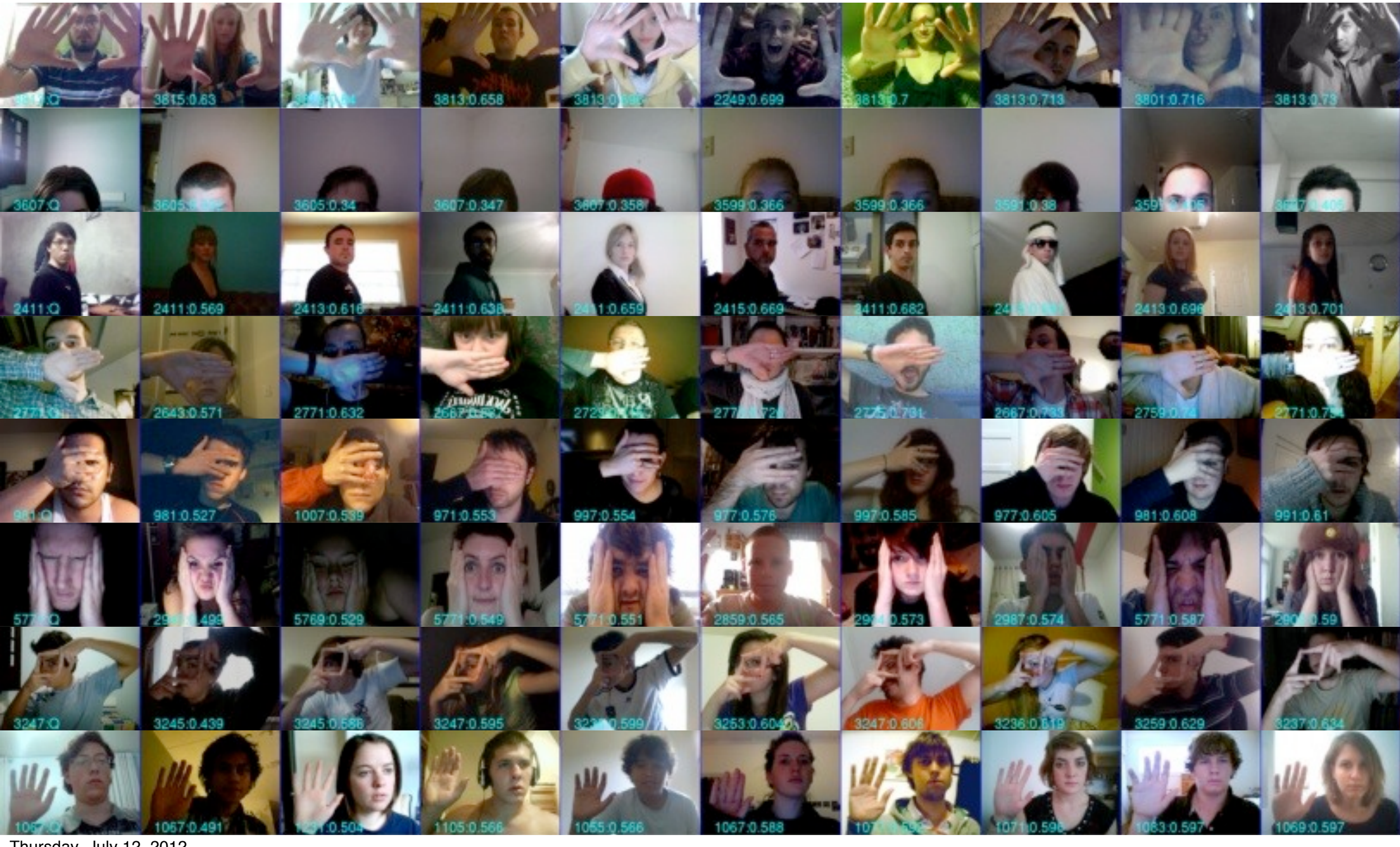
- Both pixel-based matching, and PCA perform horribly: Pixels: 0.021, PCA-32: 0.026
- Standard DrLIM does not consider graded similarity
- Performance of using a fixed-size window of constant affinity falls between DrLIM and soft methods
- Similar performance observed for K=1, K=5, K=20 NN



RETRIEVAL PERFORMANCE: QUALITATIVE

Query
(test set)

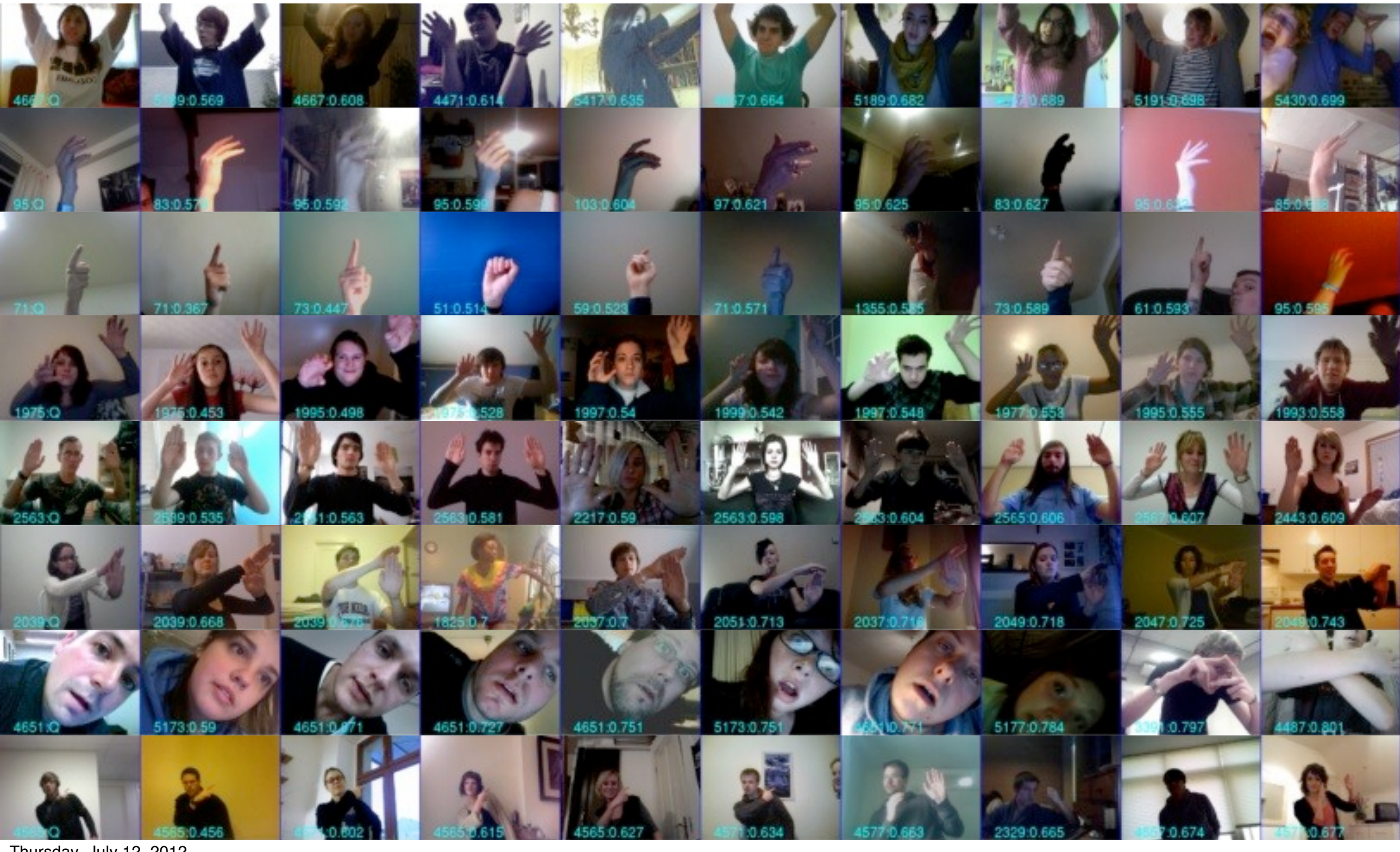
Nearest neighbours



MORE QUALITATIVE RESULTS

Query (test set)

Nearest neighbours



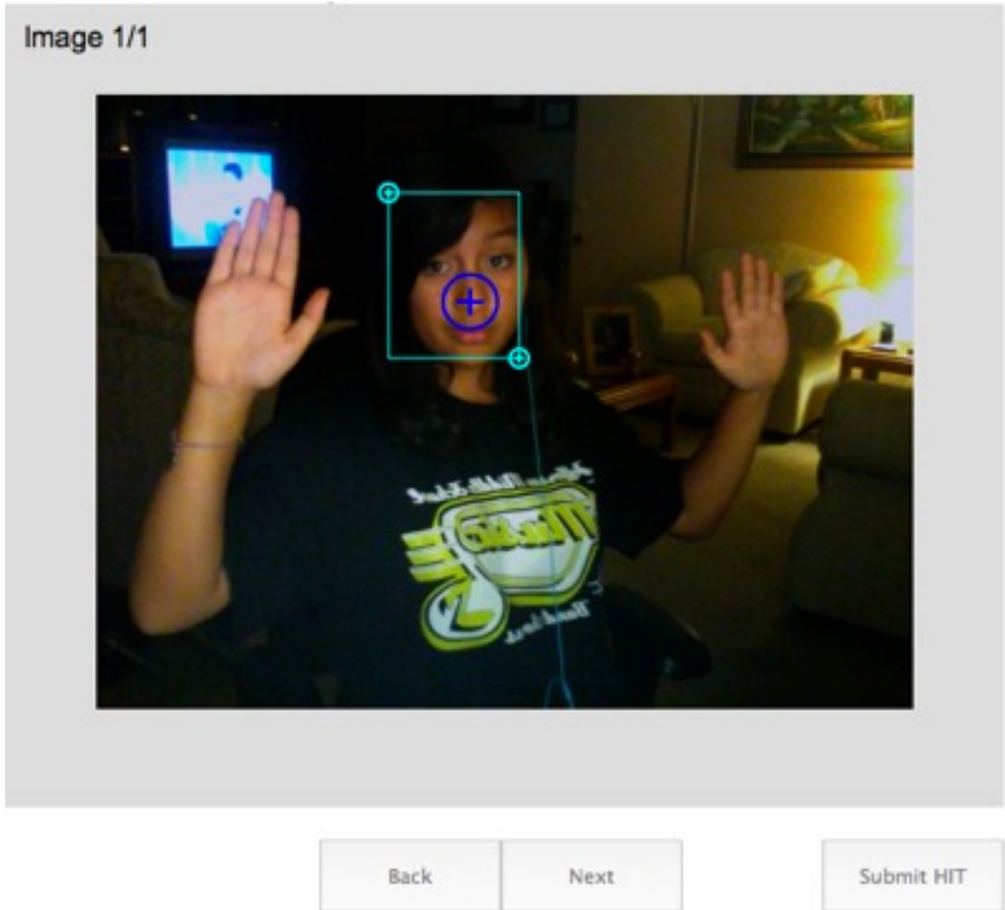
EVEN MORE QUALITATIVE RESULTS

Query
(test set)

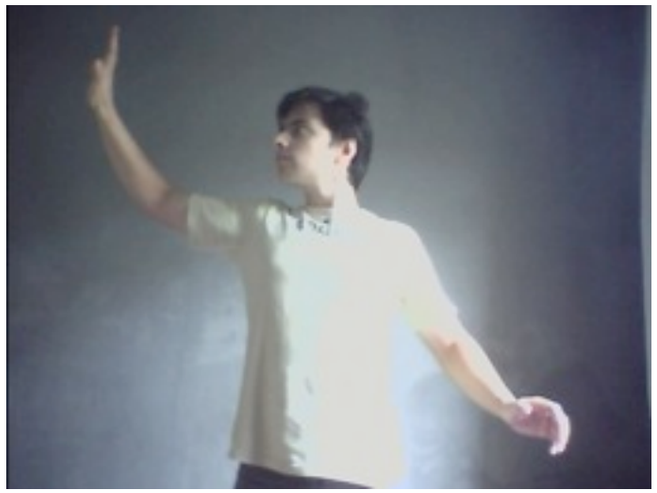
Nearest neighbours

FACE DETECTION USING OUR LEARNED EMBEDDING

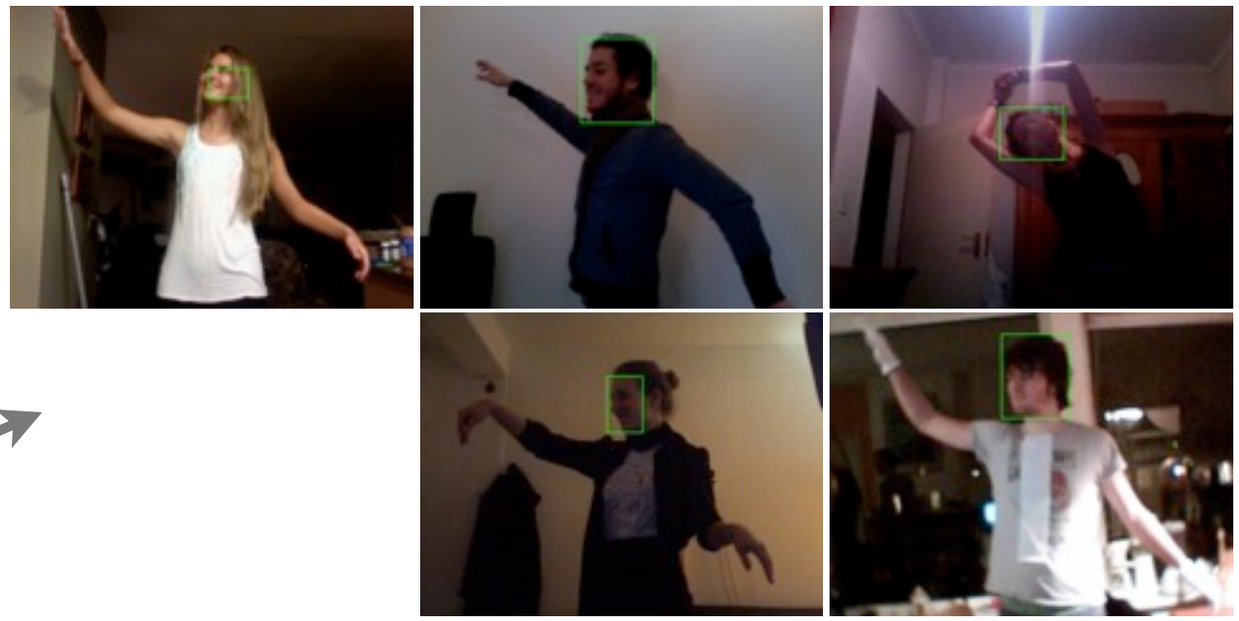
- Users on Amazon Mechanical Turk were asked to provide facial bounding boxes for the training set
- We reduced our training (and test) set to the subset of “valid” annotations - for example, some images do not contain faces and therefore were not assigned bounding boxes



NEAREST NEIGHBOR FACE DETECTION

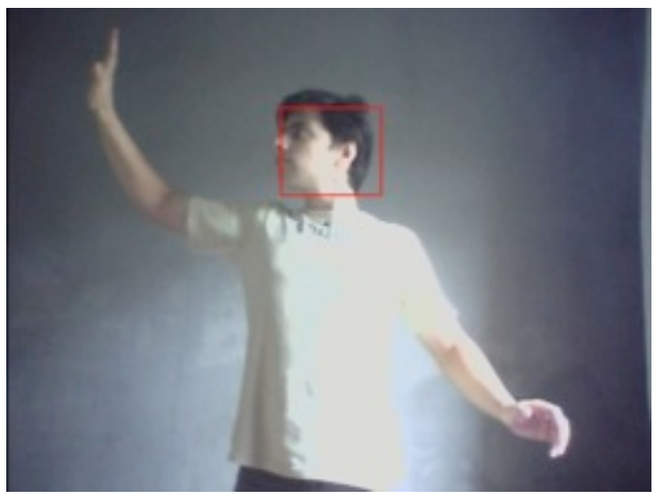


Query image (from test set)



Find nearest neighbors via learned pose-sensitive embedding

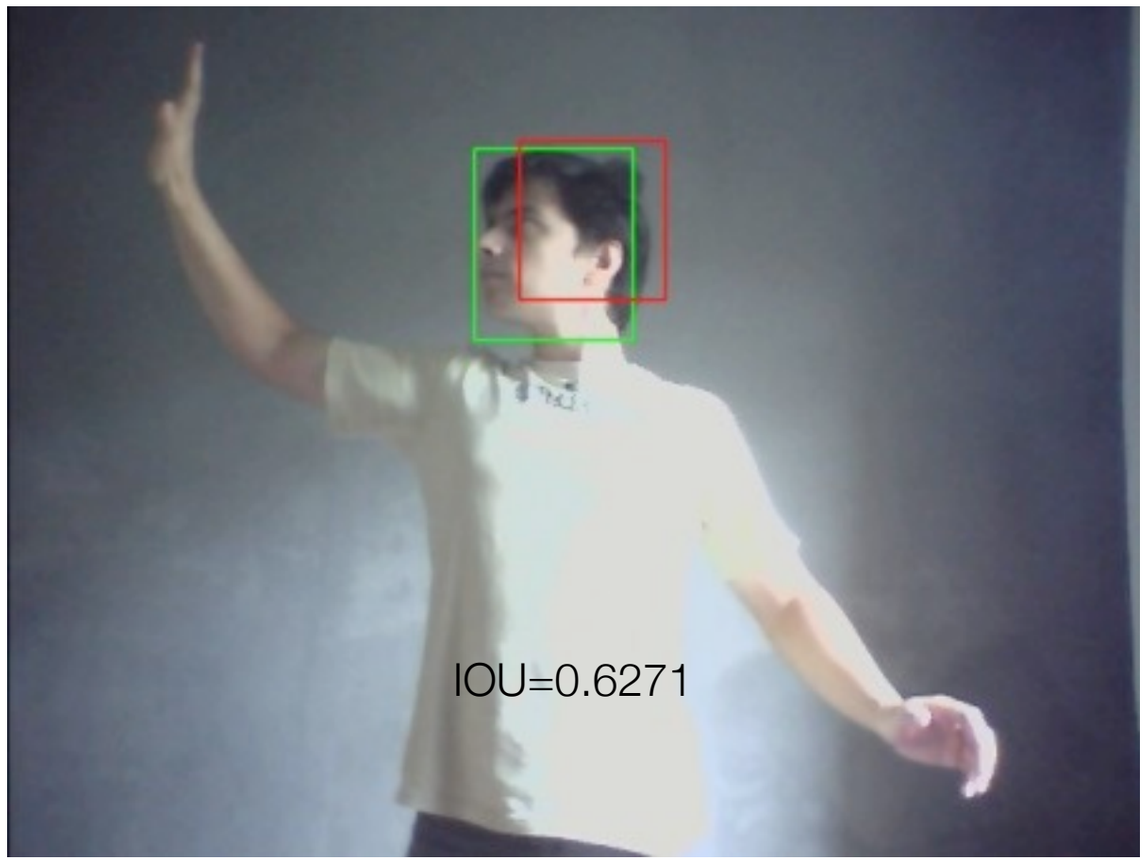
Apply median bounding box of neighbors

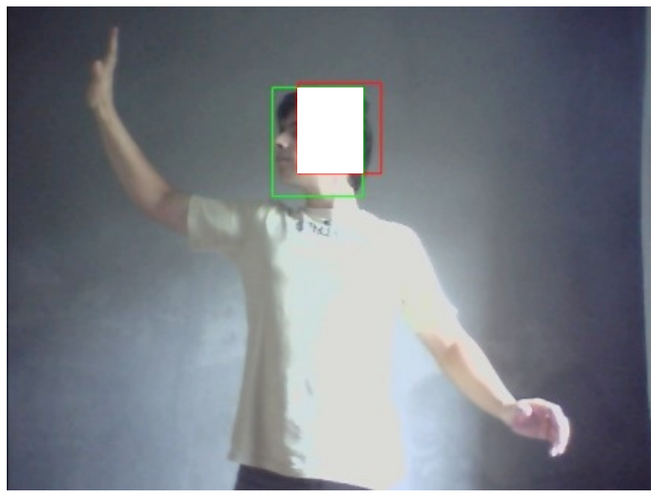


Proposed bounding box

SCORING BOUNDING BOXES

- Use Intersection over Union score (PASCAL VOC): IOU > 0.5?
- Red - our guess; Green - ground truth





Intersection

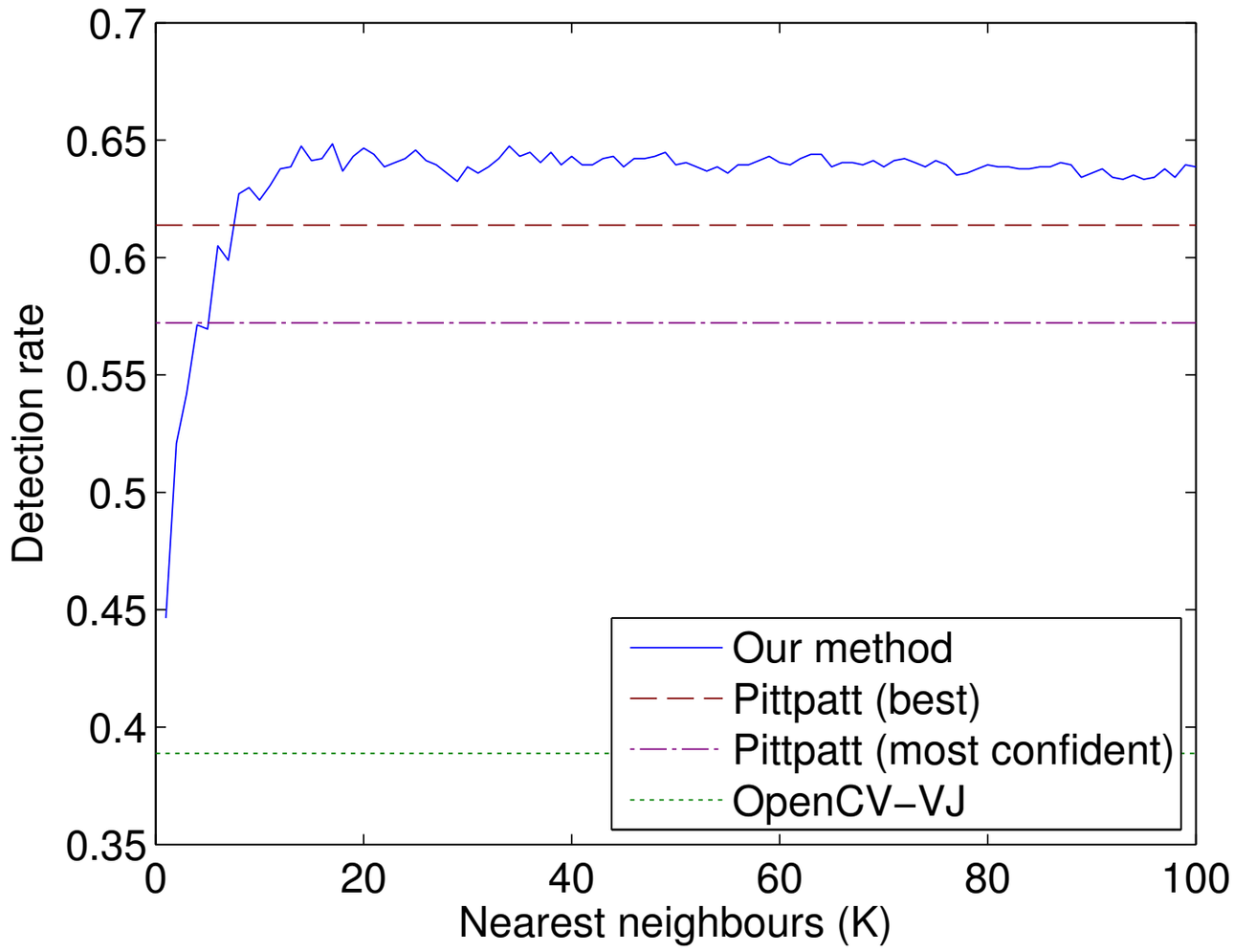
Union

OUTPERFORMING PITTPATT

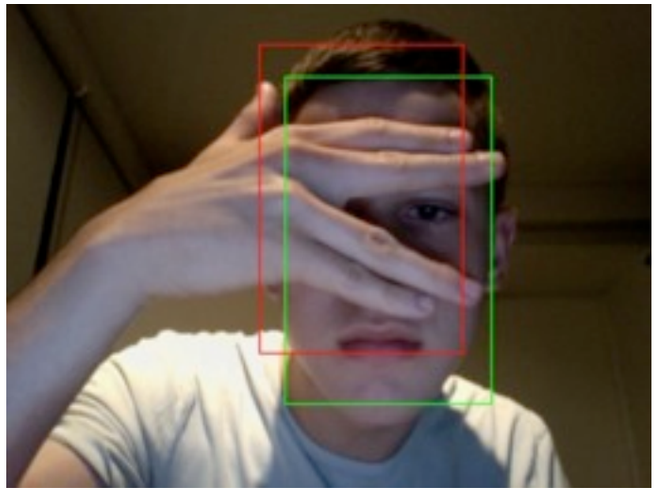
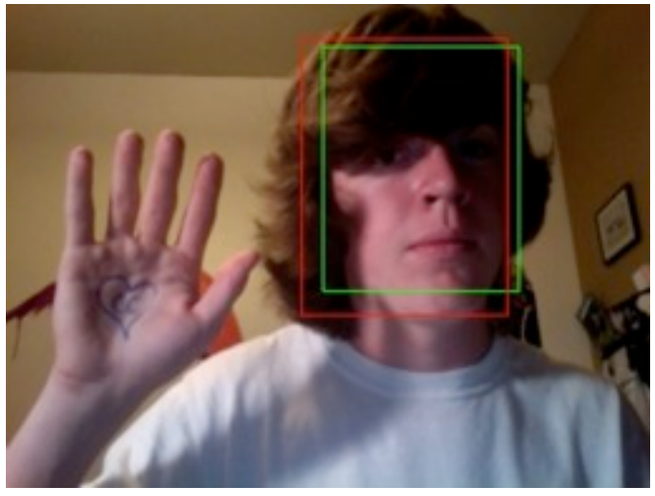
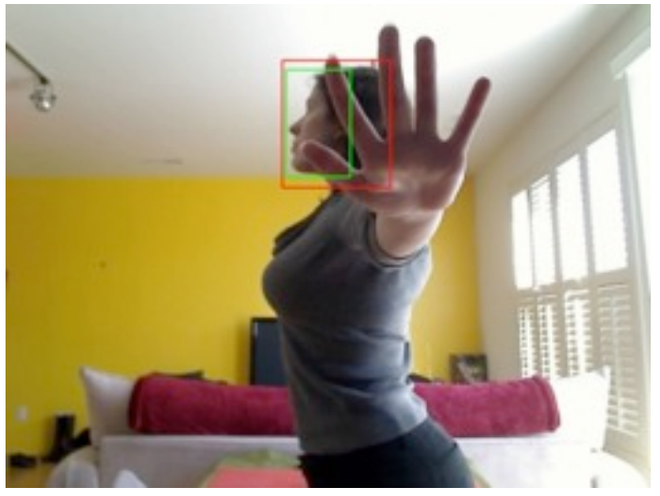
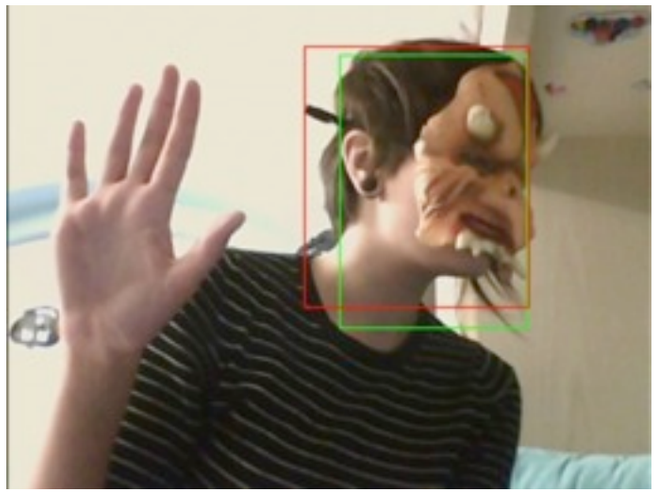
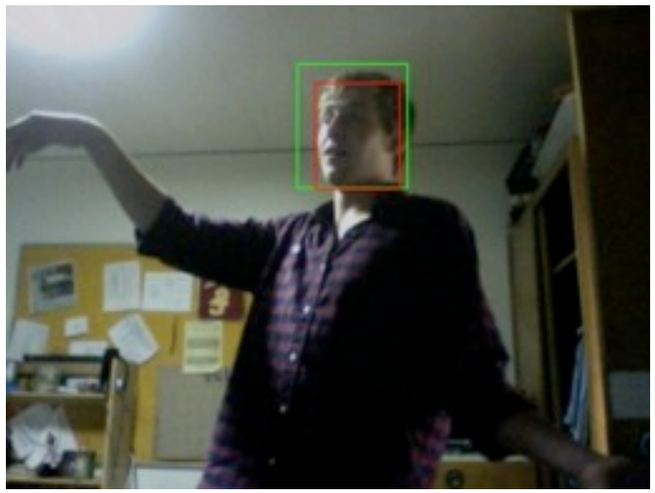
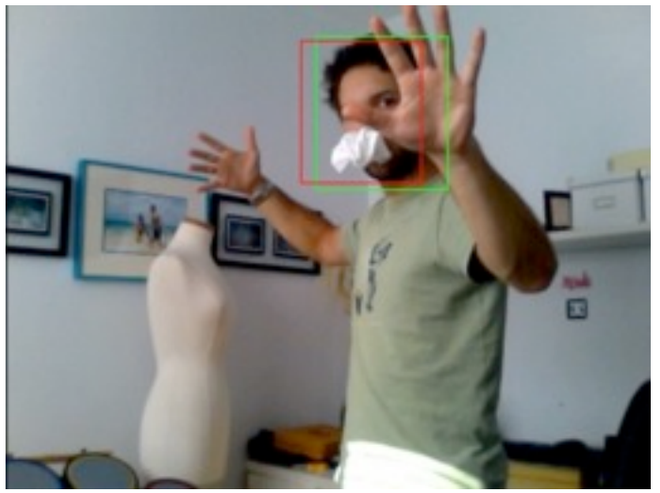
- PittPatt is a commercial face detector

- OpenCV - VJ is a commonly used implementation of boosting (Viola-Jones) - known to not work that well

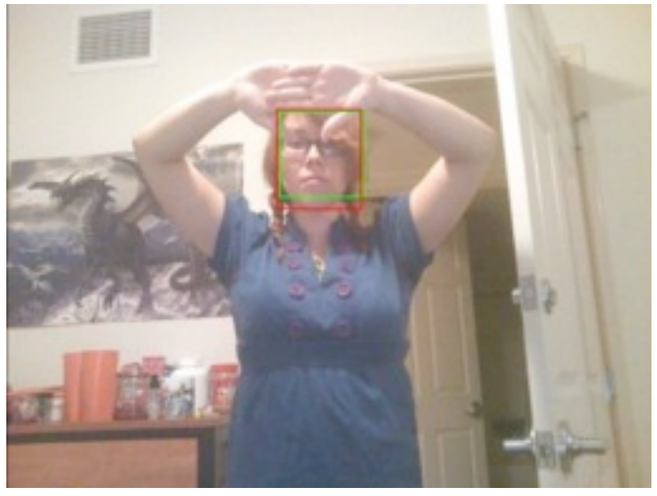
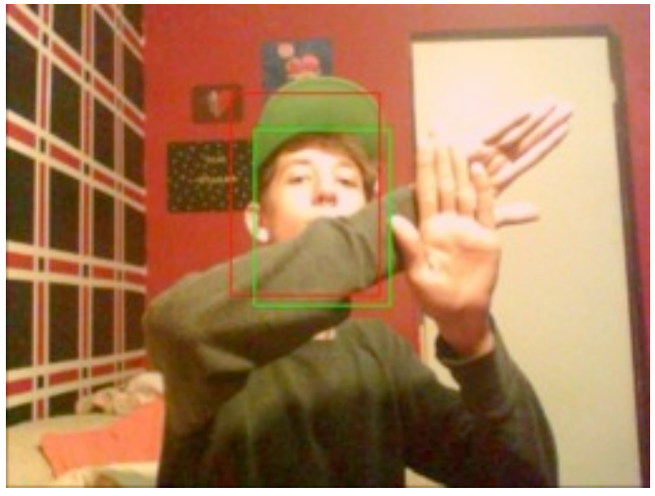
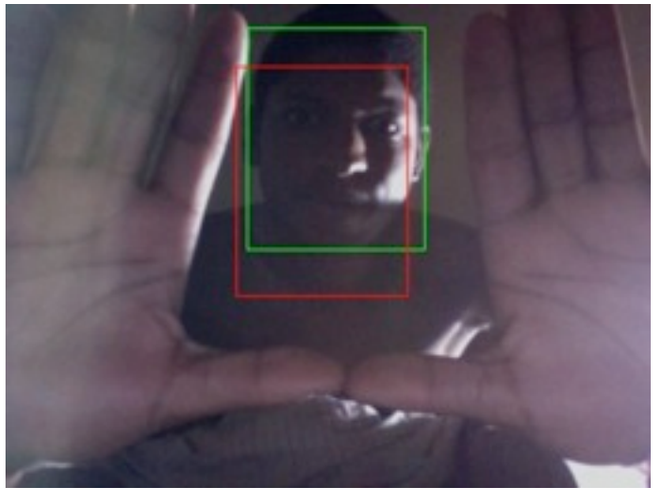
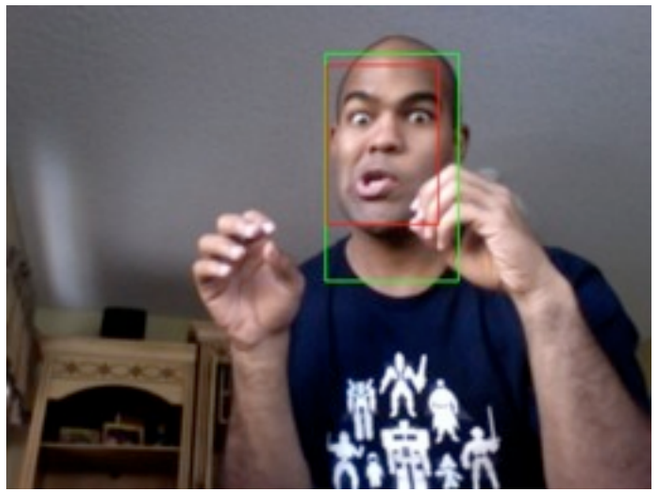
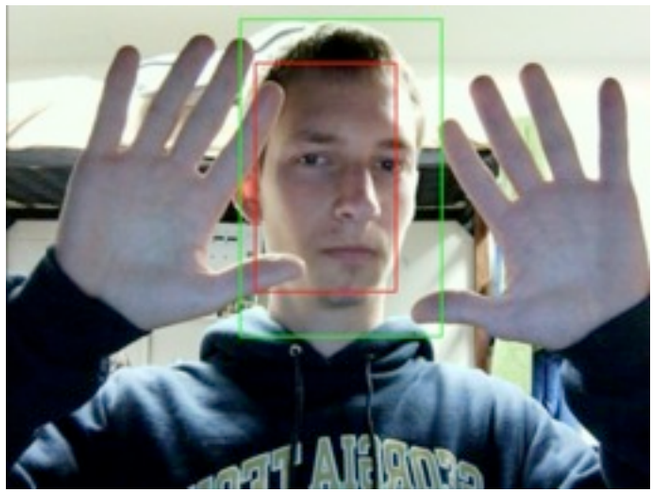
- Detection is IOU > 0.5



PITTPATT FAILURES (PSE SUCCEEDS)

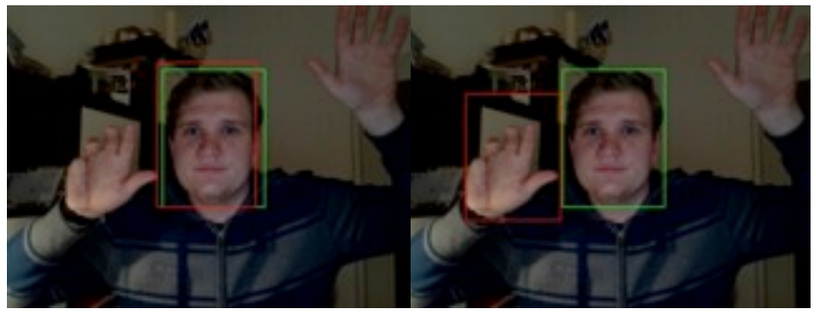







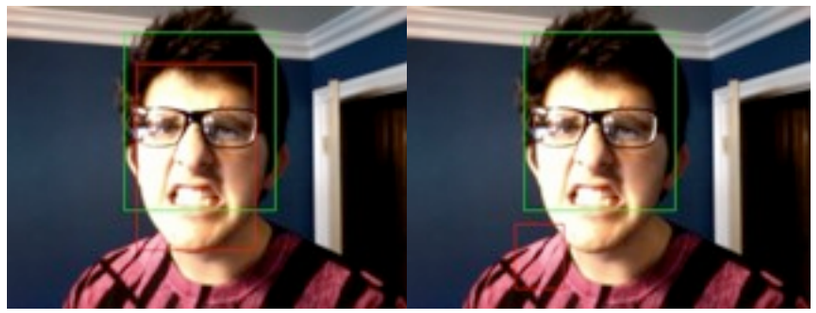
PITTPATT FAILURES (2)

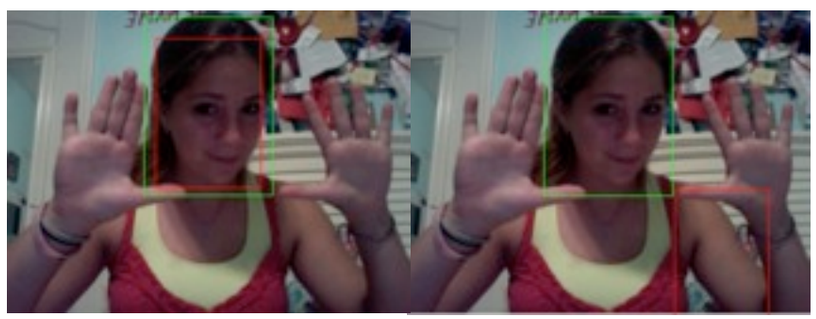







PITTPATT PRODUCES INCORRECT DETECTION

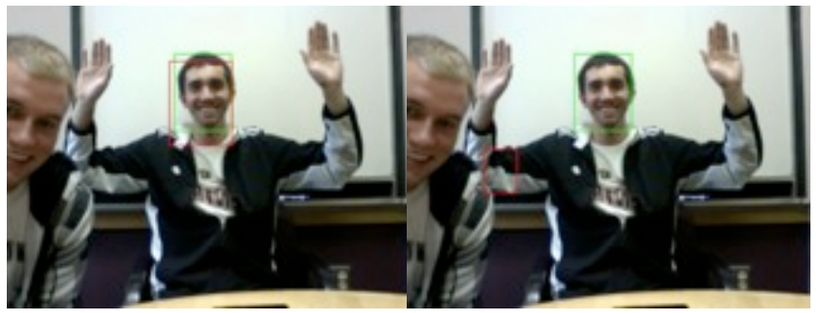
PSE PittPatt

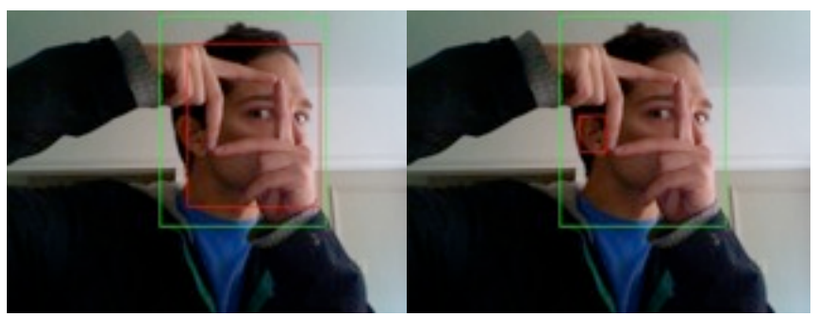






PSE PittPatt

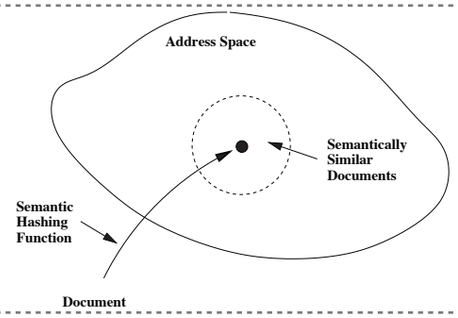




SUMMARY

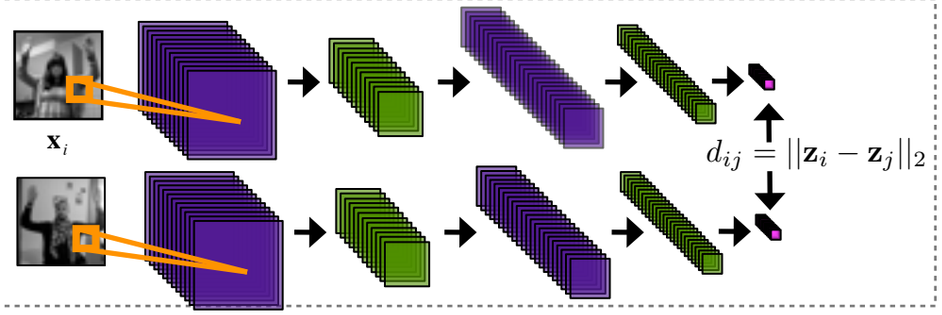
Unsupervised

Learn similarity structure completely from unlabeled data.
Difficult to ensure that similar examples map to similar codes.



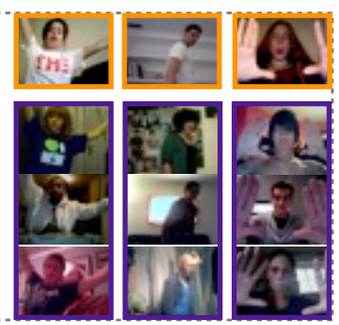
Supervised

Use labels or neighbourhood graph to inform map.
Often, this information is not available!



Weakly supervised

Use of temporal coherence to guide learning.



The University of Guelph is not in Belgium!