FACE RECOGNITION: ROBUST TRANSFER LEARNING USING THE MULTIVERSE LOSS

Prof. Lior Wolf
The School of Computer Science
Tel Aviv University

ACK: I will present work done in collaboration with

TAU students:

Etai Littwin, Dedi Gadot, Tomer Galanti

FAIR researchers:

Yaniv Taigman, Ming Yang, Marc'Aurelio Ronzato

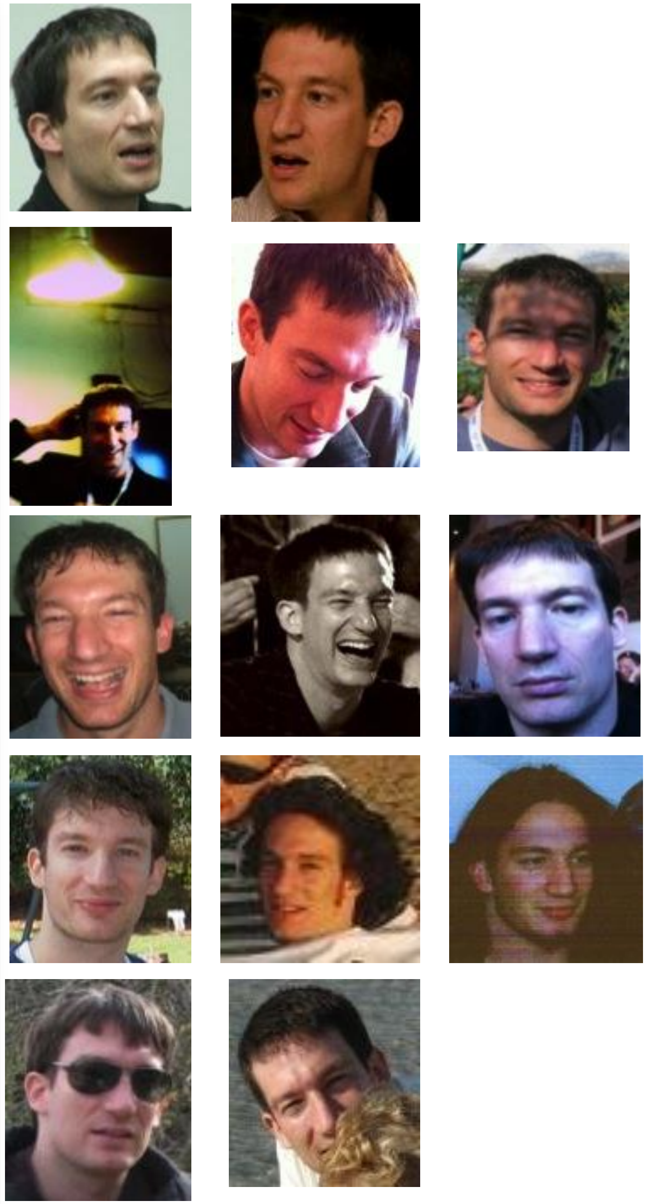
Why faces?

1. The most frequent entity in the media by far: e.g. ~1.2 faces / Photo on avg
2. Understanding identification
3. One class, billions of instances

Challenges in Unconstrained Face Recognition

1. Pose
2. Illumination
3. Expression
4. Aging
5. Occlusion

Probes for example



Gallery

Unconstrained Face Recognition Era: The Labeled Faces in the Wild (LFW)

13,233 photos of 5,749 celebrities

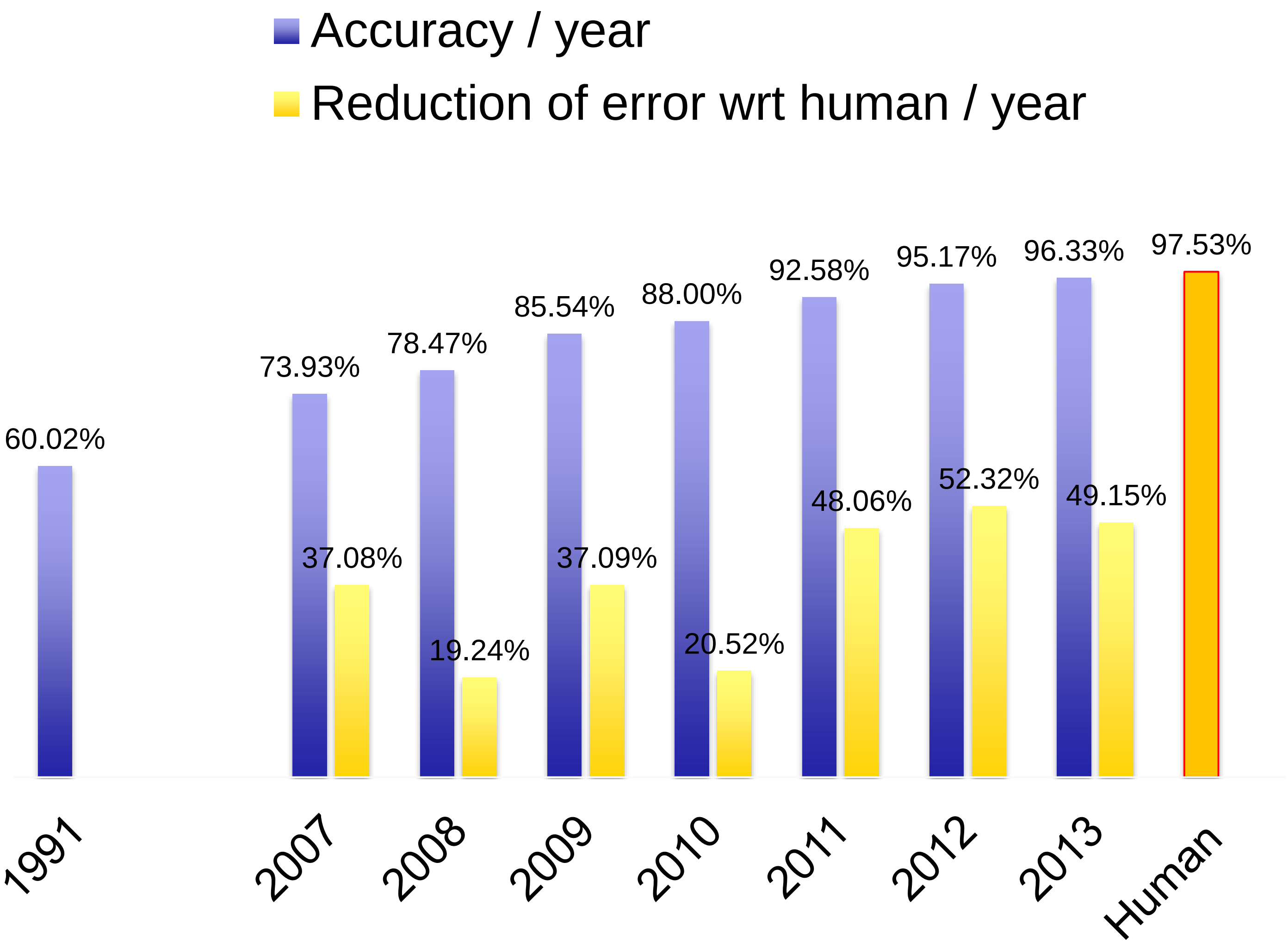
Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Huang, Jain, Learned-Miller, ECCVW, 2008

Face verification

=

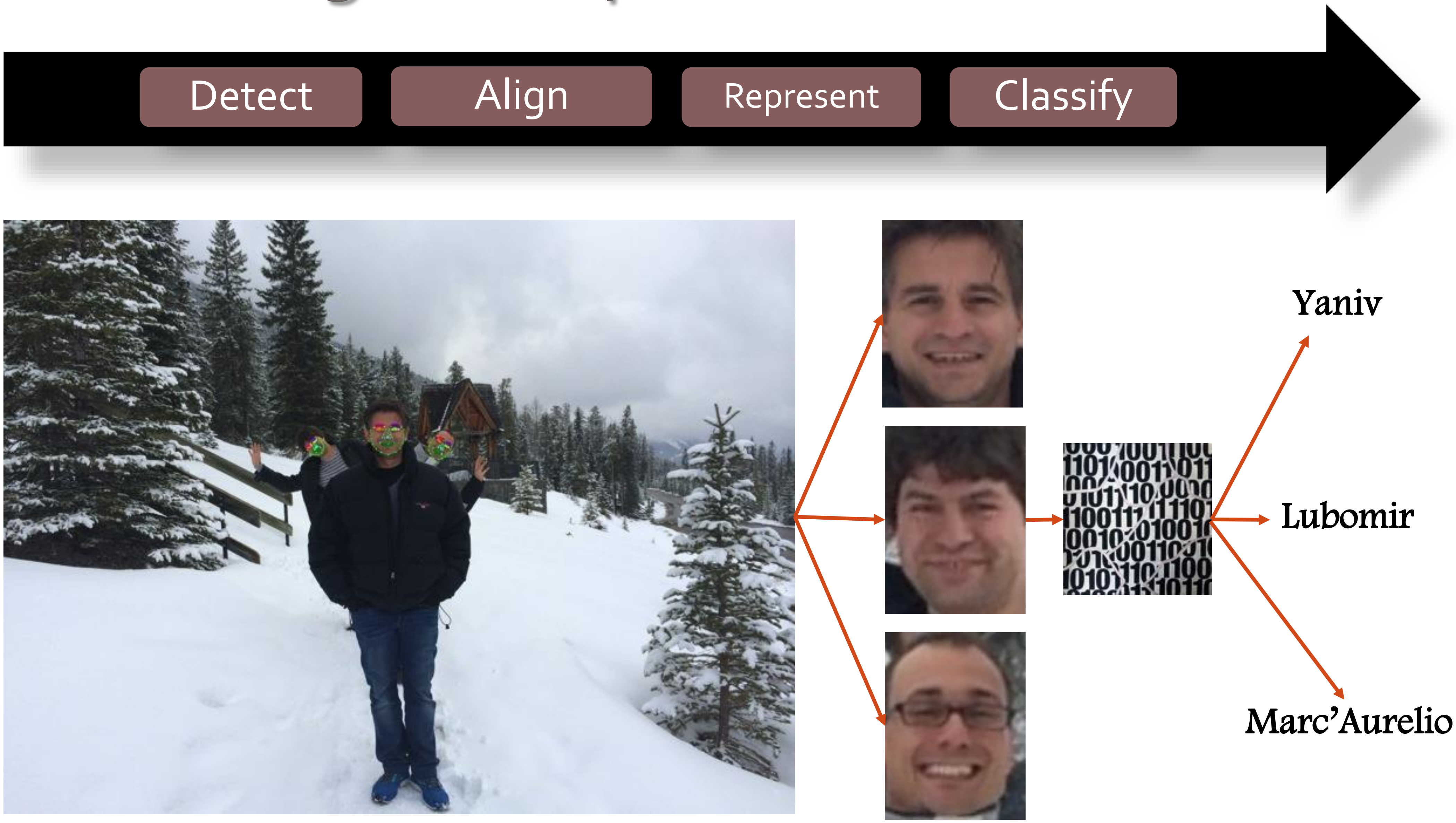
!=

Progress over the past 7 years

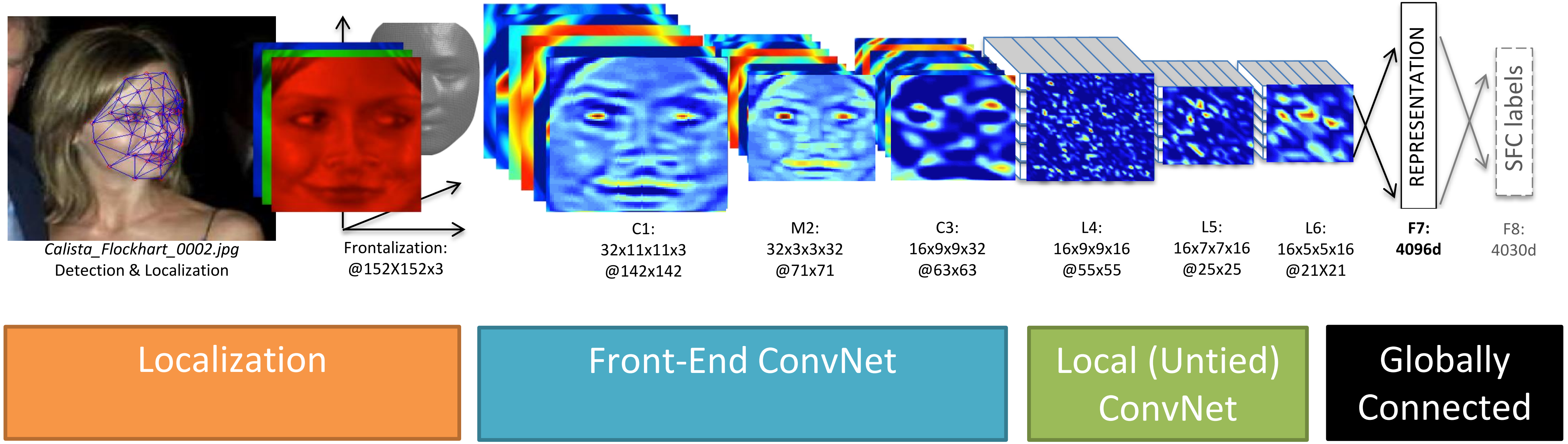


Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments (results page), Gary B. Huang, Manu Ramesh, Tamara Berg and Erik Learned-Miller.

Face Recognition Pipeline

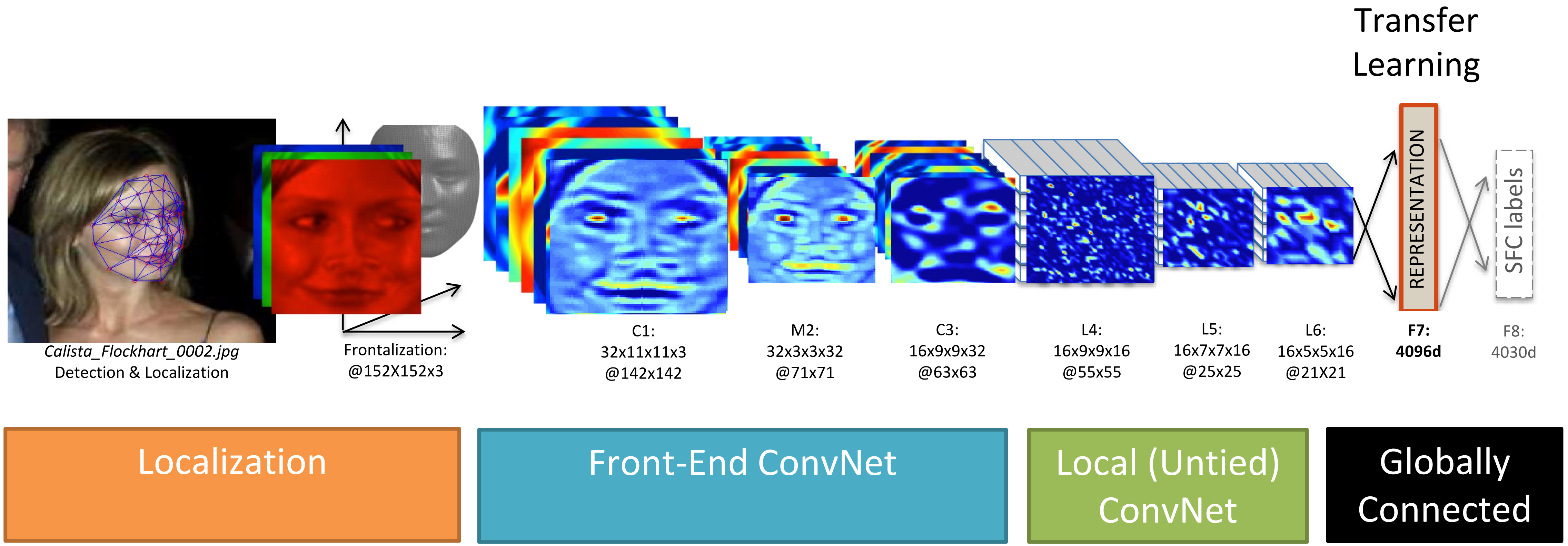


Deep Neural Networks on aligned inputs



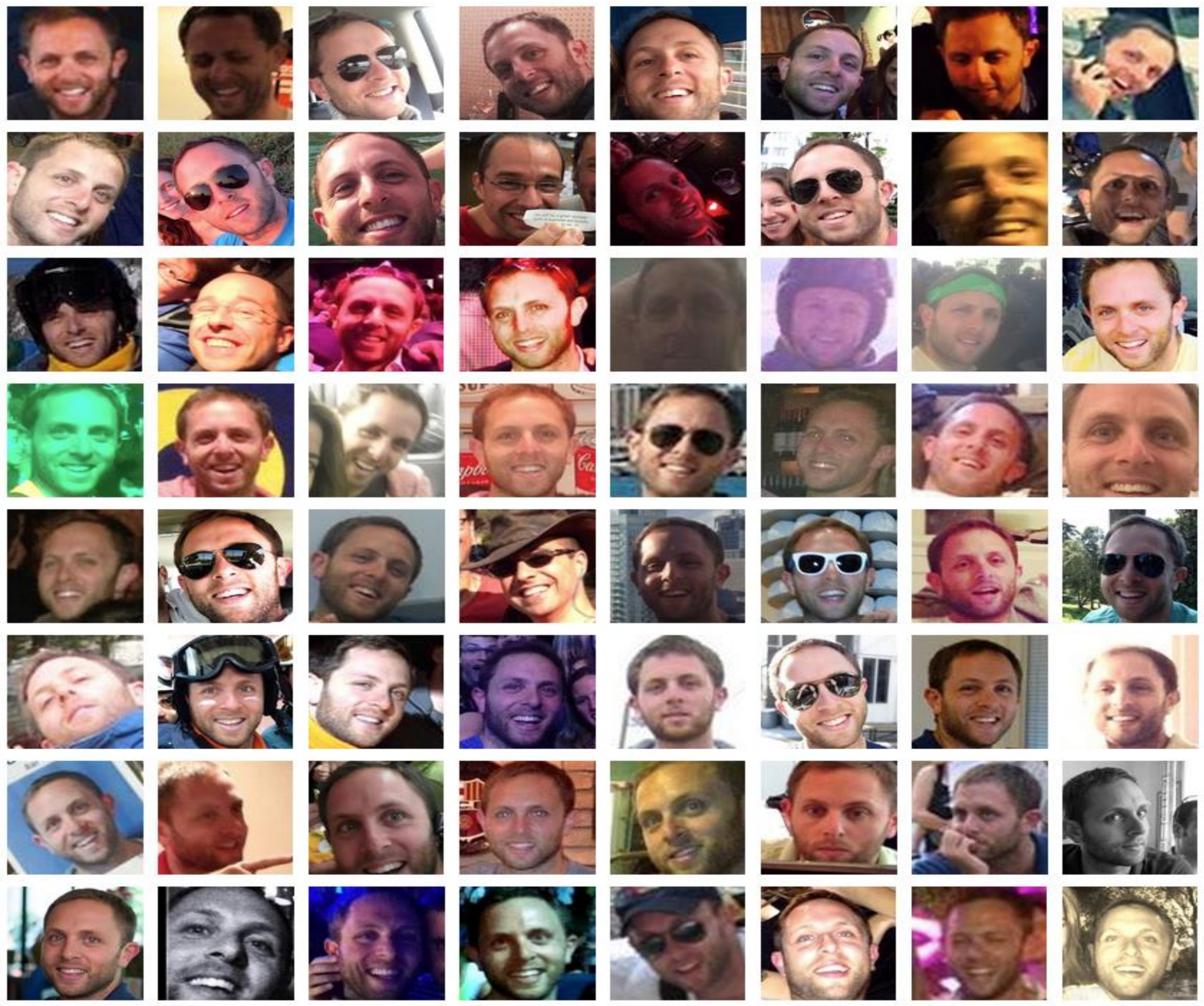
Taigman, Yang, Ranzato, Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. CVPR, 2014.

Deep Neural Networks on aligned inputs



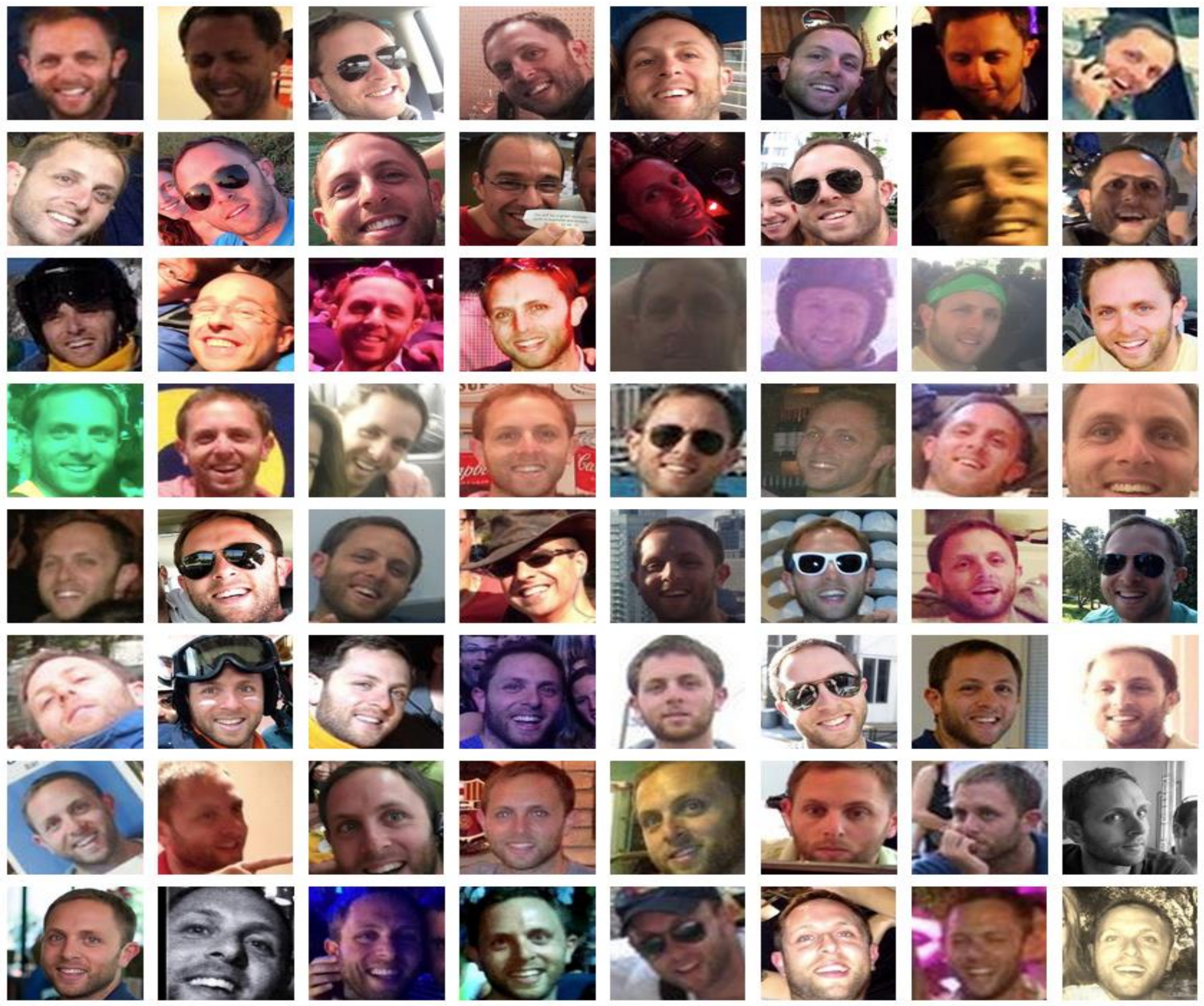
Taigman, Yang, Ranzato, Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. CVPR, 2014.

SFC Training Dataset



4.4 million photos blindly sampled, belonging to more than 4,000 identities

SFC Training Dataset



4.4 million photos blindly sampled, belonging to more than 4,000 identities

Many images per person, but not too many identities

Q1: What is better for learning a *generic* face representation: more identities or more samples per identity?

Galanti, Wolf, Hazan. A Theoretical Framework for Deep Transfer Learning. IMAIAI, 2016

The tradeoffs that govern transfer learning

I. For a given budget of samples. How to split between classes and samples per class.
II. Having too many samples and not enough classes leads to overfitting. But not the other way around.
III. The size of the representation and the number of training samples.
IV. Saturation.

4.4 million photos blindly sampled, belonging to more than 4,000 identities

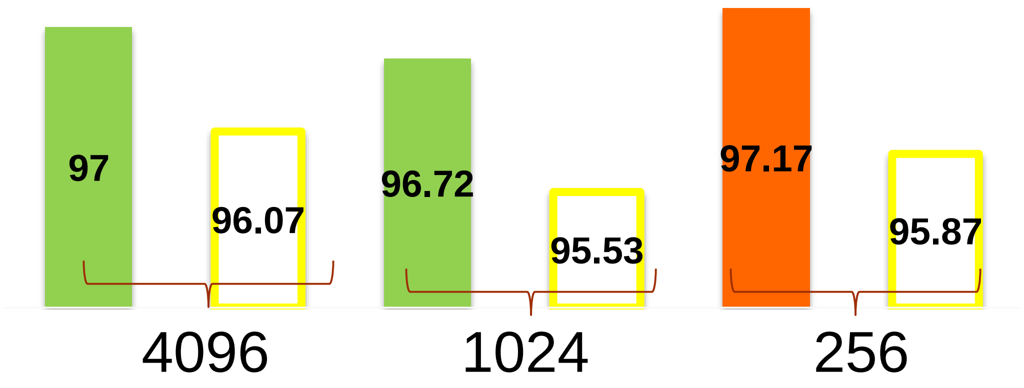
Many images per person, but not too many identities

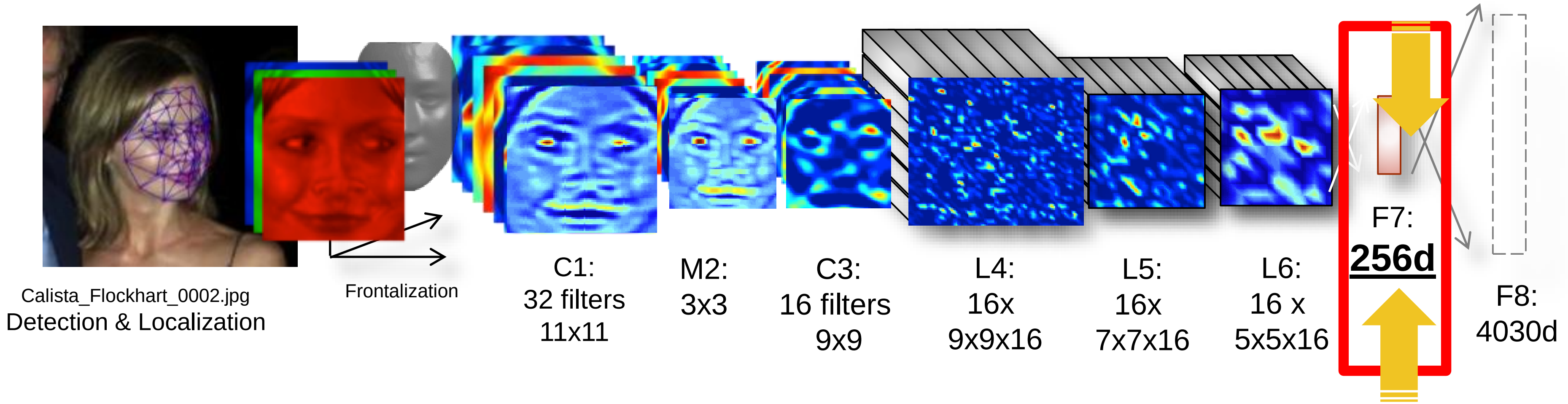
Q1: What is better for learning a *generic* face representation: more identities or more samples per identity?

Galanti, Wolf, Hazan. A Theoretical Framework for Deep Transfer Learning. IMAIAI, 2016

What size representation is ideal?

The network overfits less on the SOURCE training set, and performs better on the TARGET when reducing the representation layer (F7) from 4K dims to 256 dims.

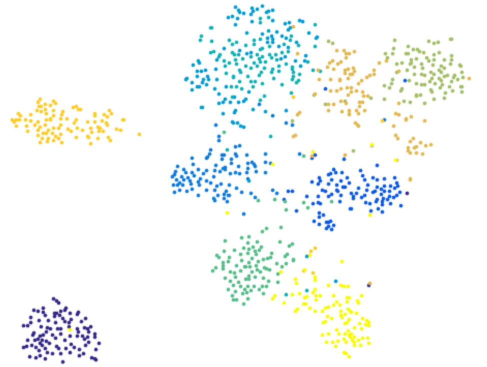




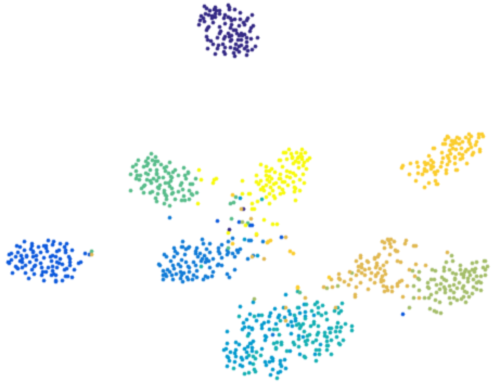
Can the data suggest optimal dim?

- The dimensionality of the representations is mostly wasted
  - Full rank representation
  - Decisions made based on few dims

Regular: 90D, little separation



Multiverse: 35D, good separation



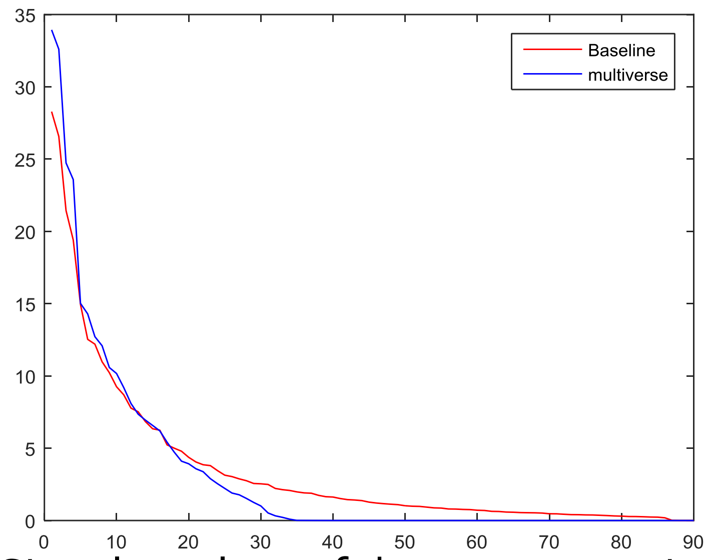
Littwin, Wolf. The Multiverse Loss for Robust Transfer Learning. CVPR 2016

Our goals

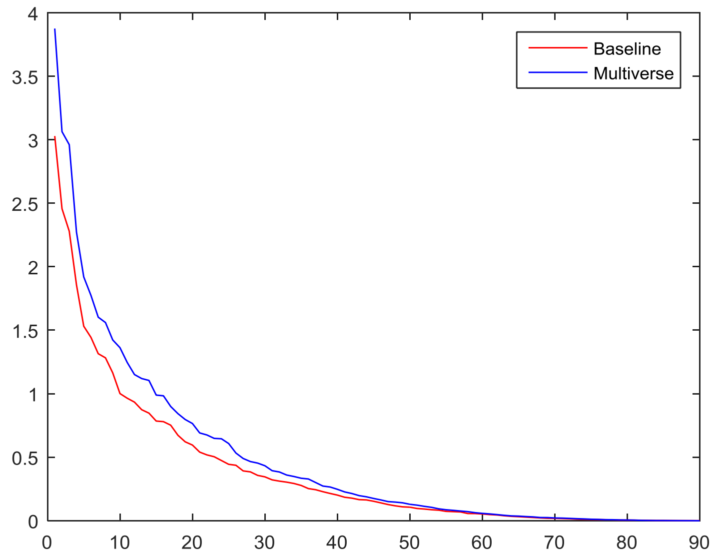
- Reduce the dimensionality of the representation

- Improve the disciminative power of each dimension

- Let the data speak
  No extra parameter

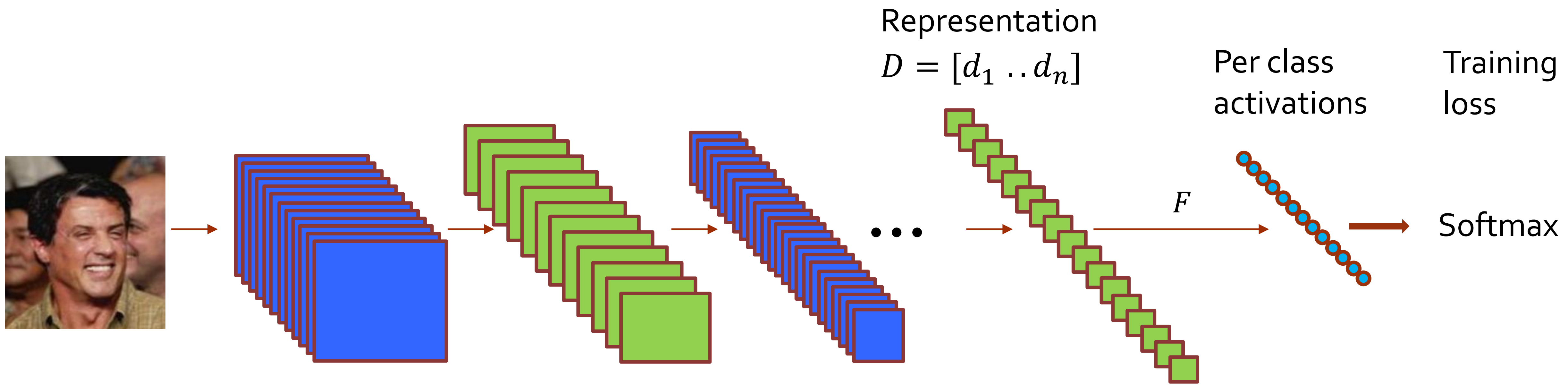


Singular values of the representation



Fisher Spectrum

A conventional network



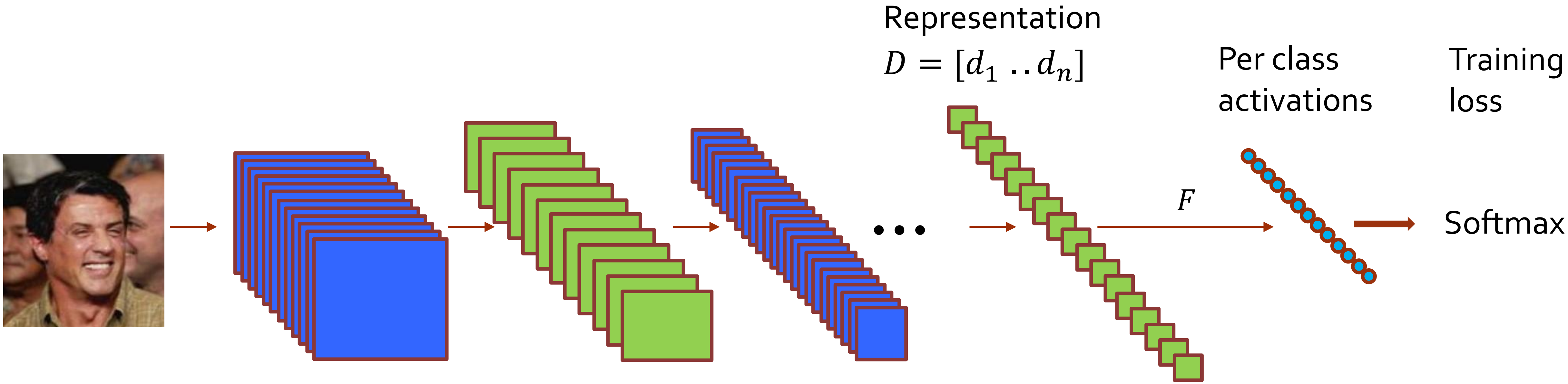
$F \in \mathbb{R}^{d \times c}$

where c is #classes

d is the representation dim

A conventional network

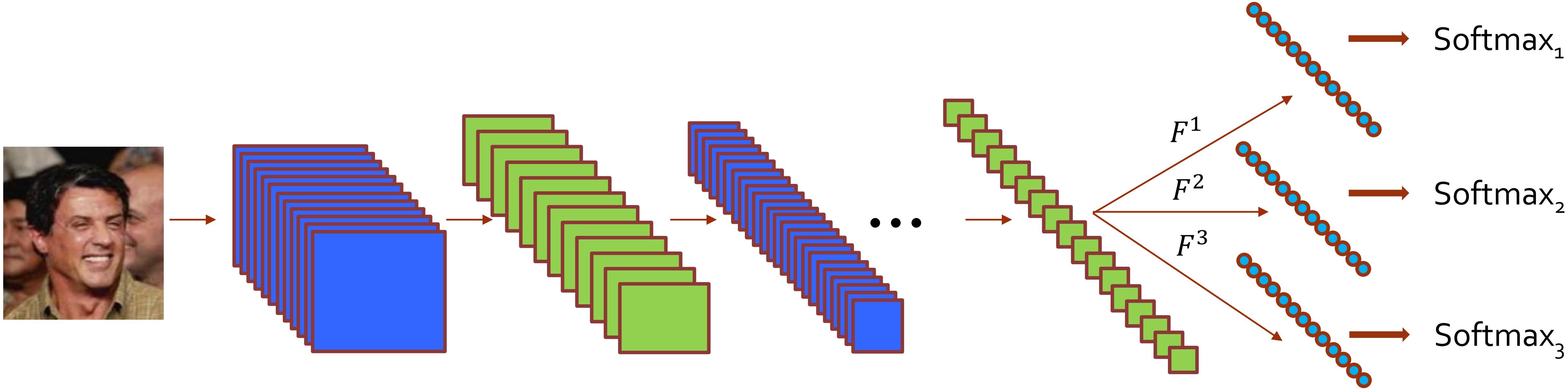
$$\sum_{i=1}^{n} -log \frac{e^{d_i^{T} f_{y_i}+b_{y_i}}}{\sum_{j=1}^{c} e^{d_i^{T} f_j \; +b_j}}$$



$F \in \mathbb{R}^{d \times c}$
where c is #classes
d is the representation dim

The multiverse network

One network, multiple parallel activations



$F^i \in \mathbb{R}^{d \times c}$

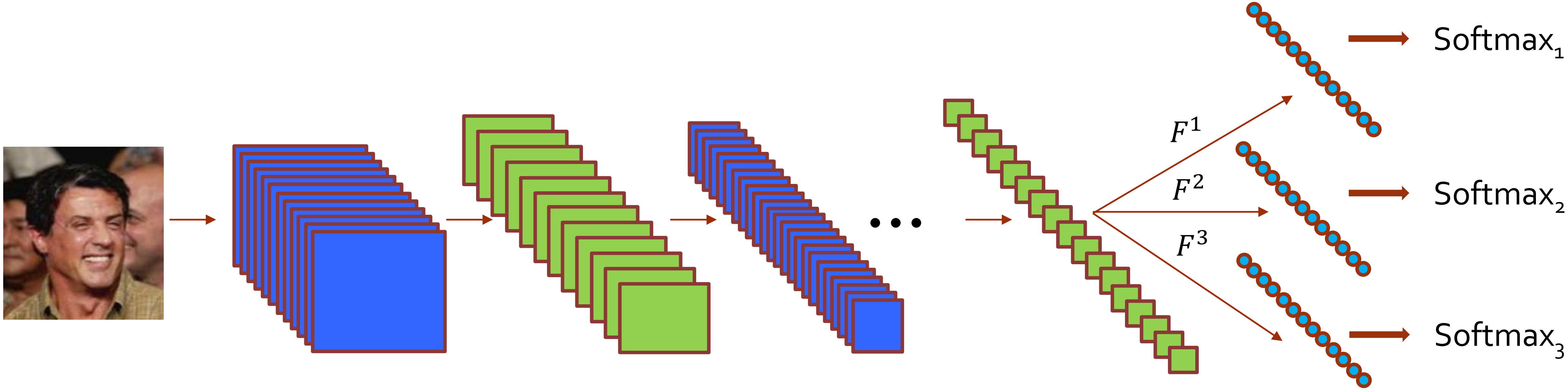
where c is #classes

d is the representation dim

The multiverse network -- loss

$$\frac{1}{m}\sum \text{loss}_i$$

m is the multiplicity



$F^i \in \mathbb{R}^{d \times c}$

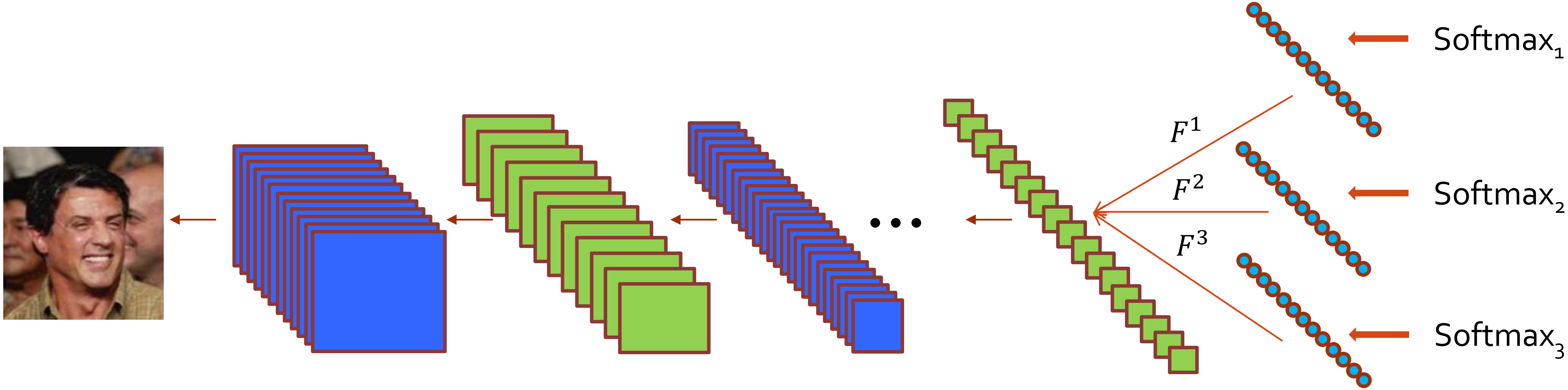
where c is #classes

d is the representation dim

The multiverse network -- backprop

$$\frac{1}{m}\sum grad_i$$

m is the multiplicity



$F^i \in \mathbb{R}^{d \times c}$

where c is #classes

d is the representation dim

Enforcing orthogonality

- Enforce orthogonal solutions:

$$F^1 = [f_1^1, f_2^1, \dots, f_c^1]$$

$$F^2 = [f_1^2, f_2^2, \dots, f_c^2]$$

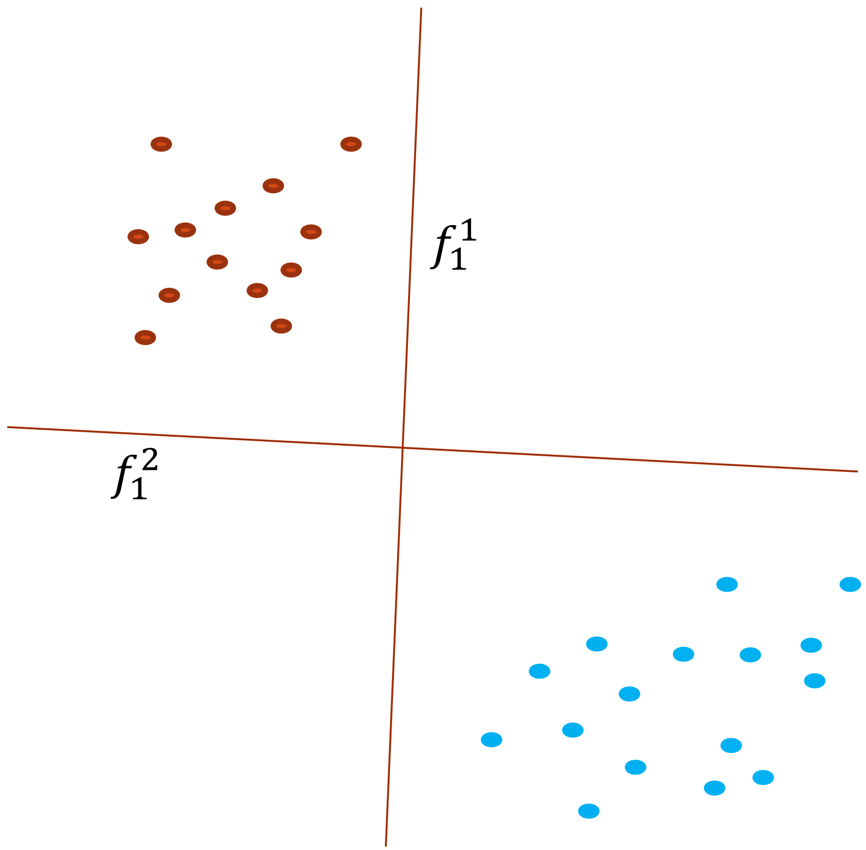
$$\forall_j f_j^1 \perp f_j^2$$

- Practically, the loss used is:

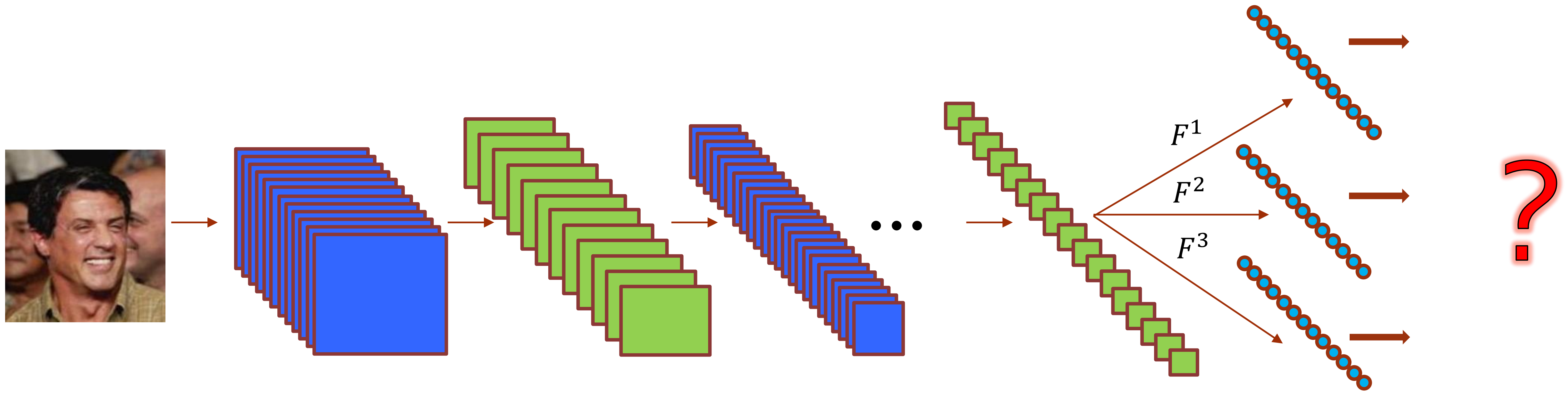
$$L' = \sum_{i=1}^{n} -log \frac{e^{d_i^\top f_{y_i}^1 + b_{y_i}^1}}{\sum_{j=1}^{c} e^{d_i^\top f_j^1 + b_j^1}} - log \frac{e^{d_i^\top f_{y_i}^2 + b_{y_i}^2}}{\sum_{j=1}^{c} e^{d_i^\top f_j^2 + b_j^2}}$$

$$+\lambda_1 \|F^1\|_2 + \lambda_1 \|F^2\|_2 + \lambda_1 \|b^1\|_2 + \lambda_1 \|b^2\|_2$$

$$+\lambda_2 \sum_{j=1}^{c} |f_j^{1\top} f_j^2|$$



The multiverse network during test

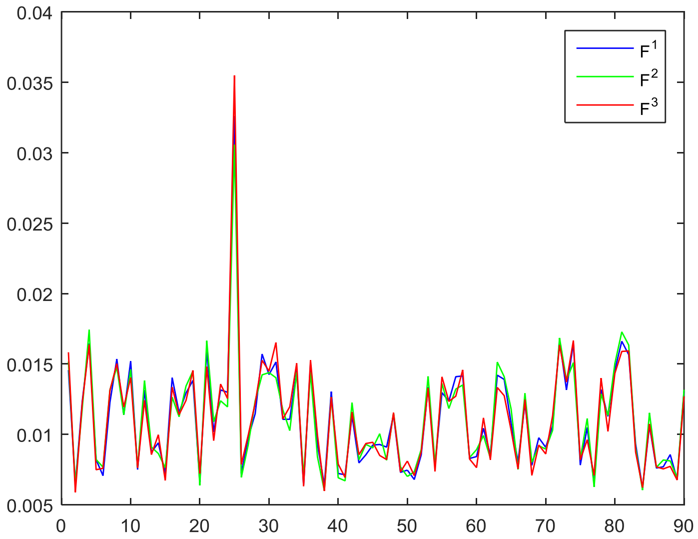


Surprising properties emerge

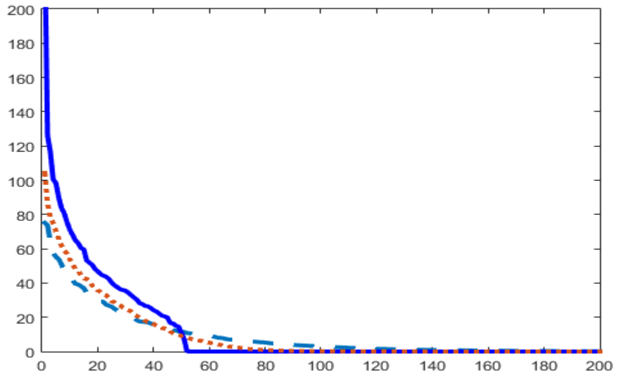
I. The solutions are indeed orthogonal…
… but they all give the same softmax probabilities

II. The dimensionality drops abruptly

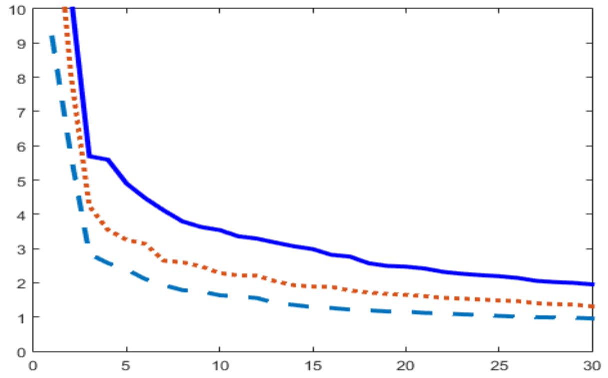
III. The Fisher Spectrum improves



Softmax probabilities (90 classes)



Representation singular values



Fisher spectrum

Cross entropy loss supports multiplicity

- Due to the properties of the softmax, there are multiple ways to get the same probabilities

Lemma 1. The minimizers F^*, b^* of the cross entropy loss L are not unique, and it holds that for any vector $v \in \mathbb{R}^c$ and scalar s, the solutions $F^* + v\mathbb{1}_c^T, b^* + s\mathbb{1}_c$ are also minimizers of L.

$Proof.$ denoting $V = v\mathbb{1}_c^T,\ \boldsymbol{s} = s\mathbb{1}.$

$$L(F^* + V, b^* + \boldsymbol{s}, D, y) =$$

$$-\sum_{i=1}^{n} \log\left(\frac{e^{d_i^T f_{y_i} + d_i^T v + b_{y_i} + s}}{\sum_{j=1}^{c} e^{d_i^T f_j + d_i^T v + b_j + s}}\right)$$

$$= -\sum_{i=1}^{n} \log\left(\frac{e^{d_i^T v + s} e^{d_i^T f_{y_i} + b_{y_i}}}{\sum_{j=1}^{c} e^{d_i^T v + s} e^{d_i^T f_j + b_j}}\right)$$

$$= -\sum_{i=1}^{n} \log\left(\frac{e^{d_i^T v + s} e^{d_i^T f_{y_i} + b_{y_i}}}{e^{d_i^T v + s} \sum_{j=1}^{c} e^{d_i^T f_j + b_j}}\right)$$

$$= -\sum_{i=1}^{n} \log\left(\frac{e^{d_i^T f_{y_i} + b_{y_i}}}{\sum_{j=1}^{c} e^{d_i^T f_j + b_j}}\right) = L(F^*, b^*, D, y)$$

If full rank, then Lemma 1 is IFF

- For full rank representation D the construction shown in Lemma 1 is the only way to obtain multiplicity

Theorem 1. Assume the minimal loss $L^* (D, y)$ is obtained at two solutions F^1, b^1 and F^2, b^2. If $rank(D) = d$, then there exists some vector $v \in \mathbb{R}^c$ and some scalar s such that $F^1 - F^2 = v\mathbb{1}_c^\top$ and $b^1 - b^2 = s\mathbb{1}_c$.

Proof gist:
From the convexity of the cross entropy loss we infer a condition on the null space of the Hessian.
We show that for full rank representation, the Hessian has a zero singular value in only a few restrictive directions.

$Proof$. Let $\Psi = [\psi_1, \psi_2, \dots, \psi_c] = F^2 - F^1$, and let ψ denote the concatenation of the column vectors $\psi_{1\dots c}$ into a single column vector. From convexity:

$$\psi^T \, \nabla^2 \, L(D,y)\Big|_{F^1} \; \psi = \psi^T \frac{\partial L(D,y)^2}{\partial F \partial F}\bigg|_{F^1} \; \psi \; = 0$$

For full rank D, we aim to prove that:

$$\psi_1 = \psi_2 \dots = \psi_c$$

Proof of theorem 1

The hessian can be written:

$$\frac{\partial^2}{\partial F_{ju} F_{j'v}} L(D,y) =$$
$$-\sum_{i=1}^{n} d_{iu} d_{iv} p_i(j)(\delta_{j=j'}(1-p_i(j)) - \delta_{j\neq j'} p_i(j'))$$

After some manipulation:

$$\psi^{T} \frac{\partial^2}{\partial F \partial F} L(D,y)\Big|_{F^1} \psi \;=$$
$$\sum_{j=1}^{c} \sum_{j'=j+1}^{c} \left(\psi_j - \psi_{j'}\right)^T \sum_{i=1}^{n} d_i d_i^T p_i(j) p_i(j')(\psi_j - \psi_j')$$

$$\sum_{i=1}^{n} d_i d_i^T p_i(j) p_i(j')$$ - PD matrix

$$\sum_{j=1}^{c} \sum_{j'=j+1}^{c} \left(\psi_j - \psi_{j'}\right)^T \sum_{i=1}^{n} d_i d_i^T p_i(j) p_i(j')(\psi_j - \psi_j')$$

Vanishes if and only if $\psi_j = \psi_{j'}$

■

… now add orthogonality to the mix

- For full rank representations D multiple **orthogonal** classifiers are only possible for very specific (degenerate) classifier collections

Theorem 2. Assume that $rank(D) = d$, that $d < c$, and that the minimal loss $L^*(D, y)$ is obtained at a solution F^1, b^1. If there exists a second minimizer F^2, b^2 such that for all $j \in [1 \dots c]$ the orthogonality constraint $f_{j^1}^1 \perp f_{j^1}^2$ holds, then F^1 admits to a stringent second order constraint.

Proof gist:
We employ theorem 1 and get equations of the form

$$F^{1T} v = - \begin{pmatrix} \|f_1^1\|^2 \\ \|f_2^1\|^2 \\ \vdots \\ \|f_c^1\|^2 \end{pmatrix}$$

The good news

- It is possible to obtain multiple orthogonal solutions that are almost as good as a single solution

- It requires the existence of small singular values in D

- **Hence the low rank property**

Theorem 3. There exist sets of weights $F^1 = [f_1^1, f_2^1, \dots, f_c^1], b^1, F^2 = [f_1^2, f_2^2, \dots, f_c^2], b^2$ which are orthogonal as follows $\forall j \;\; f_j^1 \perp \; f_j^2$, for which the joint loss:

$$J(F^1, b^1, F^2, b^2, D, y) = L(F^1, b^1, D, y) + L(F^2, b^2, D, y)$$

is bounded by

$$2L^*(D, y) \leq J(F^1, b^1, F^2, b^2, D, y) \leq 2L^*(D, y) \; + A\lambda_d$$

where A is a bounded parameter.

The good news (enlarged)

Theorem 3. There exist sets of weights
$F^1 = [f_1^1, f_2^1, \dots, f_c^1], b^1, F^2 = [f_1^2, f_2^2, \dots, f_c^2], b^2$
which are orthogonal, i.e., $\forall j \;\; f_j^1 \perp f_j^2$,
for which the joint loss:

$$J(F^1, b^1, F^2, b^2, D, y) = L(F^1, b^1, D, y) + L(F^2, b^2, D, y)$$

is bounded by

$$2L^*(D, y) \leq J(F^1, b^1, F^2, b^2, D, y) \leq 2L^*(D, y) \; + A\lambda_d$$

where A is a bounded parameter,
λ_d is the smallest singular value of D.

Proving Theorem 3

Proof gist: Using series expansion around $F^1 = F^*$

$$L(F^1 + \Psi, b^1) = L(F^1 + \Psi, b^1) + (\vec{\nabla}^T \psi)\, L(D, y)\Big|_{F^1, b^1} + R(\psi)$$

The remainder term (Lagrange form):

$$R(\psi) = \frac{1}{2}\left(\vec{\nabla}^T \psi\right)^2 L(D, y)\Big|_{\theta}$$

$$= \frac{1}{2}\sum_{j=1}^{c}\sum_{j'=j+1}^{c}\left(\psi_j - \psi'_j\right)^T \sum_{i=1}^{n} d_i d_i^T p_i(j) p_i(j')\left(\psi_j - \psi'_j\right)$$

$$\leq \frac{1}{2}\sum_{j=1}^{c}\sum_{j'=j+1}^{c}\left(\psi_j - \psi'_j\right)^T DD^T\left(\psi_j - \psi'_j\right)$$

Theorem 3 generalization. There exist sets of weights $F^1 = [f_1^1, f_2^1, \dots, f_c^1], b^1 \dots F^m = [f_1^m, f_2^m, \dots, f_c^m], b^m$ which are orthogonal as follows $\forall ijk\ \ f_j^i \perp f_j^k$, for which the joint loss:

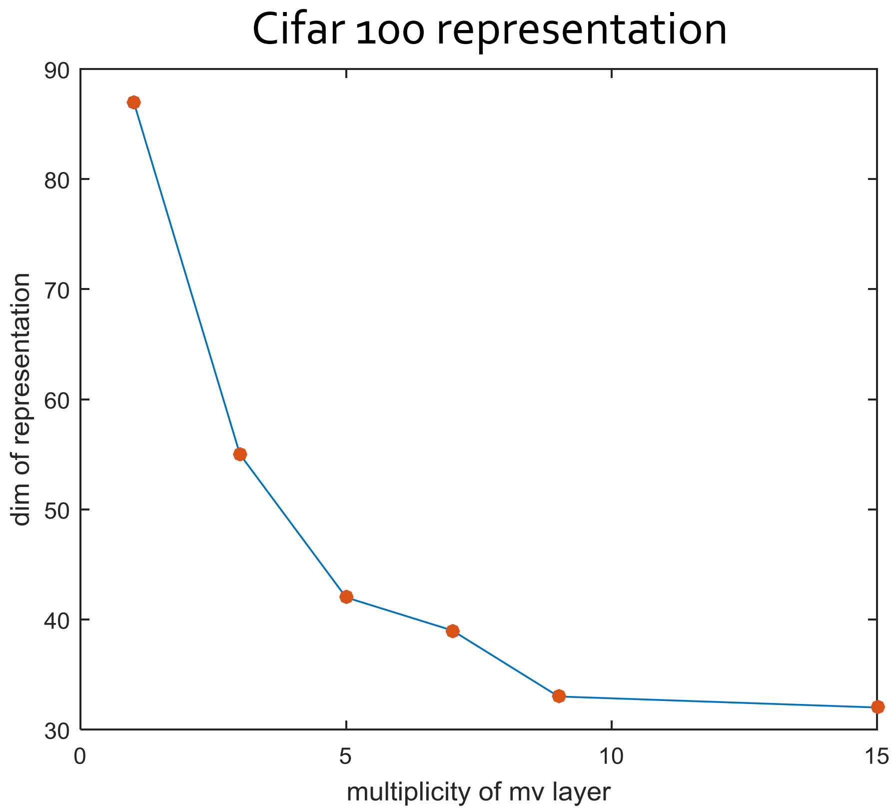
$$J(F^1, b^1 \dots F^m, b^m, D, y) = \sum_{r=1}^{m} L(F^r, b^r, D, y)$$

$$mL^*(D, y) \leq J(F^1, b^1 \dots F^m, b^m, D, y)$$

$$\leq mL^*(D, y) + \sum_{l=1}^{m-1} A_l\, \lambda_{d-l+1}$$

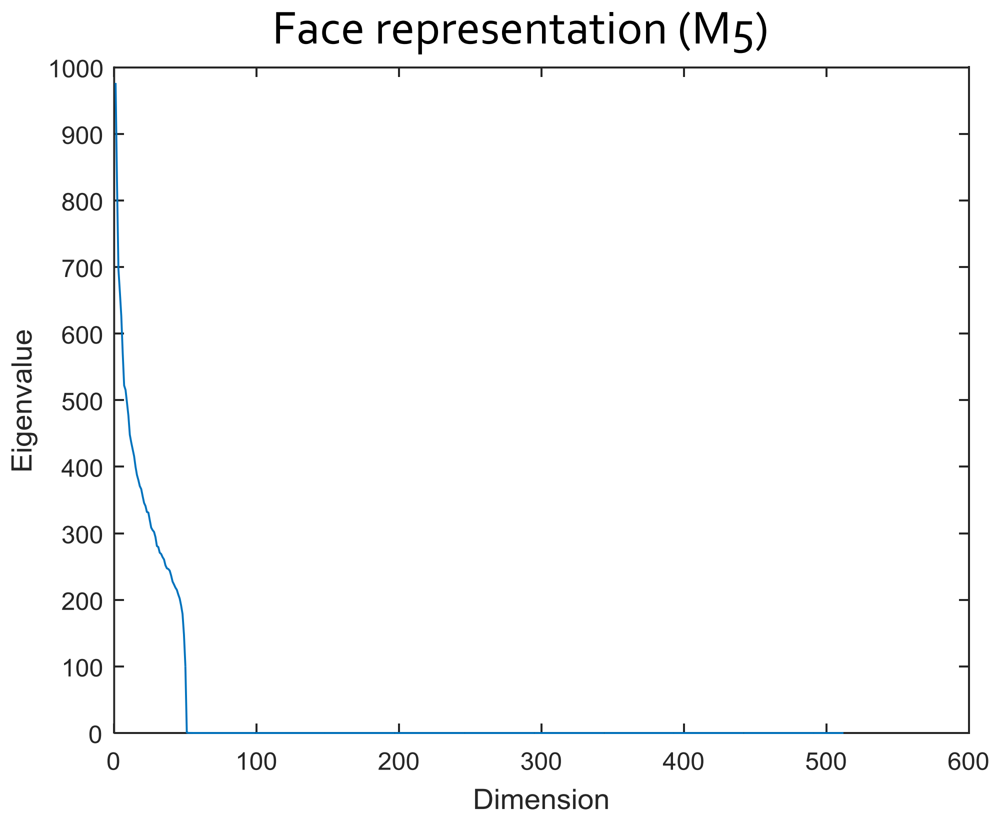
Compact representation

- Dim of representation turns out to be extremely compact
- No loss in energy
- **Convergence to "natural" dim**



Compact representation

- Dim of representation turns out to be extremely compact
- No loss in energy
- **Convergence to “natural” dim**



51 dimensional representation!

Fisher Spectrum betterment

Between class covariance:

$$S_b = \frac{1}{n}\sum_{j=1}^{c} n_j(\mu - \mu_j)(\mu - \mu_j)^T$$

Within class covariance:

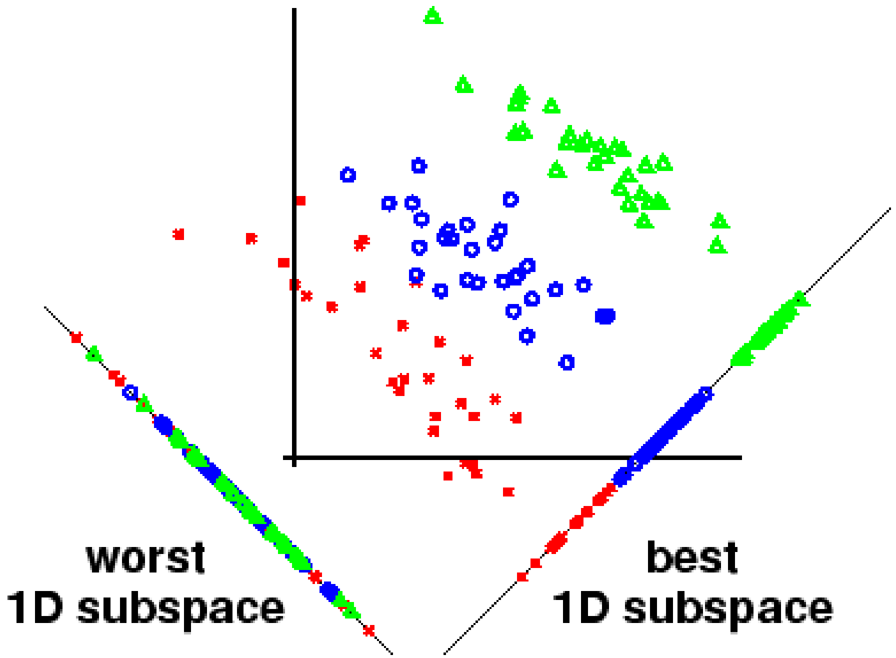
$$S_w = \frac{1}{n}\sum_{j=1}^{c}\sum_{i \in I_j}(d_i - \mu_j)(d_i - \mu_j)^T$$

Fisher spectrum:

$$S_b v = \gamma S_w v$$

Fisher ratio:

$$\sigma(v, S_b, S_w) = \frac{v^T S_b v}{v^T S_w v}$$



How to measure post-transfer success

- The Joint Bayesian (JB) method is a popular learning face verification method
  Chen et al. Bayesian face revisited: a joint formulation. ECCV, 2012

- Two densities are learned

  $P(d, d'|H)$ and $P(d, d'\,|I)$

  H: Same hypothesis
  I: Not same hypothesis

Good Fisher Spectrum ➔ Good JB separation

Theorem 5. Given data D, mean μ and labels y, for any centered data point $\widehat{d_i} = d_i - \mu$, we denote $d_i' = (S_b + S_w)^{-1}\widehat{d_i}$. Given two centered data points $\widehat{d_1}, \widehat{d_2}$ such that the fisher ratios $\sigma(d_1', S_b, S_w), \sigma(d_2', S_b, S_w) < T$, it holds that:

JB Probability of same person

$$1 - 2T \leq \frac{\log P(d_1, d_2|H) + \eta_1}{\log P(d_1, d_2|I) + \eta_2} \leq 1 + 6T$$

JB Probability of different persons

"Difficult to tell if same or not-same if all the difference between the faces is in directions with low fisher scores"

The emergence of high fisher scores

- We prove the emergence of better fisher spectrum using S_w orthogonality.

$F^1 = [f_1^1, f_2^1, \dots, f_c^1]$

$F^2 = [f_1^2, f_2^2, \dots, f_c^2]$

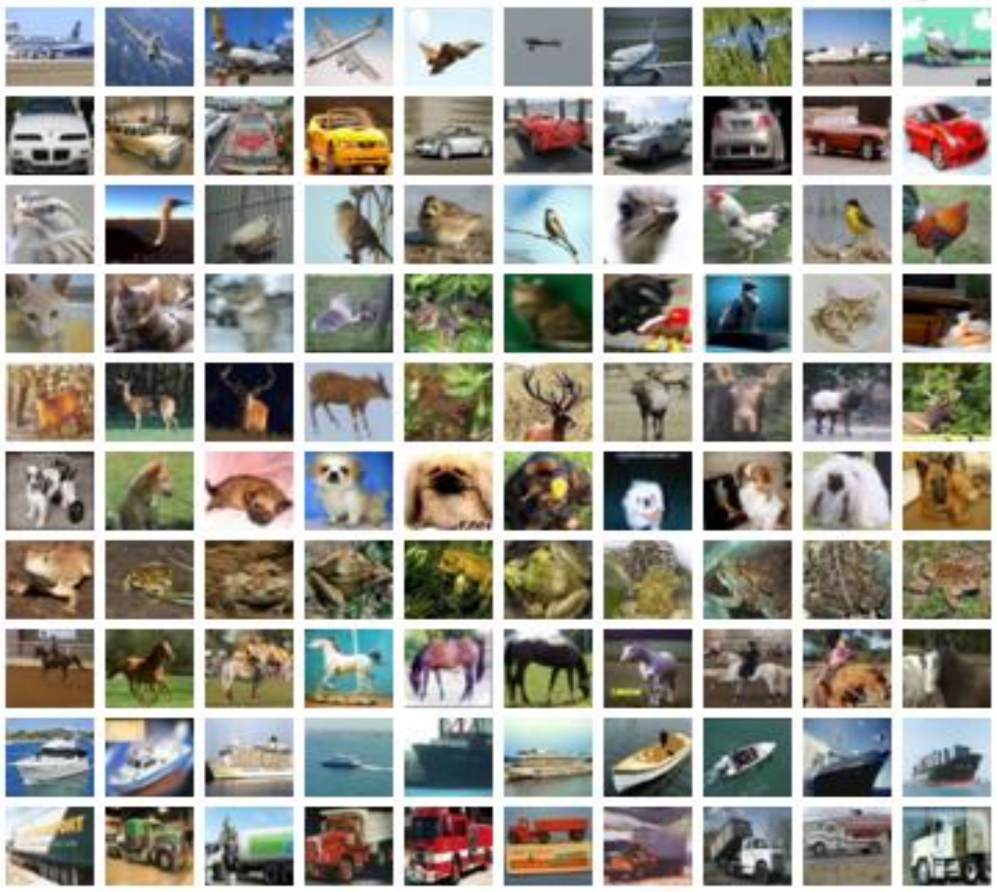
$\forall_j, f_j^1 \perp S_w f_j^2$

- Experimentally, improved fisher spectrum is demonstrated in both types of orthogonality

Theorem 6. Let $f^1 \dots f^m$ be a set of m classifiers that are S_w-orthogonal for data D and labels y, and let $\gamma = [\gamma_1 \dots \gamma_d]$ denote the Fisher spectrum. Given that $\forall\ 1 \leq r \leq m$, for some value θ, $\sigma(f^r, S_b, S_w) \geq \theta$, it holds that $\sum_{k=1}^{d} \gamma_k \geq \sqrt{m}\theta$.

Experiments

CIFAR-100 thumbnail recognition



LFW face recognition

Same

Not same

CIFAR-100 thumbnail recognition

- CIFAR-100
  - Learn on 90 classes
  - Transfer to the remaining 10

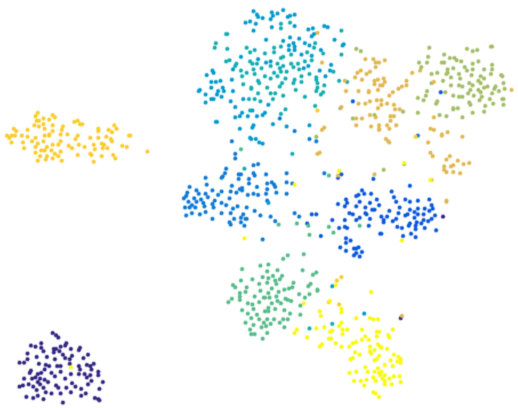
- Architecture: NIN

  Lin, Chen, Yan. Network in Network. ICLR, 2014

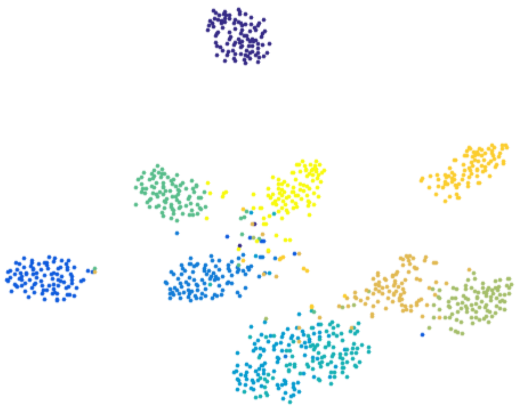
Layer	Filter/Stride	#Channel	#Filter
Conv11	5×5 / 1	3	192
Conv12	1×1 / 1	192	160
Conv13	1×1 / 1	160	96
Pool1	3×3 / 2	96	–
Dropout1-0.5	–	–	–
Conv21	5×5 / 1	96	192
Conv22	1×1 / 1	192	192
Conv23	1×1 / 1	192	100
Pool2	3×3 / 2	192	–
Dropout1-0.5	–	–	–
Conv31	3×3 / 1	192	192
Conv32	1×1 / 1	192	192
Conv33	1×1 / 1	192	100
Avg Pool	7×7 / 1	100	–
FC	1×100 / 1	100	100

CIFAR-100 Results

Domain	Source	Target (transfer)	
Metric	Val error	Cosine	JB
M1	0.340	0.789	0.800
M2	0.340	0.791	0.804
M2 (S_w-orthogonal)	0.344	0.798	0.803
M3	0.345	0.801	0.812
M3 (S_w-orthogonal)	0.346	0.799	0.811
M4	0.351	0.807	0.82
M4 (S_w-orthogonal)	0.353	0.808	0.823
M5	0.360	0.812	0.833
M5 (S_w-orthogonal)	0.362	0.811	0.831
M6	0.369	0.816	0.838
M6 (S_w-orthogonal)	0.371	0.816	0.834
M7	0.375	0.815	0.831
M7 (S_w-orthogonal)	0.377	0.816	0.830



Baseline (M1)



Multiverse (M5)

LFW face recognition

- Learn on CASIA dataset

- Use the Scratch architecture from the CASIA paper

  Yi, Lei, Liao, Li. Learning face representation from scratch. arXiv, 2014

- Transfer to LFW

- The network used

Layer	Filter/Stride	#Channel	#Filter
Conv11	$3 \times 3 / 1$	1	32
Conv12	$3 \times 3 / 1$	32	64
Max Pool	$2 \times 2 / 2$	64	–
Conv21	$3 \times 3 / 1$	64	64
Conv22	$3 \times 3 / 1$	64	128
Max Pool	$2 \times 2 / 2$	128	–
Conv31	$3 \times 3 / 1$	128	96
Conv32	$3 \times 3 / 1$	96	192
Max Pool	$2 \times 2 / 2$	192	–
Conv41	$3 \times 3 / 1$	192	128
Conv42	$3 \times 3 / 1$	128	256
Max Pool	$2 \times 2 / 2$	256	–
Conv51	$3 \times 3 / 1$	256	160
Conv52	$3 \times 3 / 1$	160	320
Avg Pool	$6 \times 6 / 1$	320	–
Dropout1-0.3	–	–	–
FC	$1 \times 320 / 1$	320	100

LFW results

Domain	Source	Target (transfer)		
Metric	Val error	Cosine	JB on source	JB on LFW splits
CASIA trained M1	0.07	0.962 ± 0.0032	0.966 ± 0.0022	0.970 ± 0.0016
CASIA trained M1 (2)	0.07	0.962 ± 0.0021	0.966 ± 0.0019	0.971 ± 0.0022
CASIA trained M1 (3)	0.07	0.961 ± 0.0022	0.966 ± 0.0013	0.971 ± 0.0015
Ensemble of 3 CASIA M1		0.968 ± 0.0019	0.972 ± 0.0021	0.975 ± 0.0025
CASIA trained M2	0.08	0.970 ± 0.0021	0.974 ± 0.0017	0.976 ± 0.0016
CASIA trained M3	0.11	0.972 ± 0.0012	0.977 ± 0.0015	0.980 ± 0.0034
CASIA trained M3 (2)	0.11	0.971 ± 0.0031	0.977 ± 0.0028	0.979 ± 0.0027
CASIA trained M5 (1)	0.12	0.973 ± 0.0011	0.978 ± 0.0014	0.981 ± 0.0019
CASIA trained M5 (2)	0.12	0.972 ± 0.0015	0.977 ± 0.0019	0.980 ± 0.0031
3rd party DB, M5	0.12	0.982 ± 0.0034	0.982 ± 0.0031	0.988 ± 0.0035
Two network ensemble		0.985 ± 0.0029	0.990 ± 0.0027	0.991 ± 0.0027

Compared to SOTA

Method	Single network	Ensemble result	#nets	Training dataset
M5	0.9814 ± 0.0019	–		CASIA [41]
M5, 3rd party DB	0.9883 ± 0.0035	0.9905 + 0.0027	2	proprietary 800k images
DeepFace [32]	0.9700 ± 0.0087	0.9735 ± 0.0025	7	proprietary, 4M images
DeepID [28]	–	0.9745 ± 0.0026	25	proprietary,160k
Original scratch [41]	0.9773 ± 0.0031	–	1	CASIA [41]
Web-Scale Training [33]	0.9800	0.9843	4	proprietary, 500M images
MSU TR [38]	0.9745 ± 0.0099	0.9823 ± 0.0068	7	CASIA [41]
MMDFR [5]	0.9843 ± 0.0020	0.9902 ± 0.0019	8	proprietary,500k
DeepID2 [25]	0.9633	0.9915 ± 0.0013	25	proprietary,160k
DeepID2+ [29]	0.9870	0.9947 ± 0.0012	25	proprietary,290k
FaceNet [23]	0.9887 ± 0.0015	0.9963 ± 0.0009	8	proprietary, 200M
FR+FCN [43](*)	–	0.9645 ± 0.0025	5	CelebFaces [27], 88k
betaface.com(*)	–	0.9808 ± 0.0016	NA	NA
Uni-Ubi(*)	–	0.9900 ± 0.0032	NA	NA
Face++ [42](*)	–	0.9950 ± 0.0036	4	proprietary, 5M face images
DeepID3 [26](*)	–	0.9953 ± 0.0010	25	proprietary,300k
Tencent-BestImage(*)	–	0.9965 ± 0.0025	20	proprietary, 1M face images
Baidu [19](*)	–	0.9977 ± 0.0006	10	proprietary, 1.2M face images
AuthenMetric(*)	–	0.9977 ± 0.0009	25	proprietary, 500k face images

Excellent single network result

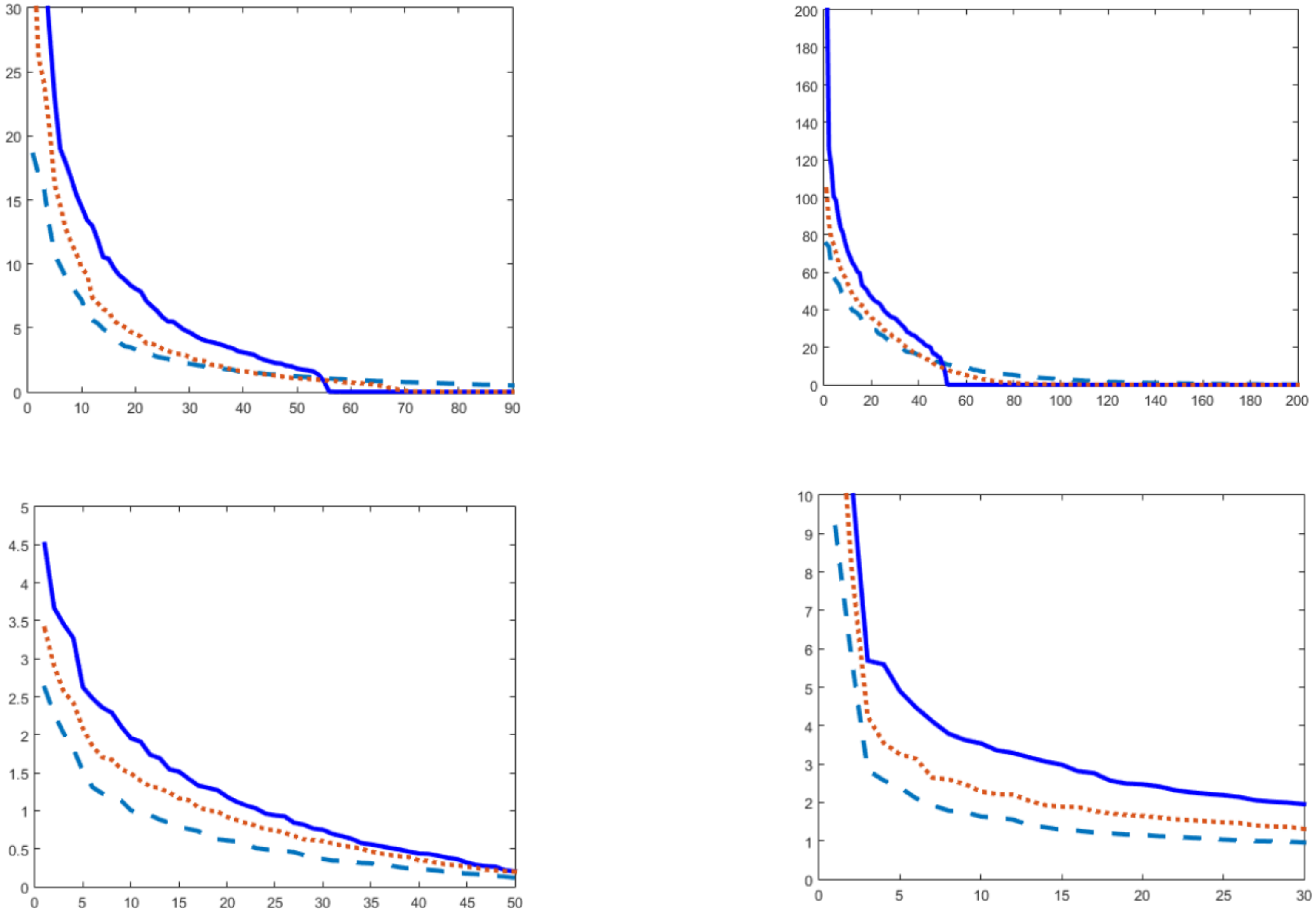
Relativley small dataset

Extrmely compact representation 51D

CIFAR-100 LFW

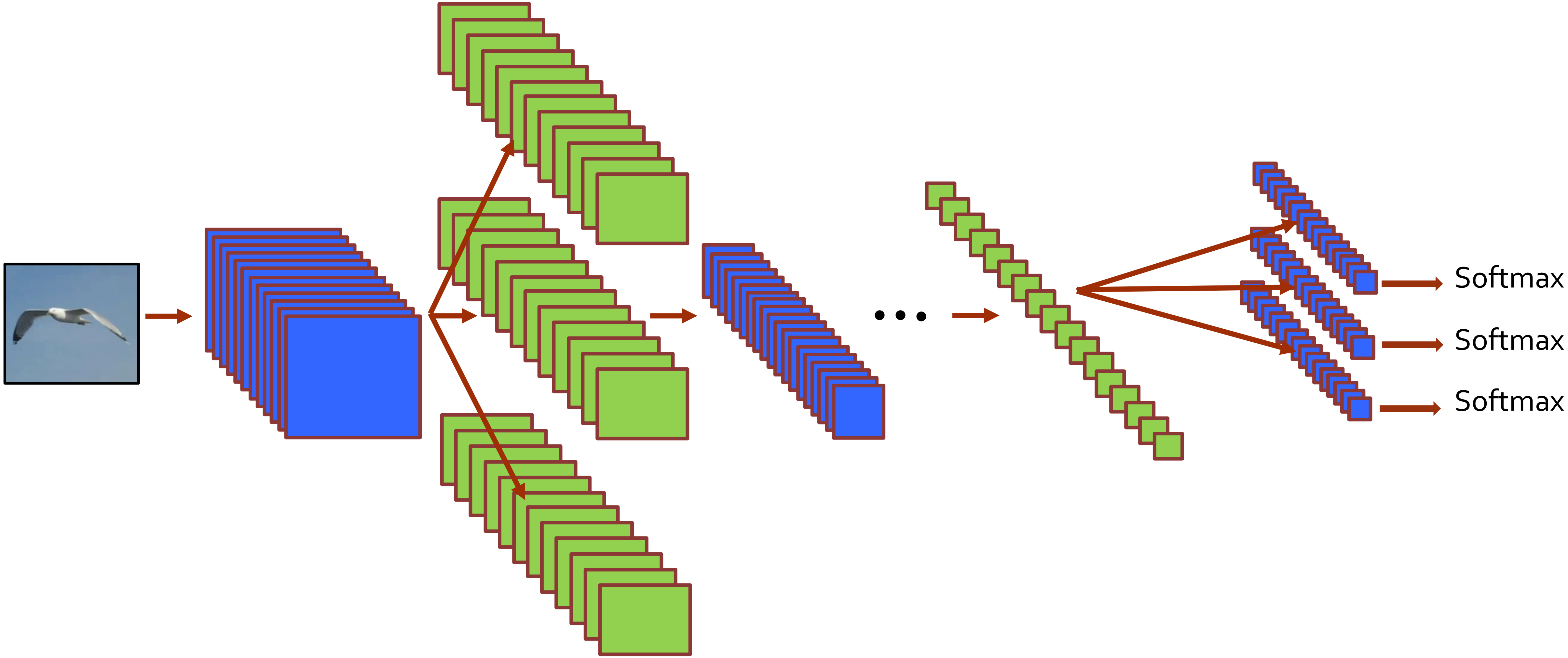
Representation singular values

Representation fisher spectrum

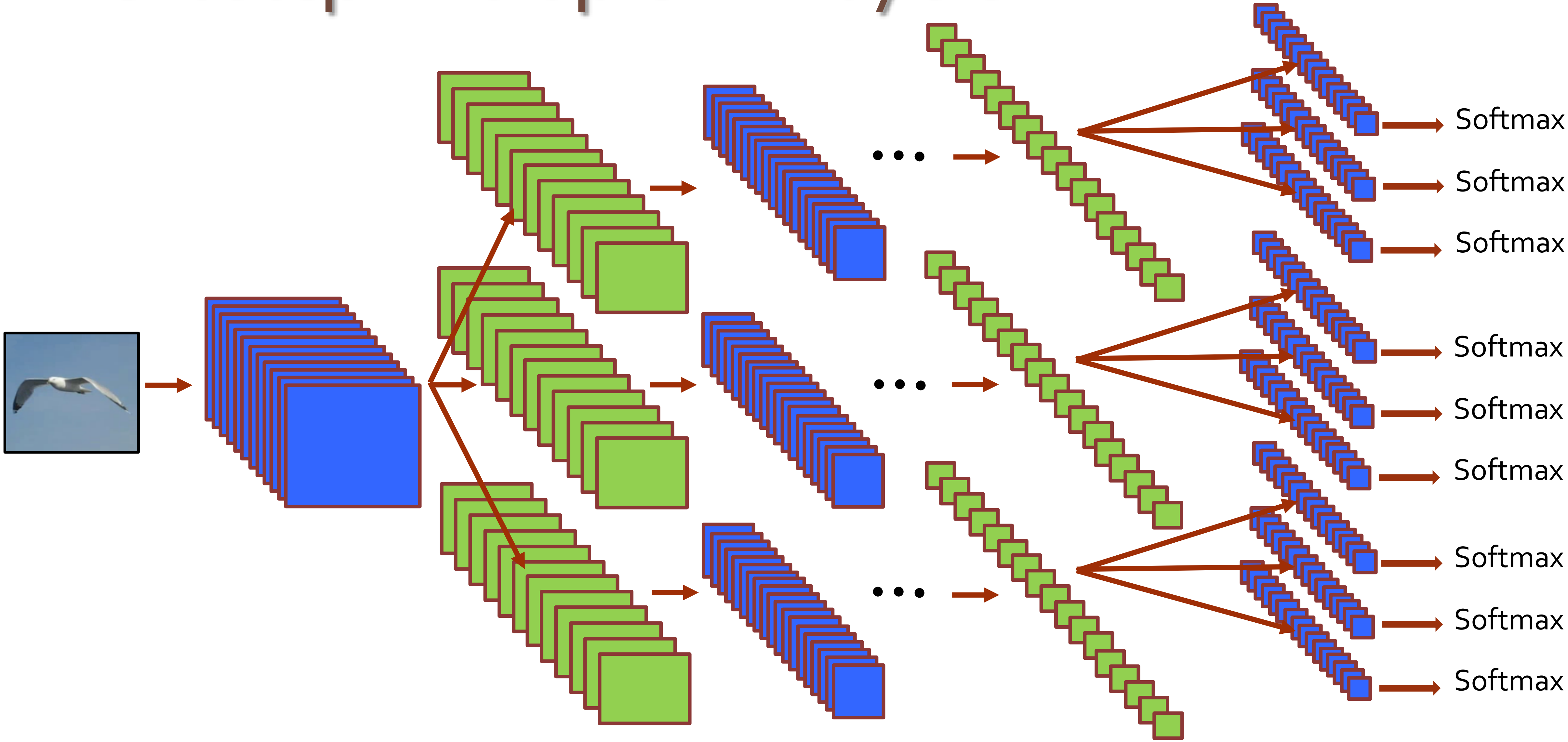


Solid blue M5, Dotted red M3, Dashed magenta M1

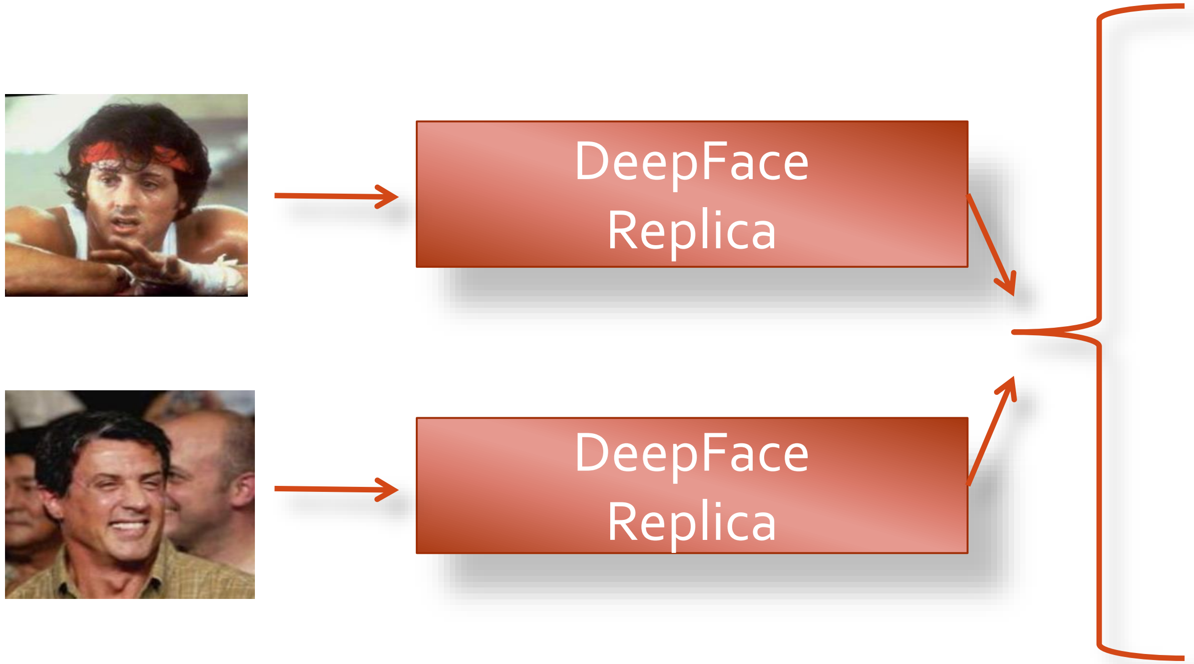
Next step: multiple mv layers



Next step: multiple mv layers



Can we use a network instead of JB?



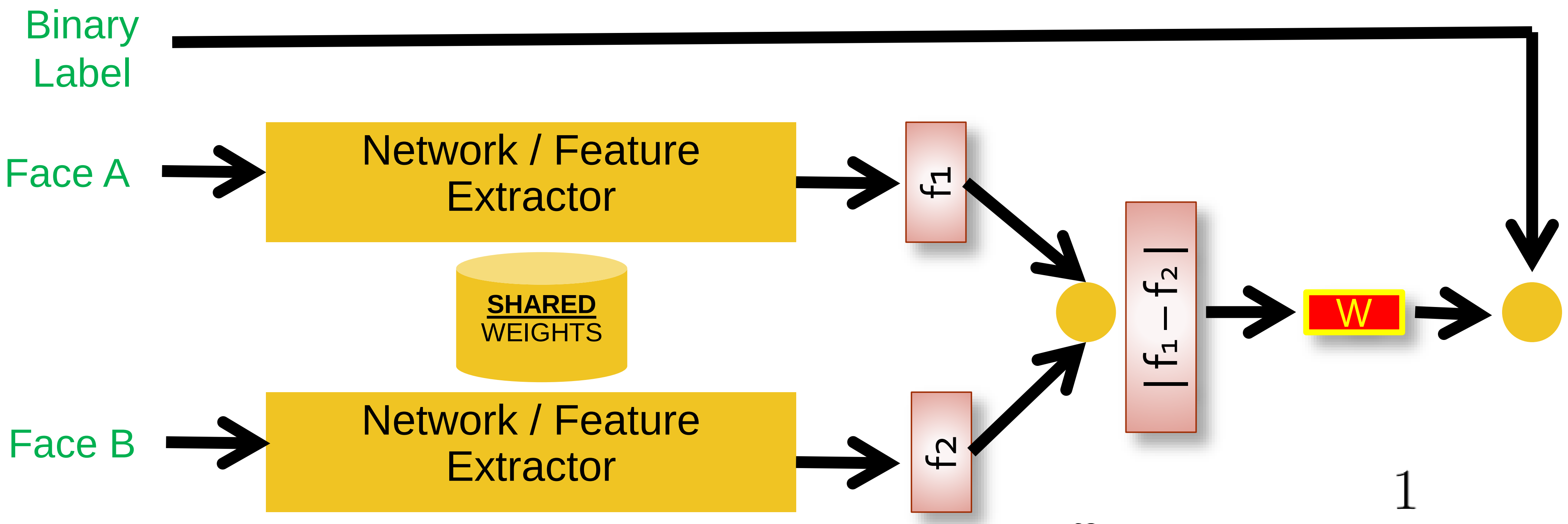
(a) Cosine angle

(b) Kernel Methods

(c) **Siamese Network**

Chopra, Hadsell, LeCun. Learning a similarity metric discriminatively, with application to face verification. CVPR, 2005.

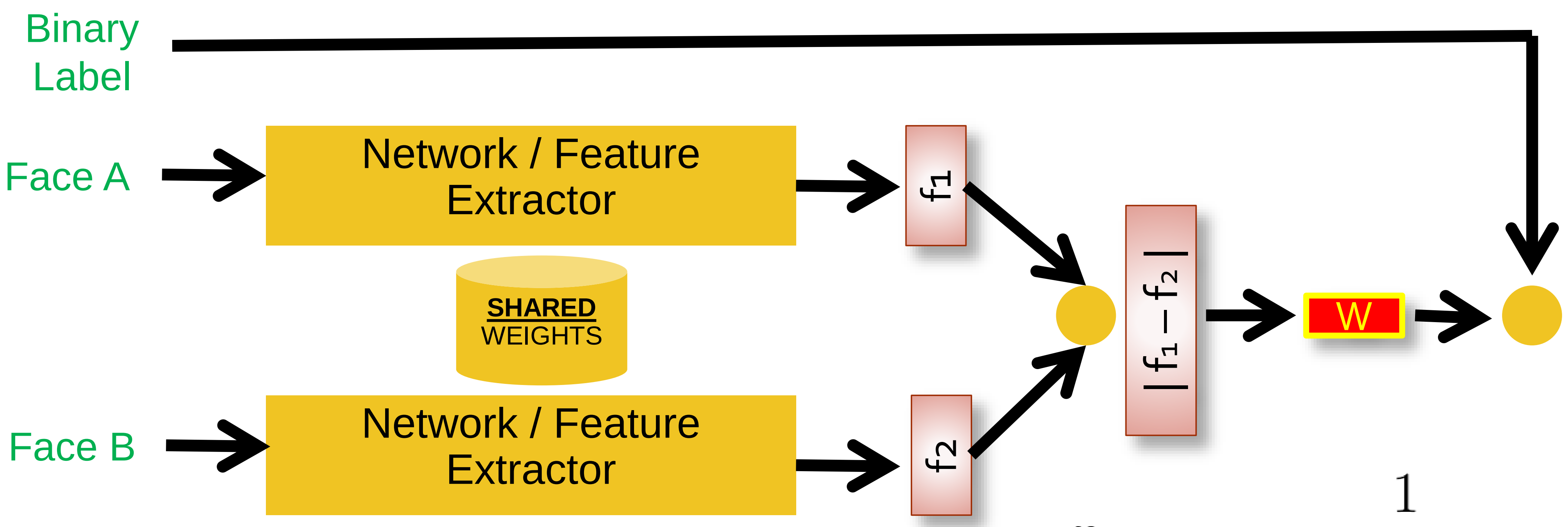
Deep Siamese Architecture



$$p = \frac{1}{1 + e^{-(W * |f_1 - f_2| + b)}}$$

$$E = -ylog(p) - (1 - y)log(1 - p)$$

Deep Siamese Architecture



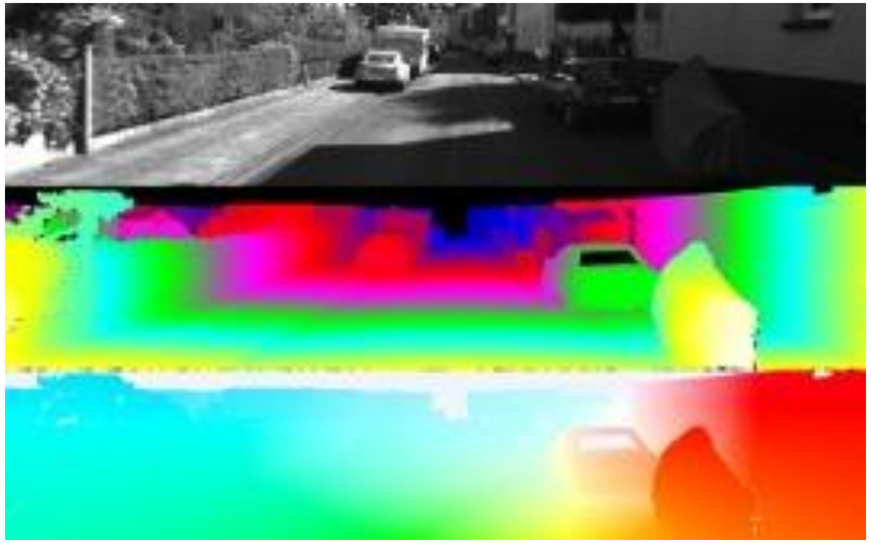
$$p = \frac{1}{1 + e^{-(W*|f_1 - f_2| + b)}}$$

$$E = -y log(p) - (1 - y) log(1 - p)$$

Q5: Is binary classification loss the most appropriate loss for a Siamese Architecture?
A: No. Gadot and Wolf. PatchBatch. *CVPR 2016*.

Optical flow

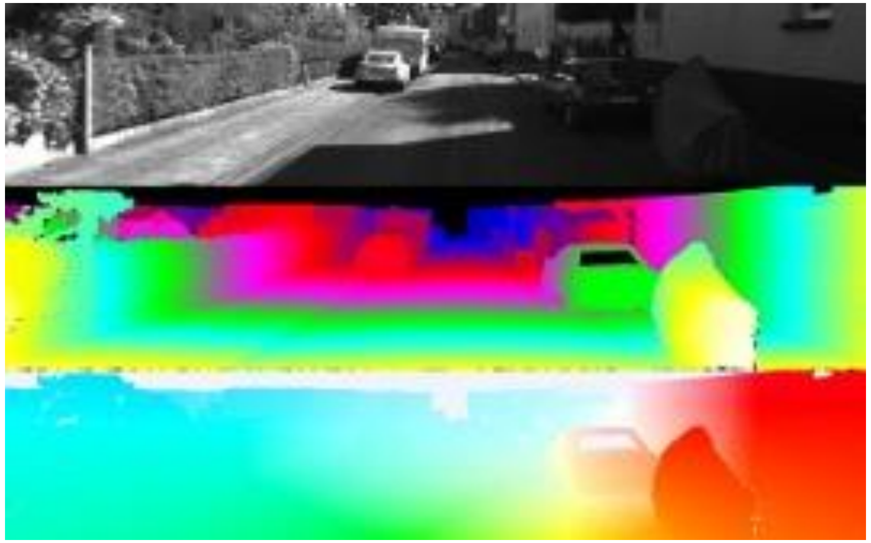
Given multiple image compute the motion field between them.



Architecture: from a patch to a representation

Layer	Filter/Stride	Output size
Input	–	$1 \times 51 \times 51$
Conv1	3×3 / 1	$32 \times 49 \times 49$
Batch Normalization	–	$32 \times 49 \times 49$
Max Pool	2×2 / 2	$32 \times 25 \times 25$
Conv2	3×3 / 1	$64 \times 23 \times 23$
Batch Normalization	–	$64 \times 23 \times 23$
Max Pool	2×2 / 2	$64 \times 12 \times 12$
Conv3	3×3 / 1	$128 \times 10 \times 10$
Batch Normalization	–	$128 \times 10 \times 10$
Max Pool	2×2 / 2	$128 \times 5 \times 5$
Conv4	3×3 / 1	$256 \times 3 \times 3$
Batch Normalization	–	$256 \times 3 \times 3$
Max Pool	2×2 / 2	$256 \times 2 \times 2$
Conv5	2×2 / 1	$512 \times 1 \times 1$
Batch Normalization	–	$512 \times 1 \times 1$

Table 1. The network model for representing a grayscale 51×51 input patch as $512D$ vector. The Batch Normalization is our fine-grained variant. Leaky ReLU units [26] (with $\alpha = 0.1$) are used as activation functions following the five batch normalization layers.



DRLIM type Loss

Hadsell, Chopra, LeCun. Dimensionality reduction by learning an invariant mapping. CVPR 2006.

Orig DrLIM

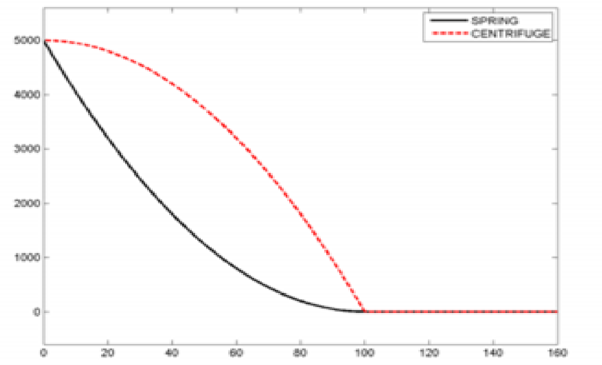
$$(1-Y)\frac{1}{2}D_w^2 + (Y)\frac{1}{2}\{\max(0, m - D_w)\}^2$$

DRLIM type Loss

Hadsell, Chopra, LeCun. Dimensionality reduction by learning an invariant mapping. CVPR 2006.

Orig DrLIM (spring model) $$(1-Y)\frac{1}{2}D_w^2 + (Y)\frac{1}{2}\{\max(0, m - D_w)\}^2$$

CENT-DrLIM $$(1-Y)D_w^2+(Y)\{\max(0,m^2-D_w^2)\}$$



(a)

DRLIM type Loss

Hadsell, Chopra, LeCun. Dimensionality reduction by learning an invariant mapping. CVPR 2006.

Orig DrLIM

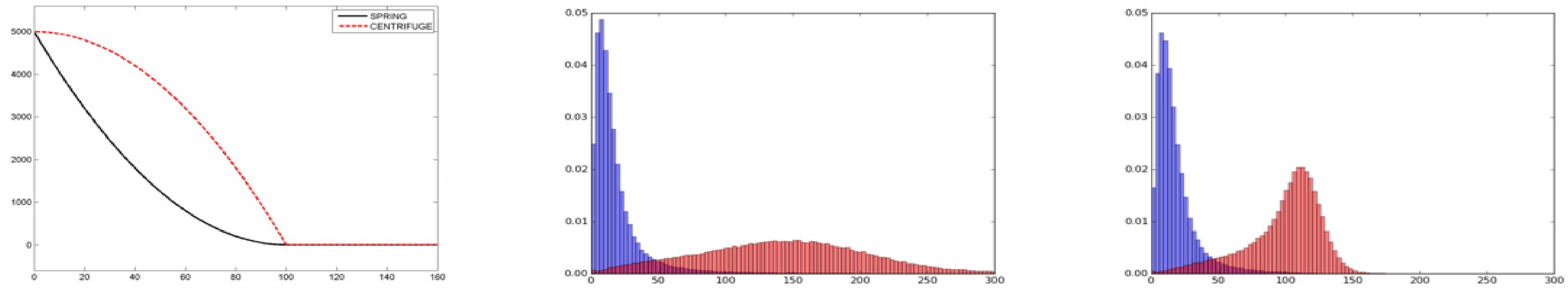
$$(1 - Y)\frac{1}{2}D_w^2 + (Y)\frac{1}{2}\{\max(0, m - D_w)\}^2$$

CENT-DrLIM

$$(1-Y)D_w^2+(Y)\{\max(0, m^2-D_w^2)\}$$

CENT-DrLIM+SD

$$(1-Y)\lambda D_w^2+(Y)\lambda\{\max(0, m^2-D_w^2)\}+(1-\lambda)(\sigma_0+\sigma_1)$$



(a) (b) (c)

Benchmarks - KITTI2012/ KITTI2015

- Raw Optical Flow on KITTI2012 validation set - **~8% err**

Method	Out-Noc	Running time
PatchBatch-ACCRTE-PS71	5.29%	60.5s
PatchBatch-ACCURATE	5.44%	50.5s
PH-Flow [39]	5.76%	800s
FlowFields [1]	5.77%	23s
CPM-Flow (anon.)	5.80%	2s
NLTGV-SC [30]	5.93%	16s
PatchBatch-FAST	5.94%	25.5s
DDS-DF [37]	6.03%	1m
TGV2ADCSIFT [5]	6.20%	12s
DiscreteFlow [28]	6.23%	3m

Table 4. Top 10 KITTI2012 Pure Optic Flow Algorithms as published on the submission date. Out-Noc is the percentage of pixels with euclidean error > 3 pixels out of the non-occluded pixels

Method	Fl-all	Running time
PatchBatch-ACCURATE	21.69%	50.5s
DiscreteFlow [28]	22.38%	3min
CPM-Flow (anon.)	24.24%	2s
EpicFlow [32]	27.10%	15s
FilteringFlow (anon.)	28.50%	116s
DeepFlow [38]	29.18%	17s
HS [35]	42.18%	2.6m
DB-TV-L1 [40]	47.97%	16s
HAOF [6]	50.29%	16.2s
PolyExpand [14]	53.32%	1s

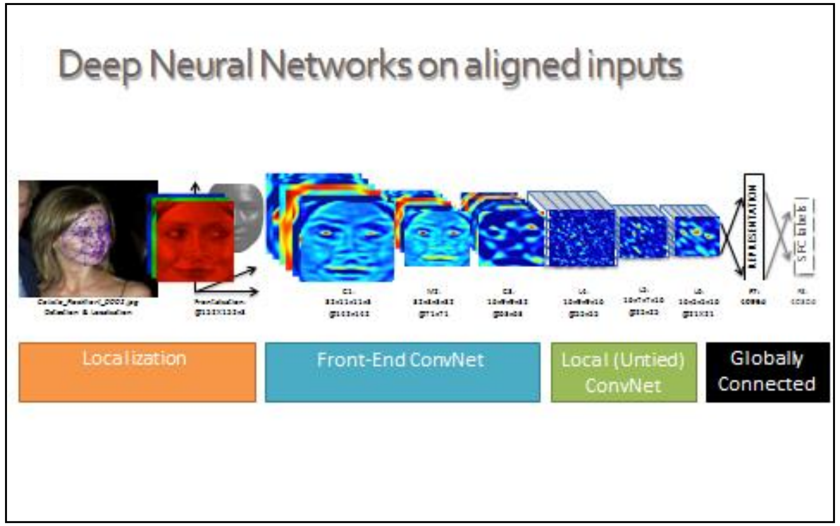
Table 5. Top 10 KITTI2015 Pure Optic Flow Algorithms as of the submission date. Fl-all is the percentage of pixels with euclidean error > 3 pixels. The FAST network was not trained on this benchmark by the submission time.

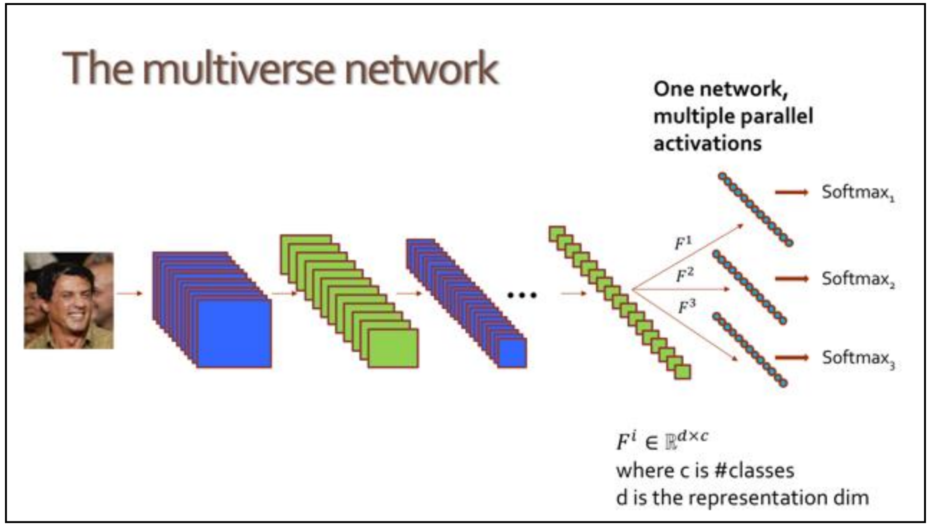
Benchmarks - MPI-Sintel

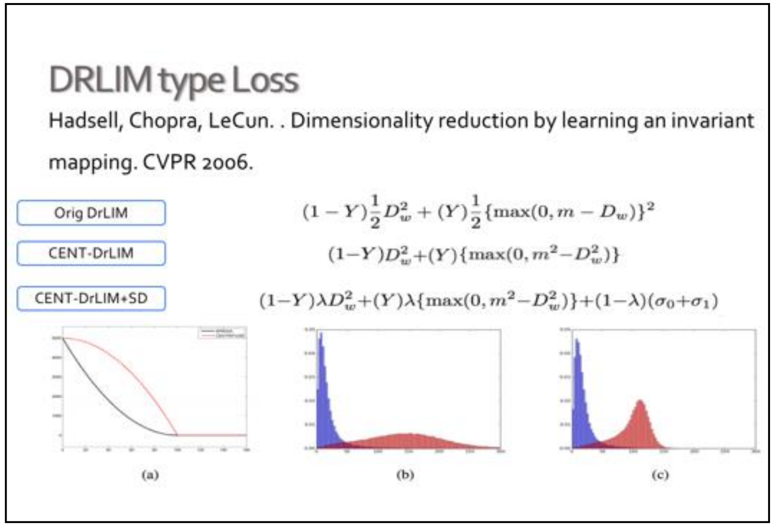
Method	EPE all, 'final' pass
FlowFields [1]	5.810
CPM-Flow (anon.)	5.960
DiscreteFlow [28]	6.077
EpicFlow [32]	6.285
Deep+R [13]	6.769
PatchBatch-CENT+SD	6.783
DeepFlow2 (anon.)	6.928
PatchBatch-SPRG	7.188
SparseFlowFused [36]	7.189
DeepFlow [38]	7.212
FlowNetS+ft+v [15]	7.218
NNF-Local [9]	7.249
PatchBatch-SPRG+SD	7.281
PatchBatch-CENT	7.323
SPM-BP [25]	7.325
AggregFlow [16]	7.329

Table 6. Top MPI-Sintel results as of the submission date. Each number represents the EPE (end-point-error), averaged over all the pixels in the comparison images, using the 'final' rendering pass of MPI-Sintel. Four ACCURATE variants are shown. The CENT-FIGURE+SD network is ranked 6th as of the paper's submission date. The FAST network was not trained on this benchmark by that date. The TF+OFM method [22] (EPE 6.727) is removed from this table since it is not a pure optical flow method.

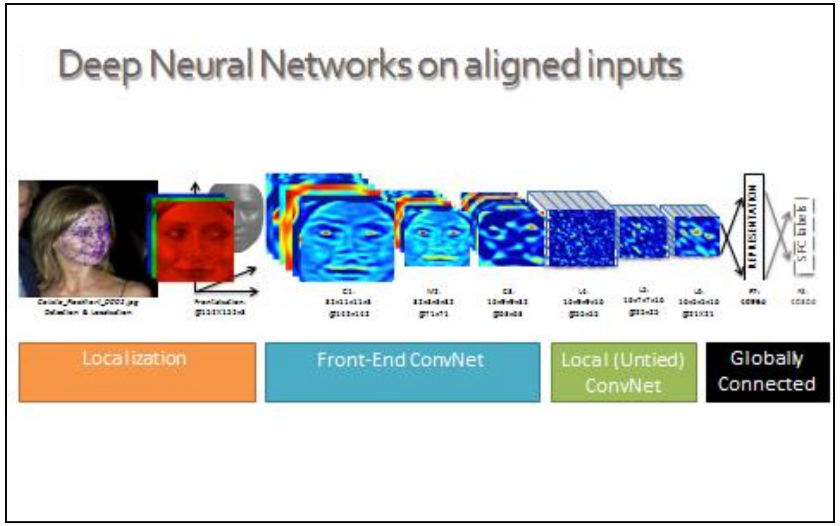
I'VE SPOKEN ENOUGH. ANY QUESTIONS?



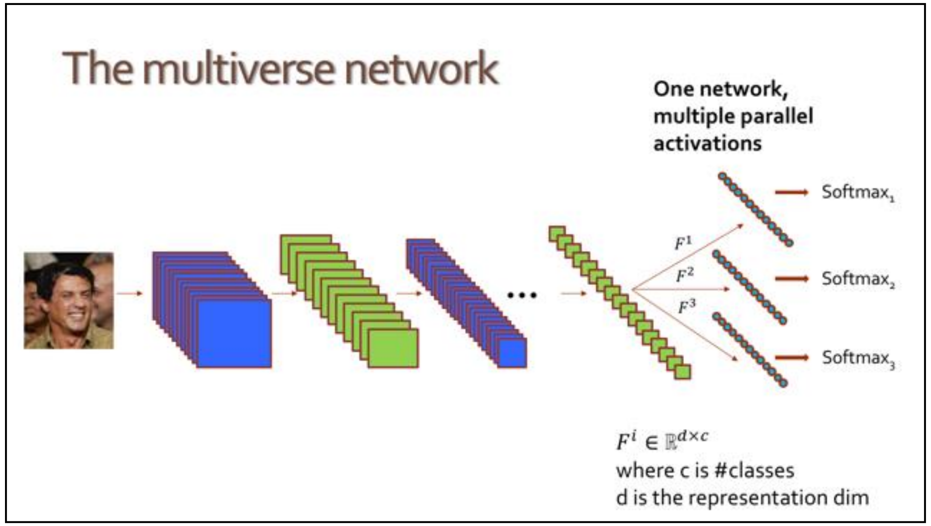




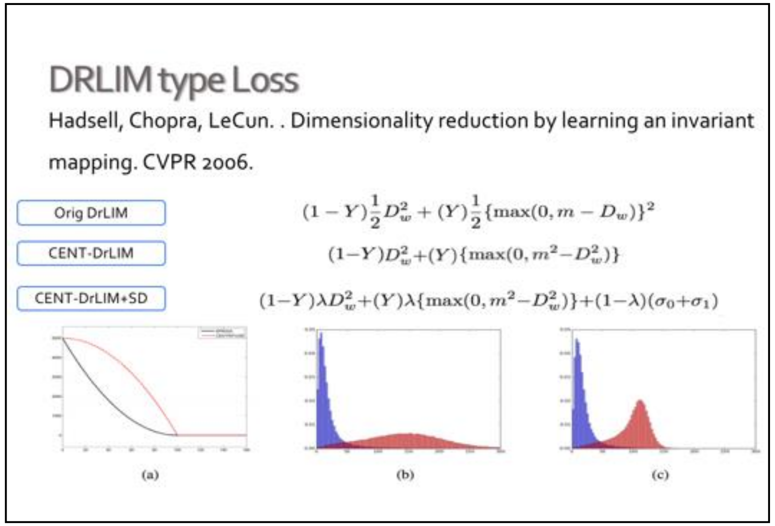
THANK YOU!



Deep Neural Networks on aligned inputs
Localization
Front-End ConvNet
Local (Untied) ConvNet
Globally Connected



The multiverse network
One network, multiple parallel activations
Softmax1
Softmax2
Softmax3
F1
F2
F3
Fi ∈ ℝd×c
where c is #classes
d is the representation dim



DRLIM type Loss
Hadsell, Chopra, LeCun. . Dimensionality reduction by learning an invariant mapping. CVPR 2006.
Orig DrLIM
CENT-DrLIM
CENT-DrLIM+SD
(a)
(b)
(c)