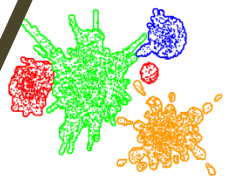
Recunoasterea Fetelor Umane

Interfata vizuala om-masina

Corneliu Florea

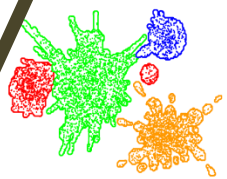


Cuprins

1. Problema recunoasterii

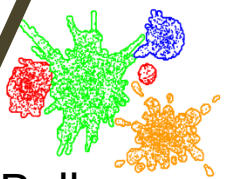
2. Normalizarea fetelor

3. Solutii :
   a. Recunoasterea fetelor pe baza LBP
      - Utilizarea LBP pentru recunoasterea fetelor
   b. Recunoasterea fetelor prin analiza imaginilor 3D
      - De ce 3D?
      - Recunoasterea in 3D
   c. Recunoasterea utilizand retele convolutionale adanci



Biometrica

- **Biometrica** – stiinta care se ocupa cu recunoasterea automata a indivizilor pe baza trasaturilor biologice si comportamentale
- **Sistem biometric** – un sistem de recunoastere care identifica o persoana prin autentificarea unor trasaturi bilogice si comportamentale specifice
- **Antropometrie** – Metode de masura ale corpului uman si partilor lui
- Oamenii se identifica prin:
  - Ceva ce au (documente de identitate)
  - Ceva ce stiu (parole)
  - Ceva ce sunt (cum arata)



[Belhumeur]

Biometrica

- Identificarea biometrica a oamenilor se poate face prin:

Retina – fiecare ochi are o retea unica de vase de sange





Amprente – masura judiciara arhicunoscuta

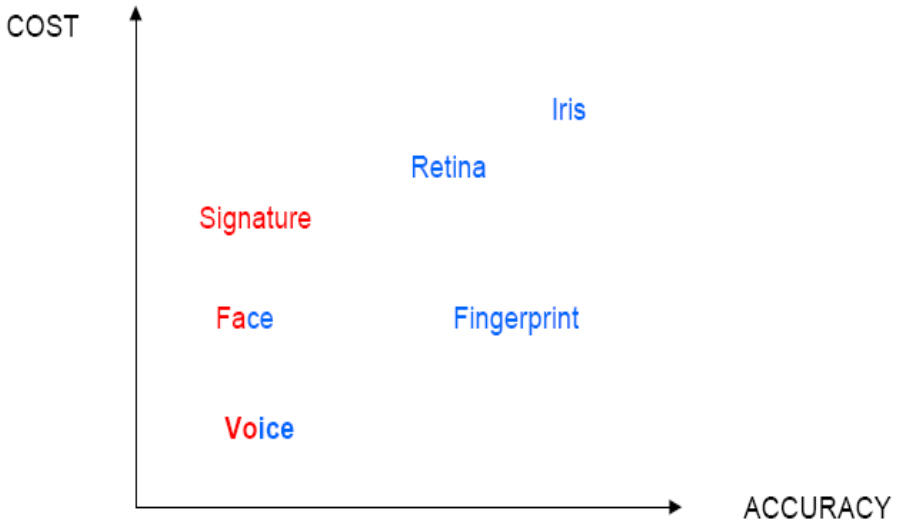
Fata

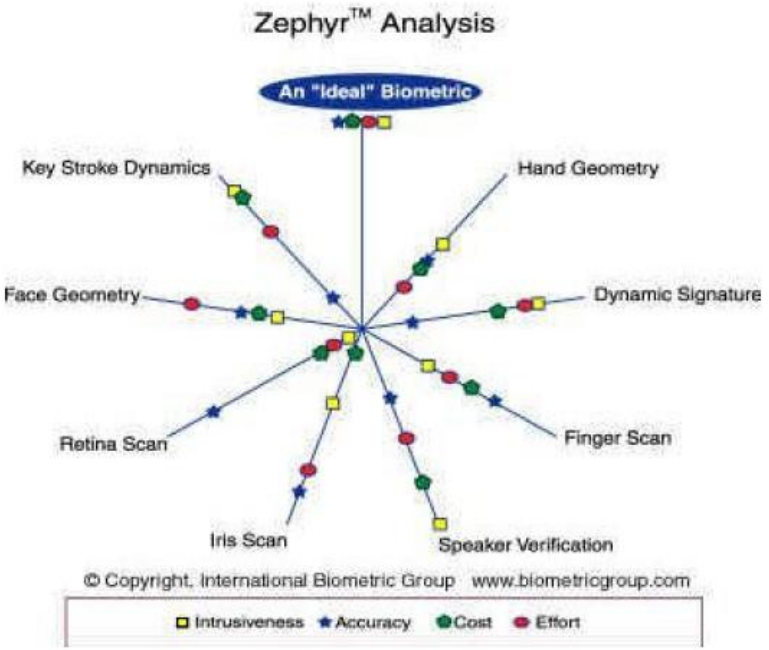


http://www.knowledgebase-script.com/demo/article-572.html

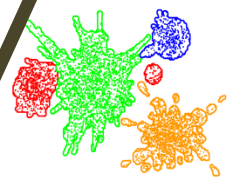
Comparatie intre tehnicile biometrice

Metode biometrice difera intre ele prin cost, acuratete, efort





[Belhumeur]



Recunoasterea fetelor

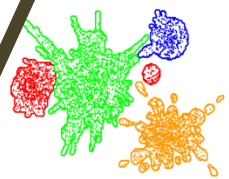
Doua aspecte:

1) Identificare

- Se raspunde la intreabarea “Cine este X?”
- Cautare “unul la multi”

2) Verificare

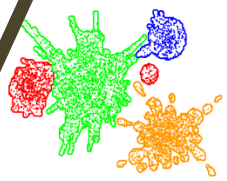
- Se raspunde la intrebarea “Este acesta X?”
- Cautare “unul la unu”



Recunoasterea fetelor

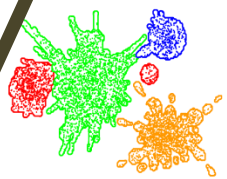
- Caracteristica biometrica cea mai utilizata in mod natural de catre oameni
- Tehnica non-intruziva care este acceptata de catre oameni ca o caracteristică biometrica
- Subiect de cercetare intensivă de peste 25 de ani

- Aplicatii:
  - Cautare de persoane in domeniul juridic sau pentru securitate
  - Autentificare
  - Personalizare automata



Recunoasterea fetelor de catre oameni

- Omul recunoaste fete comparand una noua cu una memorata si estimand gradul de similaritate
- Se pare ca recunoasterea fetelor este realizata de o parte specifica a creierului uman
- Procesul de recunoastere se bazeaza atat pe aspecte particulare cat si pe forma in ansamblu
- Memoria fetelor este dependenta de perspectiva
- Recunoasterea fetelor se face in paralel cu recunoasterea expresiilor
- Oameni recunosc mai bine persoane din propria rasa



Recunoasterea fetelor de catre oameni

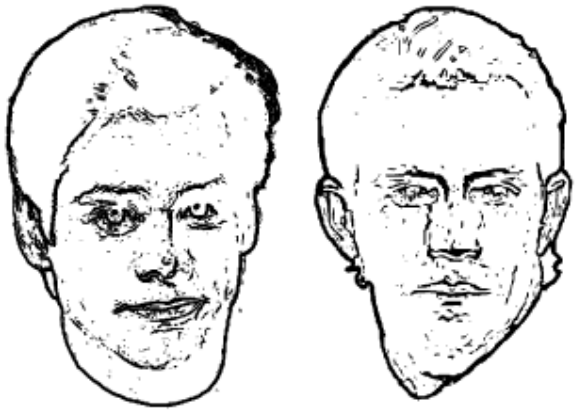
- [Sinha] studiaza intensiv recunoasterea fetelor de catre oameni si construieste exemple pentru a demonstra puterea acestui sistem



Putem recunoaste fete la rezolutie foarte mica

[Sinha]

Recunoasterea fetelor de catre oameni



Informatia de contur nu este foarte utila:

Jim Carey, Kevin Costner



Compresia (scalarea) brutala permite in continuare recunoasterea

[Sinha]

Recunoasterea fetelor de catre oameni

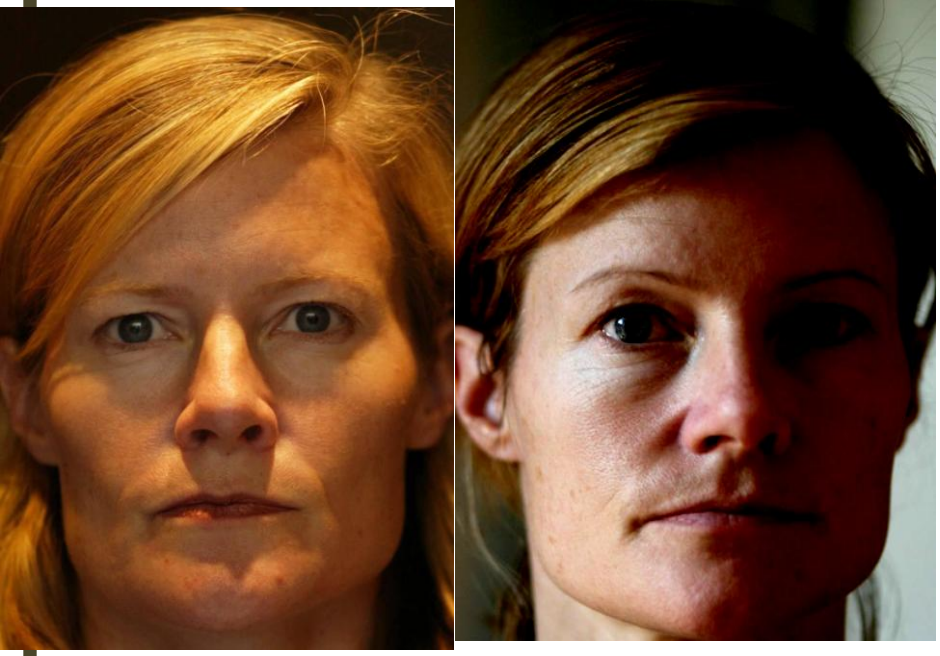


- [Sinha] foloseste acest exemplu pentru a ilustra dificultatea gasiri unei masuri potrivite de similaritate
- Cele doua imagini exterioare sunt ale aceleiasi persoane, in timp ce cea din centru este o persoana diferita. Dar cam orice masura de similaritate va spune altceva.

[Sinha]

Variatia iluminarii

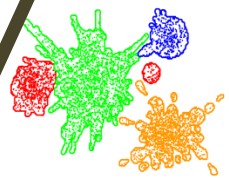
Interior





Exterior

[Belhumeur]



Variatia imaginilor

Termen scurt

Expresia



Postura



Iluminarea



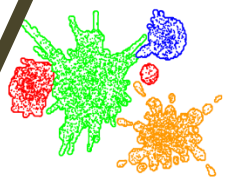
Termen lung

- Par facial
- Machiaj
- Ochelari
- Frizura
- Piercing
- Imbatranirea

[Belhumeur]

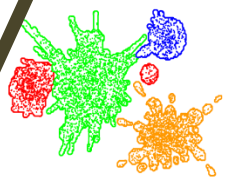
Necesitati ale recunoasterii

- Pentru a obtine rezultate bune in recunoasterea fetelor:
  - Achizitie foarte restrictiva – poze de cazier
  - Normalizarea fetelor – aducerea fetelor la un standard:
    - Iluminare – uniforma pe toata suprafata fetei
    - Pozitie - se rotesc, scaleaza si translateaza fetele astfel incat sa se potriveasca ochi si gura pe aceleasi pozitii
    - Expresie – neutra
  - Utilizarea de metode invariante la posibilitatile de variatie



Identificare vs. verificare

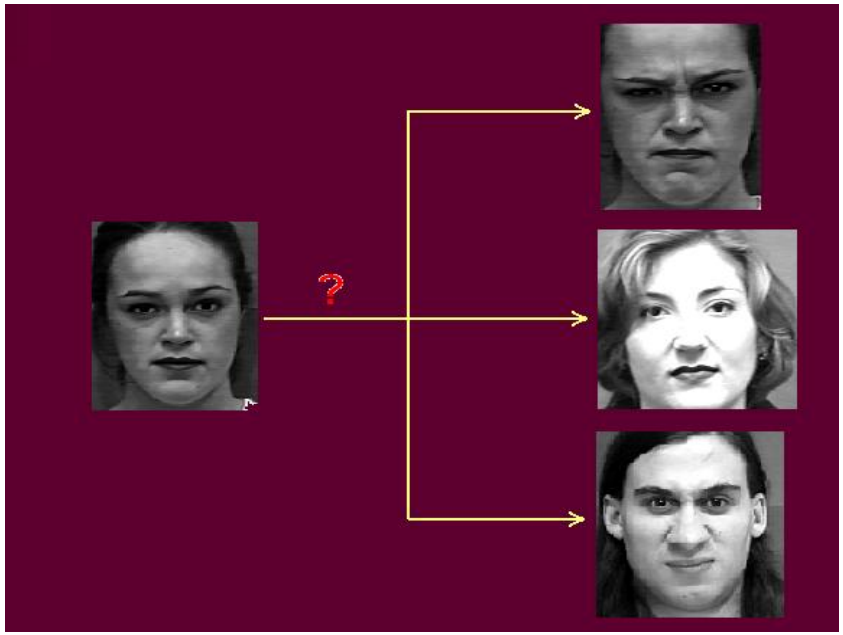
- Identificare:
  - Avand data o imagine de test si o galerie de imagini continand cel putin o imagine a persoanei din imaginea de test
  - Algoritmul ordoneaza toate pozele din galerie pe baza similaritatii cu imaginea de test
  - Masuri:
    - Curbe Cumulative Match Characteristics (CMC) : probabilitatea ca imaginea corecta din galerie sa apara in primele K
- Verificare:
  - Sunt date doua imagini
  - Algoritmul trebuie sa decida daca cele doua imagini contin aceeasi persoana sau nu
  - Masuri:
    - Calculeaza True Accept Rate (TAR) si False Accept Rate (FAR),
    - Se deseneaza curbe Receiver Operating Characteristic (ROC)



Identificare vs. verificare

Identificare: Se minimizeaza o masura de similaritate intre poza interogata sau o transformare a ei si reprezentantii fiecarei clase invatate

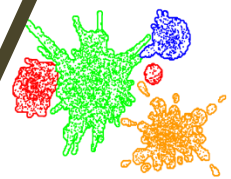
Verificare: Se calculeaza distanta pana la referinta si se compara cu un prag



Imagini din baza de date Cohn-Kanade

Ce masuri de similaritate stim?

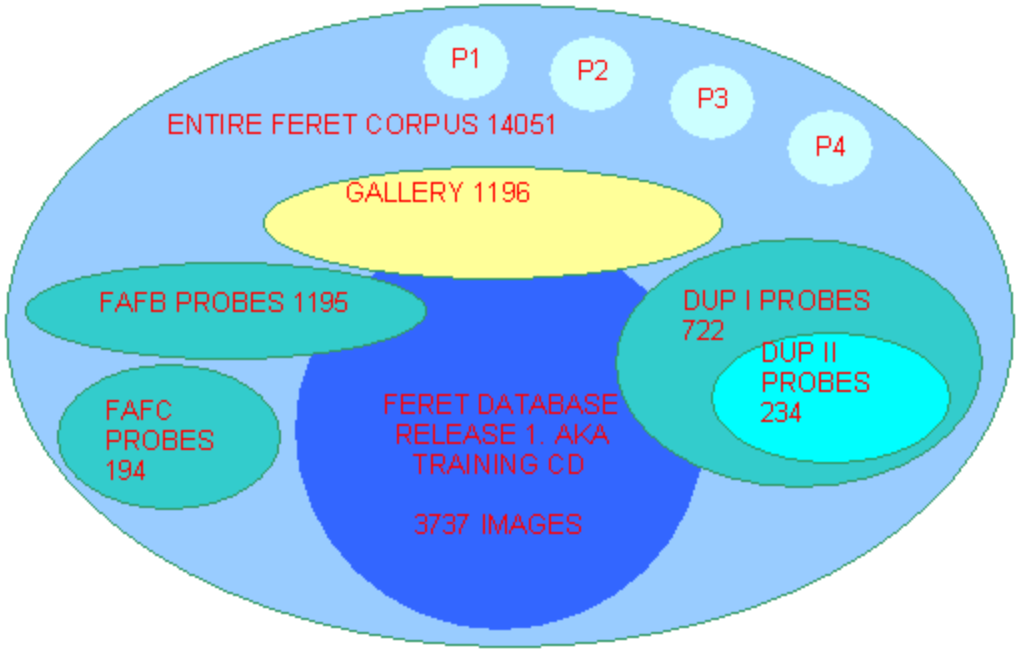
- Distante spatiale – norme
- Distante unghiulare
- Corelatie
- Distante intre functii
- ??



Evaluarea performantelor

FERET database:

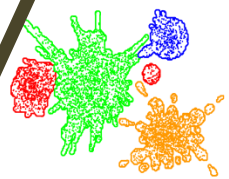
- Baza de date publica (la cerere) pentru evaluarea recunoasterii
- Contine imagini a 1196 de persoane cu pana la 5 poze per individ
- Competitie deschisa
- Rezultate publice pentru algoritmii cunoscuti



Normalizarea FERET

Pentru normalizare este necesara cunoasterea pozitiei ochilor

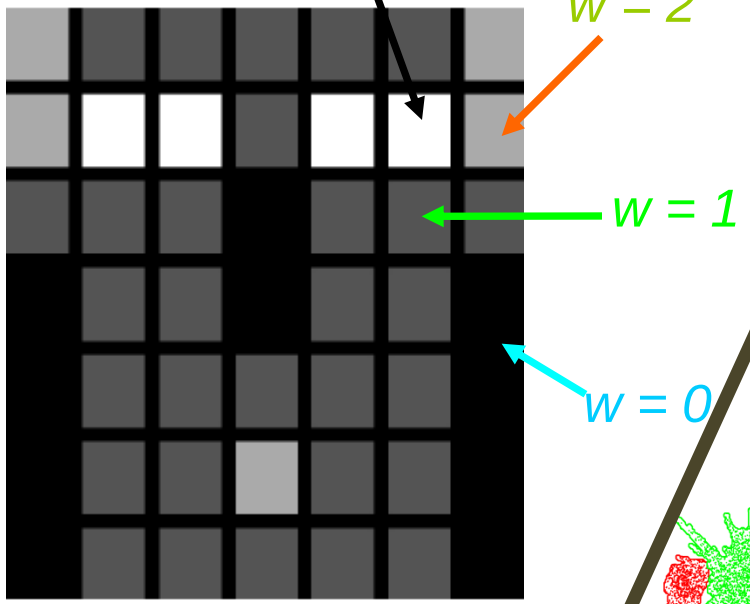
- Imaginea este rotita pentru a alinia ochii.
- Imaginea este scalata pentru a face distanţa dintre ochi constanta. Imaginea este trunchiată astfel incat sa contina doar fata.
- Se aplică o mască pentru a anula pixeli din afara unui oval ce contine o fata tipica. Ovalul este generat analitic.
- Se egalizeaza histograma pentru a netezi distribuţia valorilor de gri a pixelilor din oval
- Imaginea este normalizata astfel incat pixelii din oval sa aiba medie zero si deviatie standard 1.



Ponderarea blocurilor

- Pe fiecare bloc se calculeaza histograma LBP
- Blocurile sunt ponderate
  - Ochii si gura conteaza mai mult cand se face diferentierea intre persoane
  - Zonele uniforme cele mai putin semnificative





[Pietikäinen]

Similaritatea - divergenta KL

Cum verificam carei fete ii corespunde?

Trebuie verificata **similaritatea** intre doua histograme

Masuri de similaritate intre histograme :

- Divergenta Kullback – Leibler (entropia relativa) este:

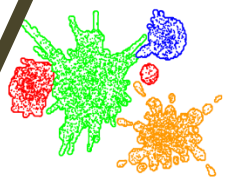
$$D_{KL}(P \parallel Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}.$$

Se arata ca :

$$D_{KL}(P||Q) \geq 0$$

Cu egalitate daca $P(x) = Q(x)$

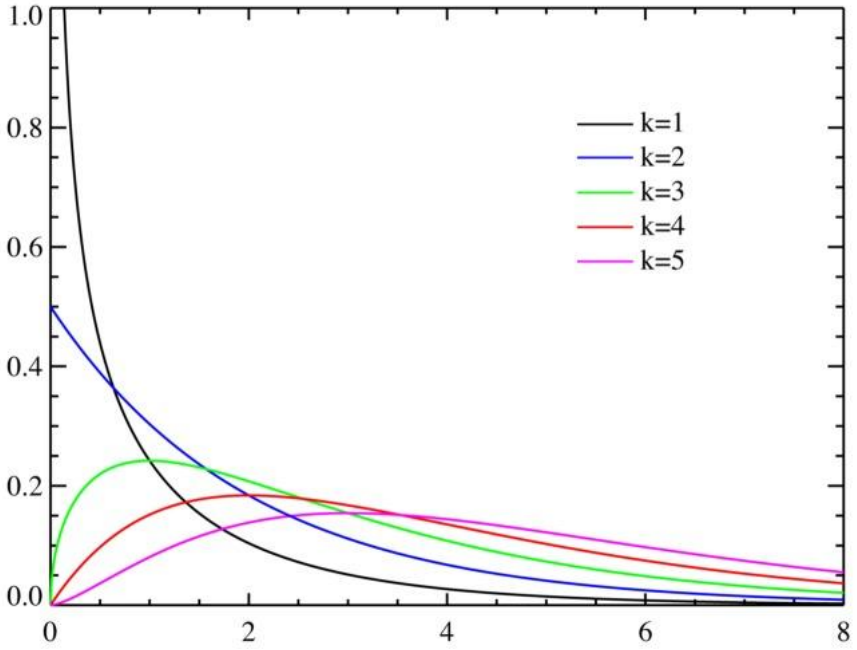
→ Ii va corespunde fetei pentru care divergenta este minima

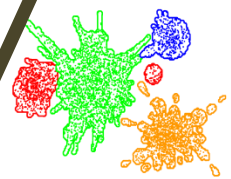


Similaritatea - legea χ^2

Masuri de similaritate intre histograme :

- Legea χ^2 (chi square)
  - In fapt χ^2 este un test daca un set de date este distribuit conform unei legi cunoscute
  - Legea χ^2 reprezinta distributia unei sume de patrate de v.a. gaussiene independente





Similaritatea - legea χ^2

Testul χ2 pentru verificarea unei distributii:

- Se partioneaza datele in *M* intervale
- Se determina probabilitatile fiecarui interval
- Se alege pragul testului: *α* cuantila superioara (la 0.05), notat x_α
- Se formeaza valoarea de test : $z = \sum_x \frac{(m_x - NP_x)^2}{NP_x}$.
- Daca $z > x_\alpha$, atunci datele sunt distribuite conform legii

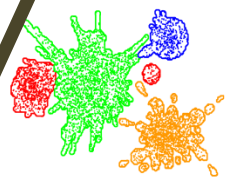
Pentru comparare de histograme se cauta valoarea maxima pentru:

$$\chi_w^2(x,\xi) = \sum_{i,j} w_j \frac{(x_{i,j} - \xi_{i,j})^2}{x_{i,j} + \xi_{i,j}}.$$

Histograma fetei de referinta

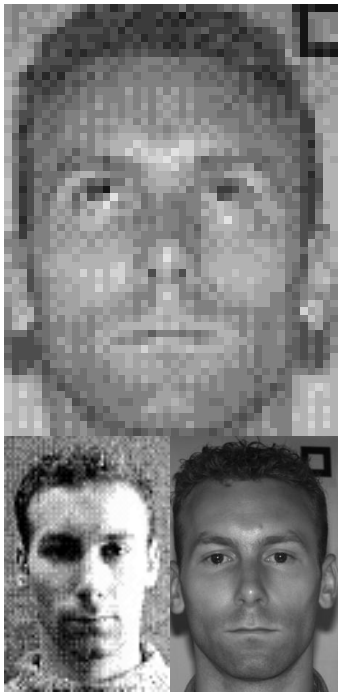
ponderi

Histograma curenta



Face 3D - Limitari pentru 2D

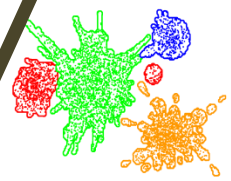
Eficienta sistemului este dependenta de conditiile de achizitie.



- **Conditiile de iluminare:**
  - Condiţii diferite de iluminare pentru înscriere respectiv interogare.
  - Pot provoaca saturaţia imaginii.
- **Orientarea capului:**
  - Distantele calculate pe caracteristici 2D se distorsioneaza.
- **Calitatea imaginii.**
  - CCTV, Web-cams, etc
- **Expresia faciala**.
  - Modificări în locaţia şi forma caracteristică.
- **Ocluzii Parţiale**
  - Pălării, eşarfe, ochelari etc



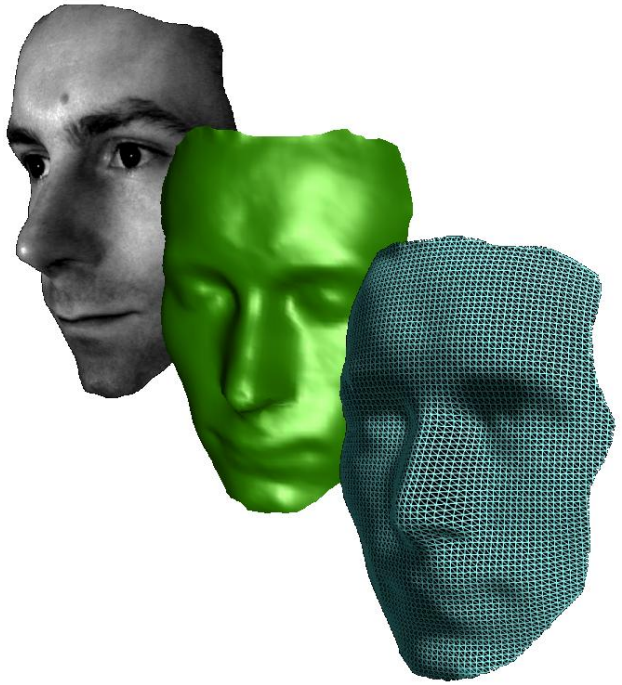
[Heseltine]

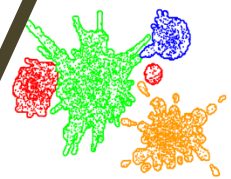


Recunoasterea fetelor 3D

Recunoasterea fetelor prin modelare 3D:

- Achizitie 3D a fetelor umane – noi modele de camere 3D sau prin stereografie
- Construirea unui model 3D si adaptarea lui in functie de imaginea 2D data
- Beneficii:
  - Se utilizeaza informatie geometrica de adancime in loc de culoare si textura
  - Este invarianta la variatia iluminarii
  - Permite rotatii – invarianta la postura
  - Permite scalare – invarianta la zoom

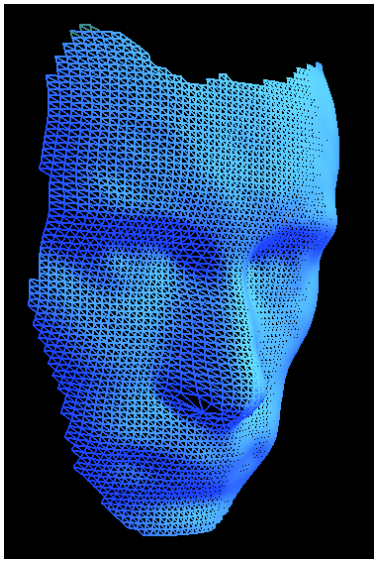




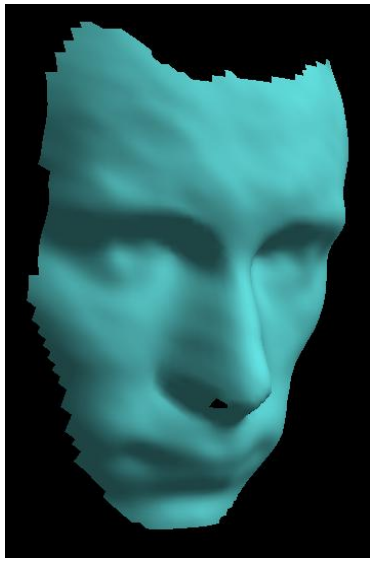
[Heseltine]

Portrete 3D

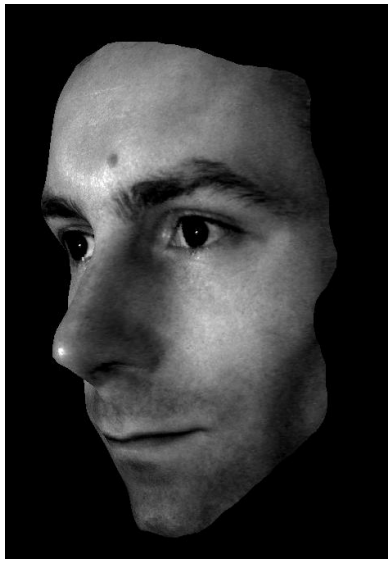
- Generate cu ajutorul unei camere stereo
- Suprafata fiecarei fete este reprezentata de ~ 8000 de puncte.
- Pe suprafata se mapeaza textura din imaginea cu tonuri de gri.



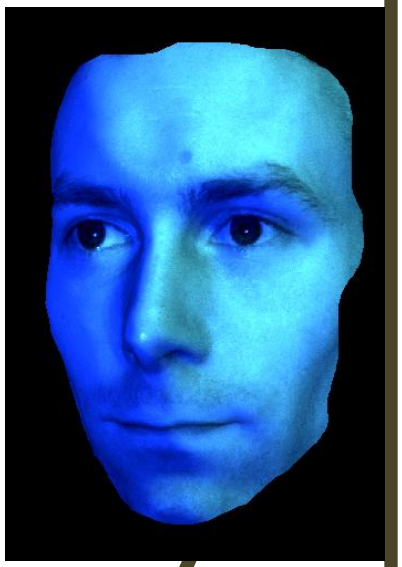
Reteaua de puncte



Poligoanele



Textura



Iluminarea

[Heseltine]

Baza de date

- Baza de date de la universitatea York
  - 5000 modele de fete a peste 350 de persoane.
  - Variatie mare in raport cu expresia orientarea, sexul, apartenenta etnica, varsta.
- Un subset (1770 modele) este folosit pentru antrenare si testare.
  - 300 modele 3D models a 50 de persone in antrenare
  - 1470 modele 3D a 280 de persoane pentru testare



3D face model captured using a stereo vision camera



Fete Fisher pentru 3D

Adaptare a fetelor Fisher (LDA) la date 3D:

- Modelele 3D sunt aliniate frontal
- Modelele 3D se convertesc in harti de adancime
  - Dimensiune 60 x 90 pixeli.
  - Se extrag caracteristici derivate
- Se utilizeaza LDA pentru reprezentare simplificata
- Hartile de adancime sunt comparate pe baza distantei euclidiene sau a distantei *cos*

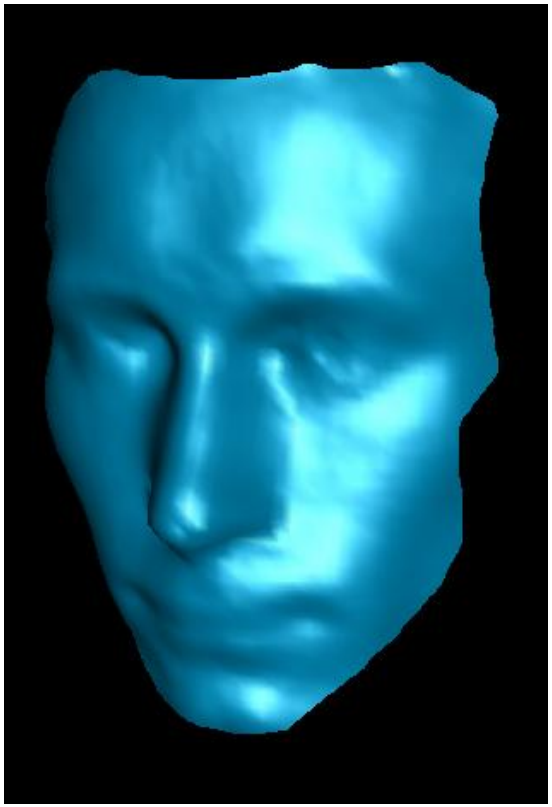
$$d_{Euclid} = \left\| \Omega_a - \Omega_b \right\| \qquad d_{\cos} = 1 - \frac{\langle \Omega_a, \Omega_b \rangle}{\left\| \Omega_a \right\| \cdot \left\| \Omega_a \right\|}$$



[Heseltine]

Caracteristici derivate pentru fete 3D

Pe langa harta de adancime se mai adauga 16 caracteristici ale fetei 3D

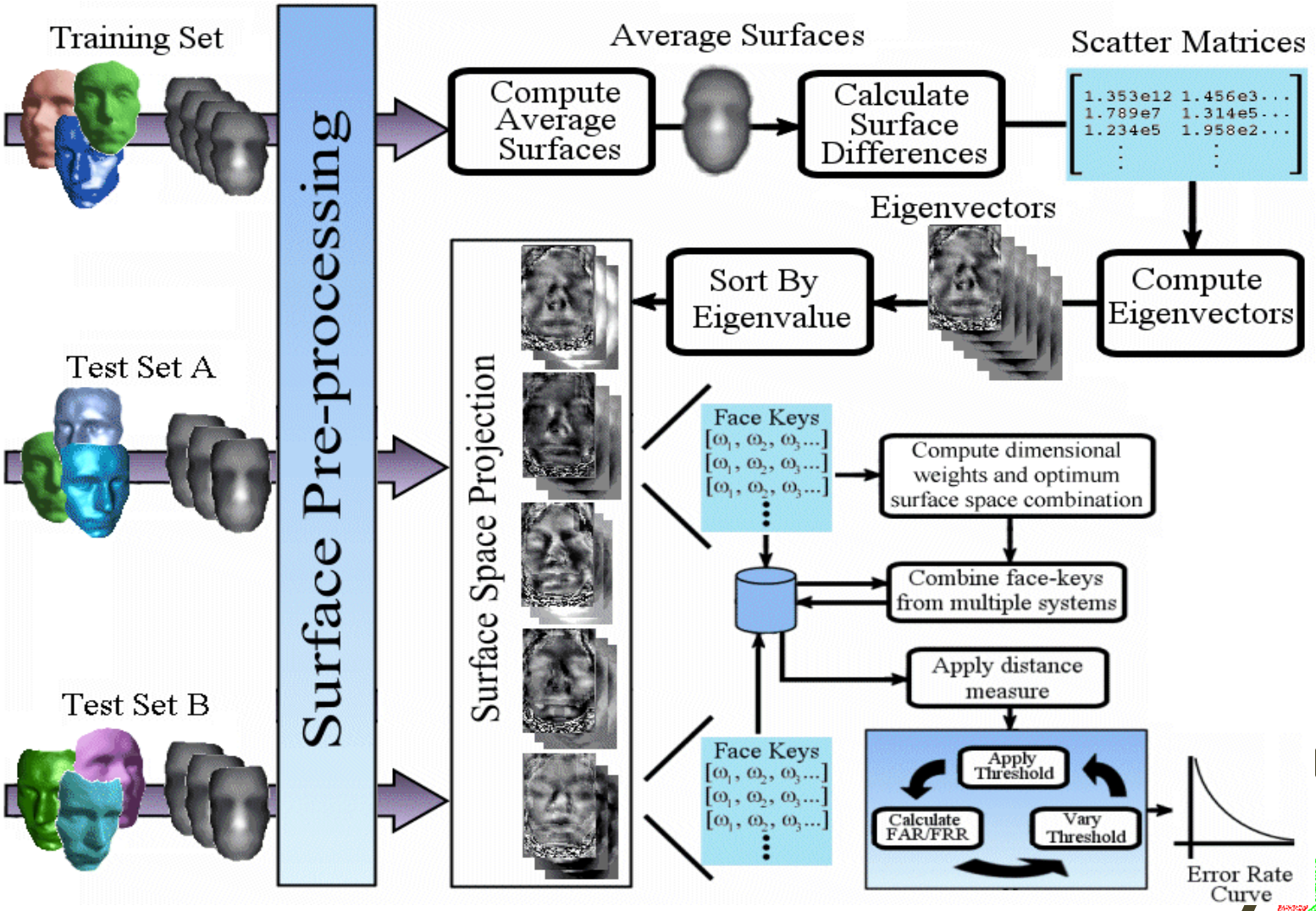




Horizontal Gradient	Vertical Gradient	Horiz. Gradient Large	Vert. Gradient Large
 -1 1	 -1 1	 -1 0 0 0 1	 -1 0 0 0 1
Applies the 2x1 kernel to compute the horizontal derivative	Applies the 1x2 kernel to compute the vertical derivative	Horizontal gradient over a greater horizontal distance	Vertical gradient over a greater vertical distance
Laplacian	**Sobel X**	**Sobel Y**	**Sobel Magnitude**
 0 1 0 1 −4 1 0 1 0	 -1 0 1 -2 0 2 -1 0 1	 1 2 1 0 0 0 -1 -2 -1	
An isotropic measure of the second spatial derivative	Application of the horizontal sobel derivative filter	Application of the vertical sobel derivative filter	The magnitude of Sobel X and Y combined.
Horizontal Curvature	**Vertical Curvature**	**Curvature Magnitude**	**Curve Type**
			
Applies sobel X twice to calculate the 2nd horizontal derivative	Applies sobel Y twice to calculate the 2nd vertical derivative	The magnitude of the vertical and horizontal curvatures	Segmentation of the surface into 8 discreet curvature types
Min Curvature	**Max Curvature**	**Abs Min Curvature**	**Abs Max Curvature**
			
The minimum of the horizontal & vertical curvature values	The maximum of horizontal & vertical curvature values	The minimum of absolute horizontal & vertical curvatures	The maximum of absolute horizontal & vertical curvatures

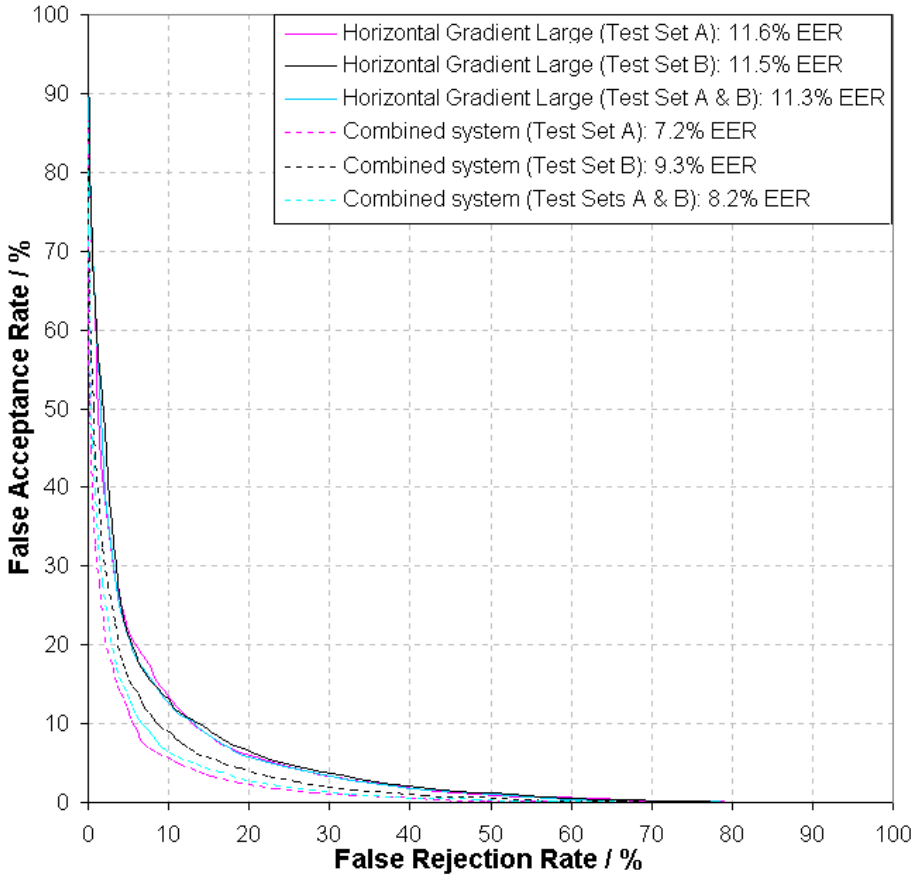
[Heseltine]

Procedura de test



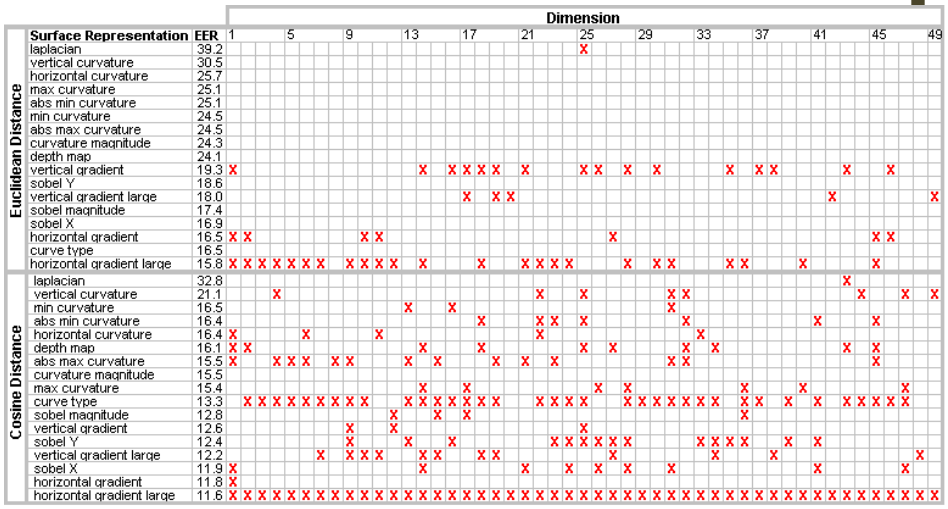
[Heseltine]

Rezultate finale



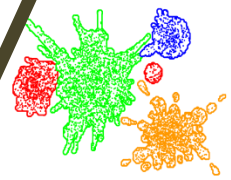
Dimensiunile relevante din reprezentarile unei singure trasaturi de suprafata sunt incorporate intr-o reprezentare compacta a fetelor 3D

Se foloseste metrica pe baza de distanta *cos*.

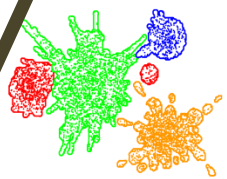


8.2% EER pe testul complet (11.3% single)

7.2% EER pe setul de test set redus (11.6% single)



RECUNOASTEREA FETELOR CU RETELE CONVOLUTIONALE ADANCI

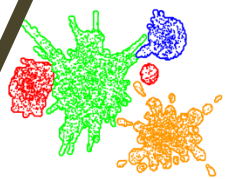


Milestone : DeepLearning

- Solutii bazate pe DeepLearning au permis aparitia de solutii care obtin performante mai bune decat omul in conditii standard
  - Conditii non-standard necesita solutii speciale – future
- Au aparut in paralel cu baze mari de date:

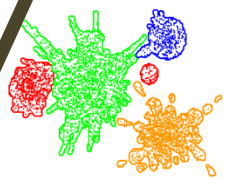
  -Exemple:

  - Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. CVPR Link - https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf
  - Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering, CVPR Link -https://arxiv.org/pdf/1503.03832v3.pdf
  - Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015, September). Deep face recognition. In *BMVC* (Vol. 1, No. 3, p. 6).Link: http://cis.csuohio.edu/~sschung/CIS660/DeepFaceRecognition_parkhi15.pdf



Deep face models

- De ce Deep?
- Ipoteza: Modelele de reprezentare faciala se află intr-un spatiu multi-dimensional complex, neliniar și non-convex.
- Abordarea curenta – Deep:
  - Obtinem baze de date mai mari : web-crawling – more faces!
  - Solutia celor de Google / Facebook etc.
  - Hardware mai puternic in sistemele embedded – putem lucra cu modele mai puternice

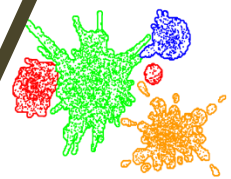


DeepFace

Wolf- 2014

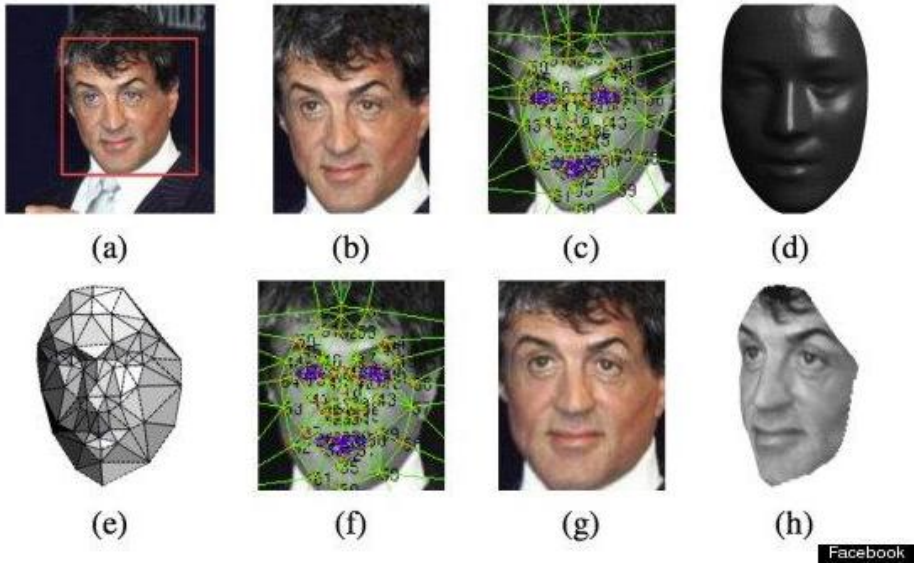


- Modelare 2D/3D si aliniere folosind transformari afine
- Retea convolutionala adanca cu 9 straturi si 120 milioane de parametri



DeepFace- Aliniere

- Se determina punctele cheie (landmark-uri)
- Se construiesc modele 2D/3D ale fetei
- Se aplica o transformare a afina asupra imaginii curent pentru a o aduce la o forma prestabilita (frontal)

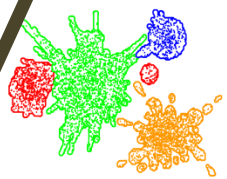


http://web.eng.tau.ac.il/deep_learn/wp-content/uploads/2016/11/Face-Recognition-A-Deep-Learning-Approach.pdf

DeepFace- DeepLearning

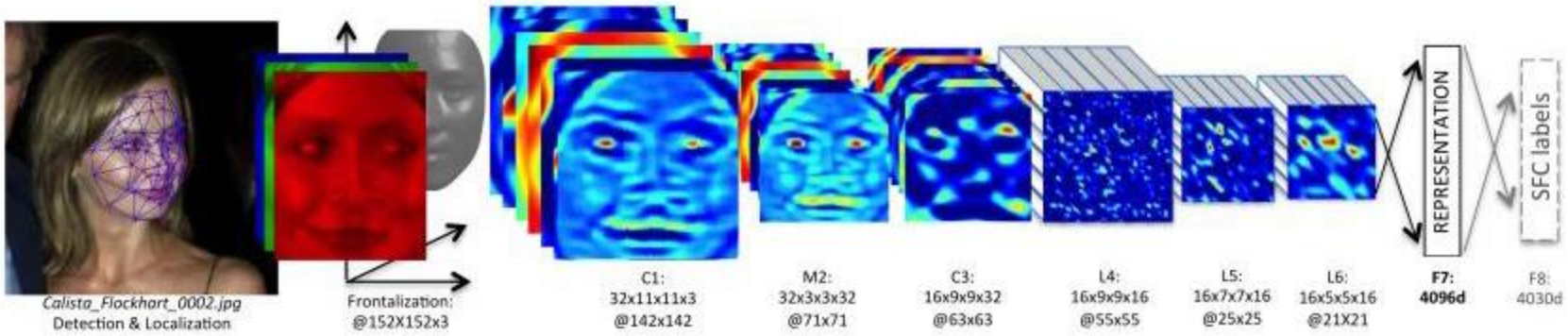
Reteaua

- Date de intrare: Imagini 3D (RGB) de 152x152 pixeli cu fate
  - pre-rulat face detector
- Retea convolutionala cu 9 straturi
- Include soft max pentru minimizarea “cross entropy”
- Antrenare cu SGD, regularizare cu Dropout
- Reducere dimensionalitatii cu ReLU
- Date de iesire: probabilitati pentru k classe



DeepFace -Deep Learning

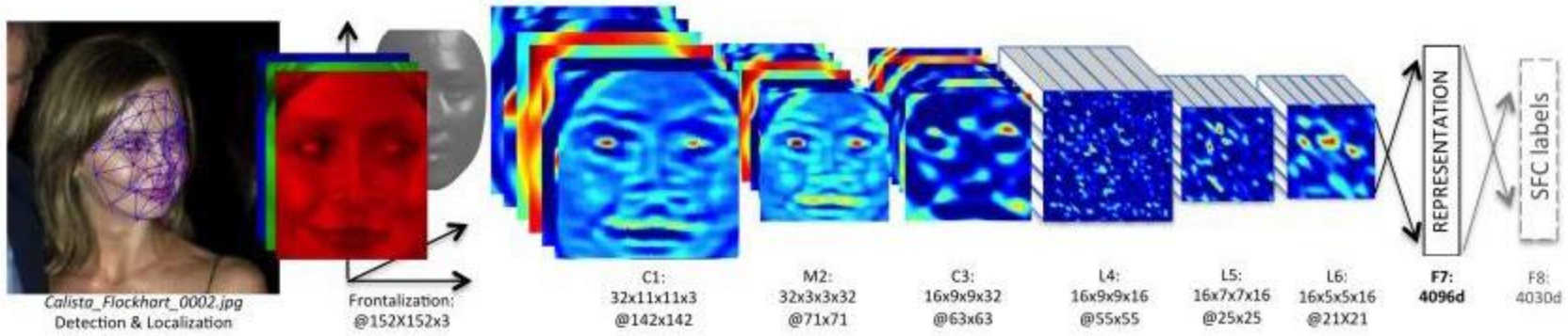
- Arhitectura



http://web.eng.tau.ac.il/deep_learn/wp-content/uploads/2016/11/Face-Recognition-A-Deep-Learning-Approach.pdf

DeepFace -Deep Learning

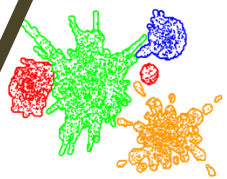
- Arhitectura - Detalii
  - Straturile 1-3 filtre simple pentru descriptori relevanti
  - Straturile 4-6 se concentreaza pe zonele relevante din imagine
  - Stratul 7 – conectat complet, genereaza un vector 4096 dimensional
    - Reprezentare (sparse) – rara: 75% din valori sunt zero
  - Stratul 8 conectatat complet – 4030 clase - Softmax



http://web.eng.tau.ac.il/deep_learn/wp-content/uploads/2016/11/Face-Recognition-A-Deep-Learning-Approach.pdf

Antrenare si performante

- Antrenata pe SFC cu 4M faces
  - 4030 identities, 800-1200 imagini/persona
- Evaluare pe Labeled Faces in the Wild (LFW)
- Antrenare cu SGD cu momentum de 0.9
- Learning rate 0.01 cu descrestere manuala pana la 0.0001
- Initializare aleatorie
- 15 epoci de antrenare
- 3 zile pe un GPU online



Antrenare si performante

- Acuratete 0.9735±0.0025

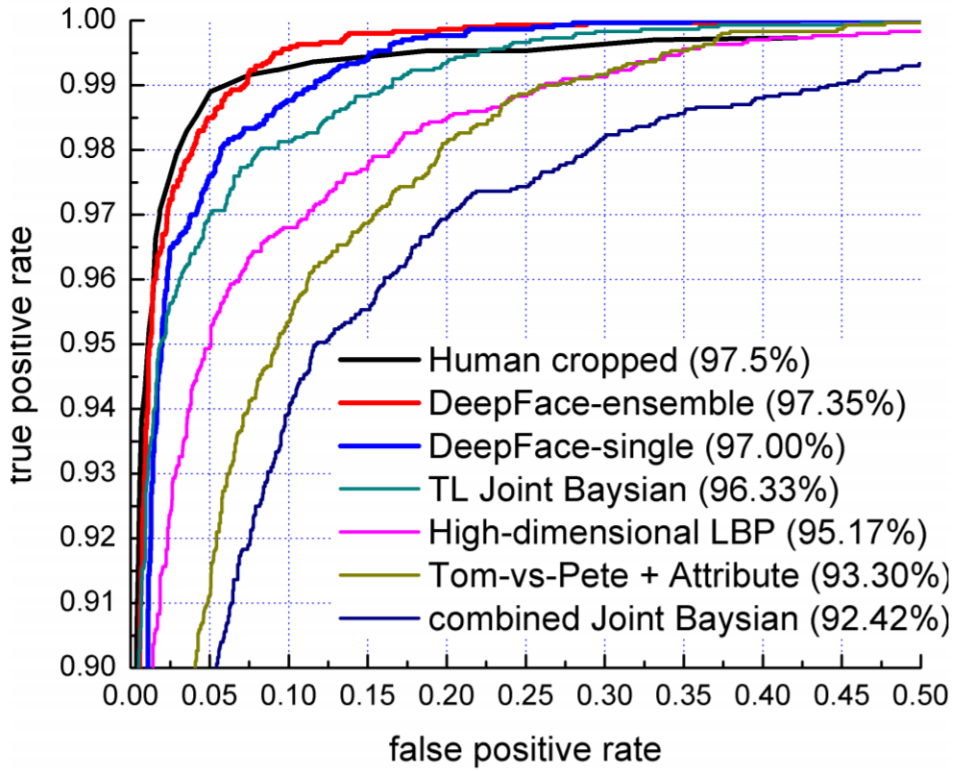
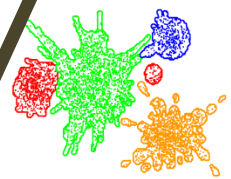
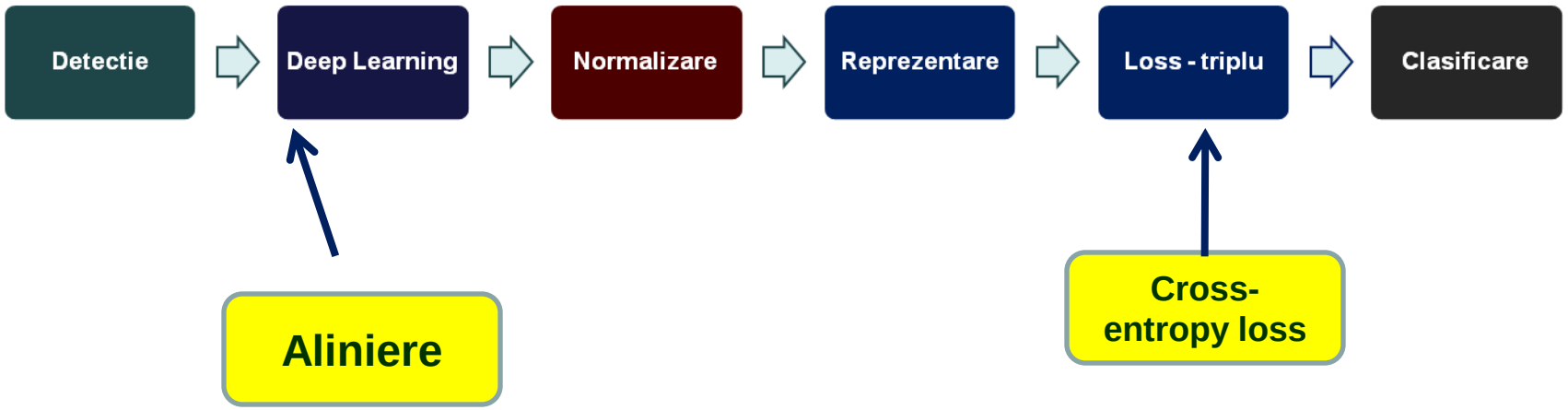


Figure 3. The ROC curves on the *LFW* dataset. Best viewed in color.



FaceNet

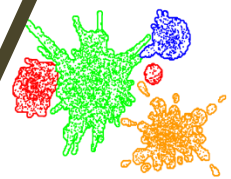
- Schroff and Philb in 2015



FaceNet

Arhitectura:

- Deep CNN : 22 straturi
- Antreneaza/testeaza pe datele originale
- Stratul final - embedding cu doar 128 dimensiuni/face –> f.f.eficient!
- Imagini triplet pt antrenare si functie obiectiv
- Foloseste SGD, Dropout, ReLU



FaceNet

- Optimizare dupa arhitectura retelei

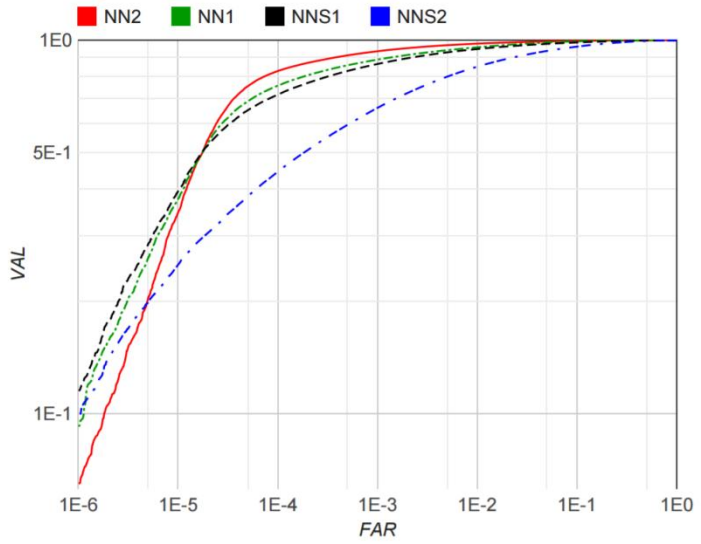


Figure 5. **Network Architectures.** This plot shows the complete ROC for the four different models on our personal photos test set from section 4.2. The sharp drop at 10E-4 FAR can be explained by noise in the groundtruth labels. The models in order of performance are: **NN2:** 224×224 input Inception based model; **NN1:** Zeiler&Fergus based network with 1×1 convolutions; **NNS1:** small Inception style model with only 220M FLOPS; **NNS2:** tiny Inception model with only 20M FLOPS.

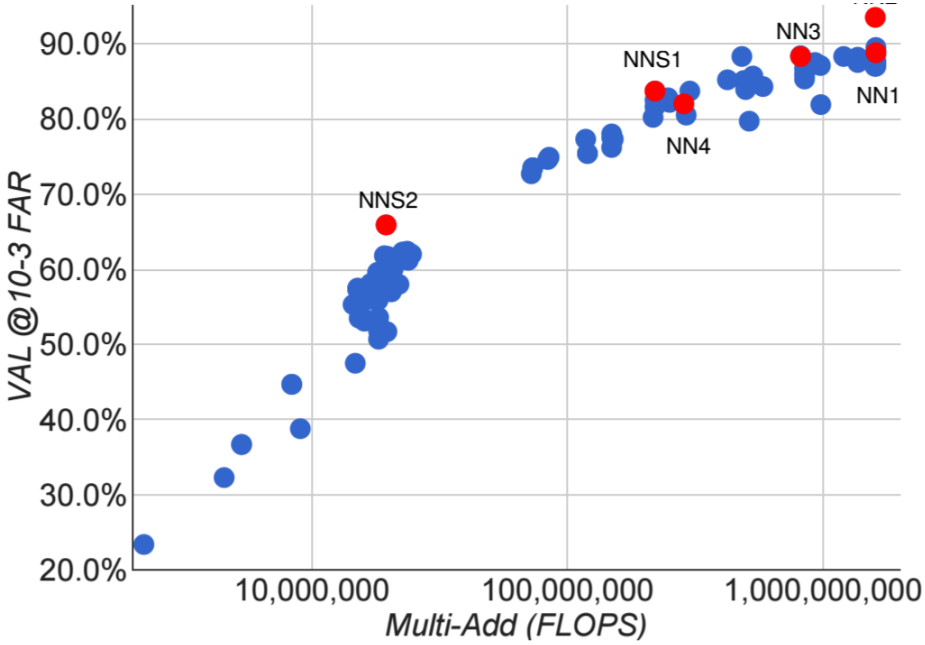


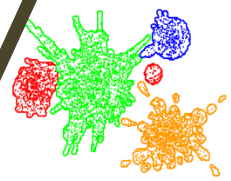
Figure 4. **FLOPS *vs*. Accuracy trade-off.** Shown is the trade-off between FLOPS and accuracy for a wide range of different model sizes and architectures. Highlighted are the four models that we focus on in our experiments.

FaceNet

Arhitectura:

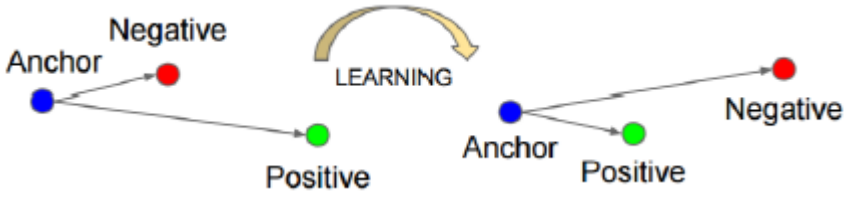
- 22 straturi:
  - 11 convolutii
  - 3 normalizari
  - 4 max-pooling
  - 1 concatenation
  - 3 fully-connected
- 140 million parameters

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

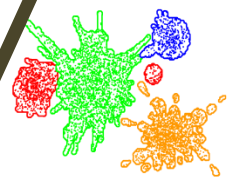


FaceNet

- Antrenare
  - Triplet
  - Model antrenat cu triplete aliniate grosier



$$L = \sum_{i=1}^{N} \left\| f\left(x_i^a\right) - f\left(x_i^p\right) \right\|_2^2 - \left\| f\left(x_i^a\right) - f\left(x_i^n\right) \right\|_2^2 + \alpha \qquad \forall \left(x_i^a, x_i^p, x_i^n\right) \in \mathrm{T}, f\left(x\right) \in \mathbb{R}^d$$



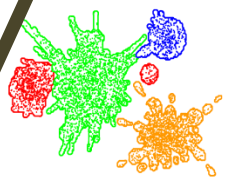
FaceNet- tripletul

- Imperativ pentru convergenta rapida
- Tripletul trebuie sa respecte conditia

$$\left\| f\left(x_i^a\right) - f\left(x_i^p\right) \right\|_2^2 + \alpha < \left\| f\left(x_i^a\right) - f\left(x_i^n\right) \right\|_2^2 \qquad \forall \left(x_i^a, x_i^p, x_i^n\right) \in \mathrm{T}$$

- Selectie online

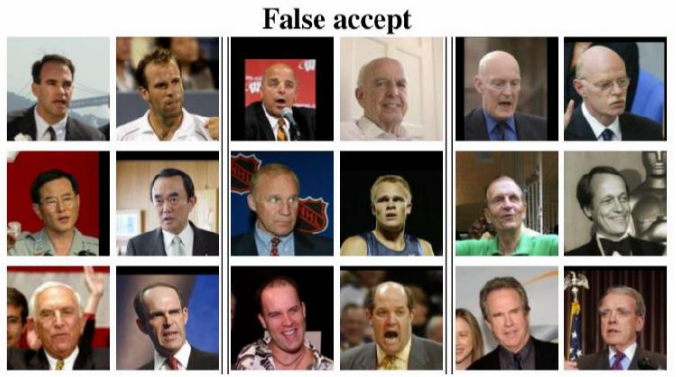
$$\arg\max_{x_i^p} \left\| f\left(x_i^a\right) - f\left(x_i^p\right) \right\|_2^2, \arg\min_{x_i^n} \left\| f\left(x_i^a\right) - f\left(x_i^n\right) \right\|_2^2$$



FaceNet-Evaluare

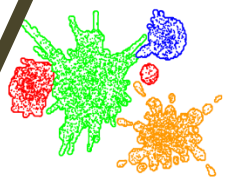
- Evaluare pe LFW
  - Acuratete **0.9963±0.09** cu aliniere





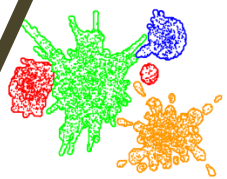
Deep Face Recognition

- Parkhi et al BMVC 2015
- Idei principale:
  - Constructia unei baze de date foarte mari cu resurse limitate
    - Introduc o baza de date cu 2.6 Million de fete
  - Recunoasterea fetelor competitiva cu Google si Facebook
    - Rezultate de top pe baza de date Youtube Faces
    - Performanta de top pe LFW



Deep FR – baza de date

1. Cautare pe Web dupa nume de persoane celebre
    1. Cautare pe IMDB, etc.
    2. 5000 identitati
2. Verificare manuala a identitatilor
    1. Selectat imagini relevante
    2. 200 img/ celebritate
    3. Eliminat cazurile slab reprezentate pe Google
    4. Eliminat suprapunerile cu bazele publice



DFR - baza de date

- Ex: Robert Downey Jr - popular



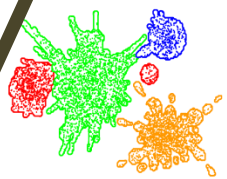
- Ex: Ashley Hamilton – mai putin popular



DFR – baza de data

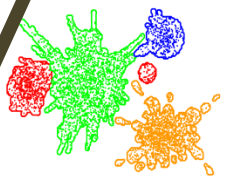
- Ordonarea exemplelor
  - Se aleg primele 1000 de exemple per identitate dupa o cautare a numelui +”actor” pe Google si Bing
- Rezulta o baza de date cu ~2.6 M fete
- Eliminarea duplicatelor
  - Folosesc VLAD descriptor pentru identificarea dublurilor
- Filtrare manuala
  - Curata setul de date prin verificari manuale.

- Procesul a durat 14 zile de munca manuala + timp de rulare in “retrieval”



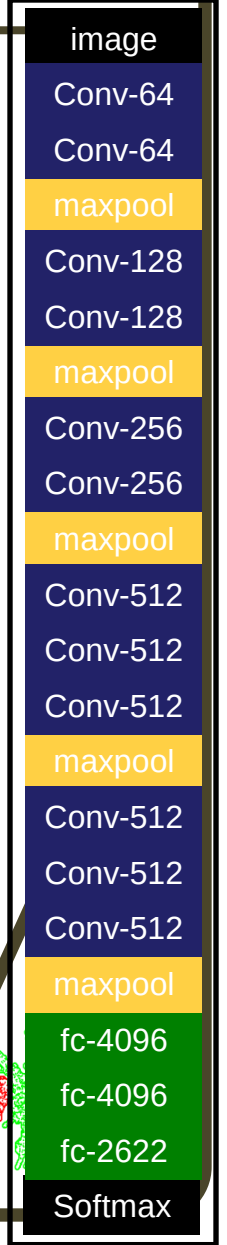
FaceReg – baze de date

No.	Aim	# Persons	Total # images
1	Labeled Faces In the Wild	5,749	13,233
2	WDRef	2995	99,773
3	Celeb Faces	10177	202,599
4	DeepFace Rec	2622	1,635,159
5	Facebook	4030	4.4M
6	Google	8M	200M



DFR - reteaua

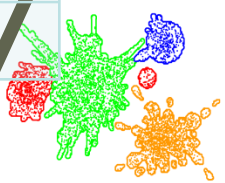
- Arhitectura este “VGG-Face:
  - Nuclee mici 3 x 3
  - Conv. Stride 1 px.
  - Relu non-linearity
  - No local contrast normalisation
  - 3 Fully connected
- Antrenare cu **triplete**
- Antrenare pe 2.6 M faces
  - Solutia foloseste embedding-urile finale



DFR - Rezultate

- LFW

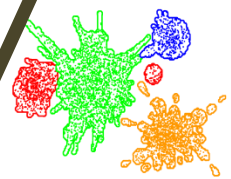
No.	Method	# Training Images	# Networks	Accuracy
1	Fisher Vector Faces	-	-	93.10
2	DeepFace	4 M	3	97.35
3	DeepFace Fusion	500 M	5	98.37
4	DeepID-2,3	Full	200	99.47
5	FaceNet	200 M	1	98.87
6	FaceNet+ Alignment	200 M	1	99.63
7	VGG Face	2.6 M	1	98.95



DFR - Rezultate

- YouTube - database

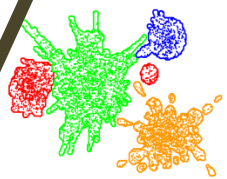
No.	Method	# Training Images	# Networks	100%-EER	Accuracy
1	Video Fisher Vector Faces	-	-	87.7	93.10
2	DeepFace	4 M	1	91.4	91.4
4	DeepID-2,2+,3		200	-	93.2
5	FaceNet + Alignment	200 M	1	-	95.1
7	**VGG Face**	**2.6 M**	**1**	**97.4**	**97.3**



Center Loss

- A new loss function (called center loss) to:
  - minimize the intra-class distances of the deep features.
- Joint supervision of the center loss and the softmax loss

  => highly discriminative features for robust face recognition.
- Easy to implement in the CNNs
- CNN models are trainable and can be directly optimized by the standard SGD.



[Wen2016]

Center Loss

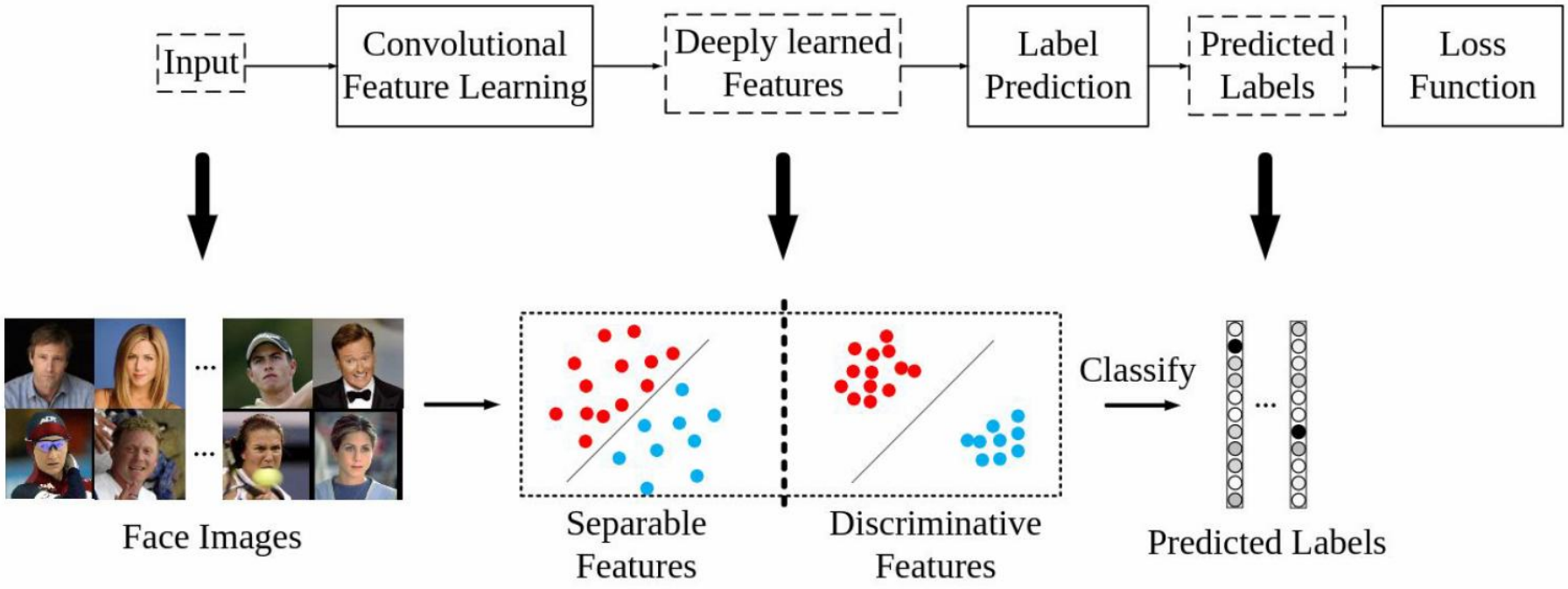
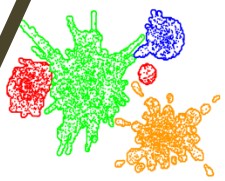


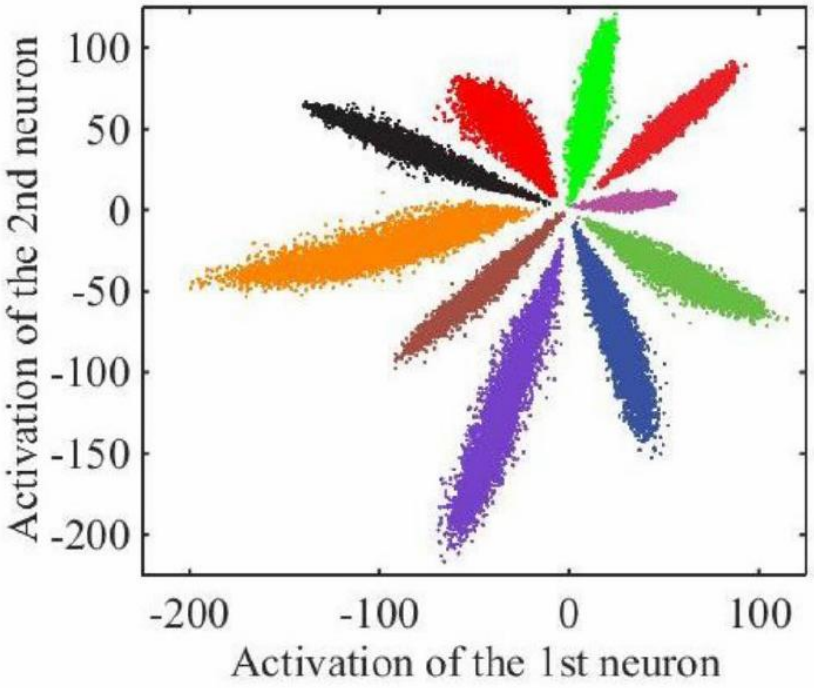
Fig. 1. The typical framework of convolutional neural networks.

Separable vs Discriminative

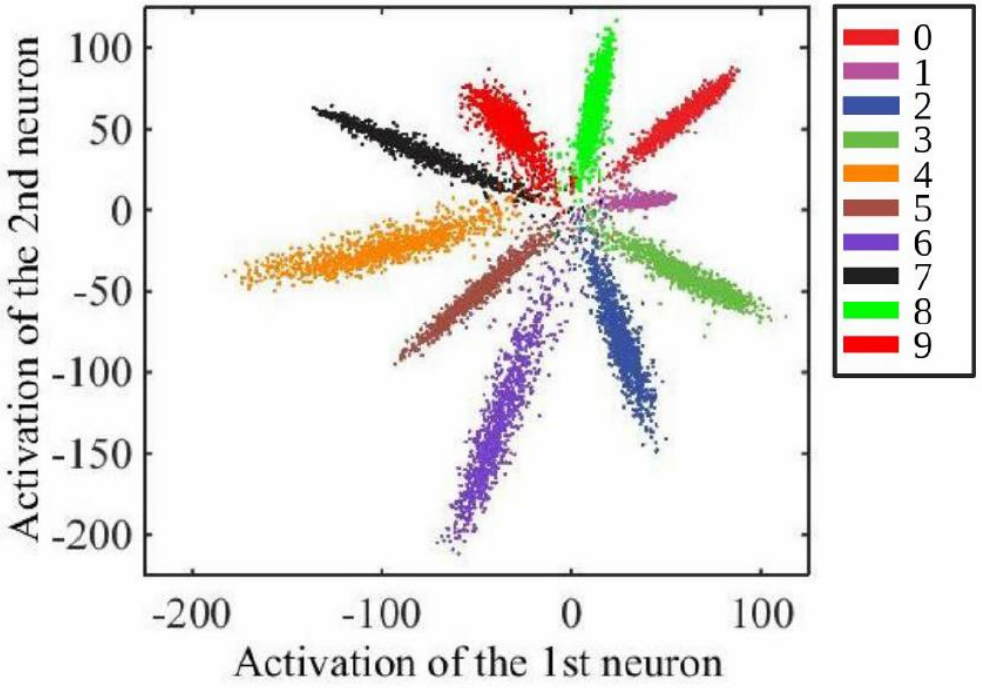


[Wen2016]

Is Softmax loss good for clustering?



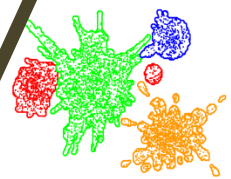
Training Set (50K)



Test Set (10K)

Separable, but not **discriminative** enough
(by intra-class variation)

[Wen2016]



Arhitecture

C: The convolution layer
P: The max-pooling layer
LC: The local convolution layer
FC: The fully connected layer

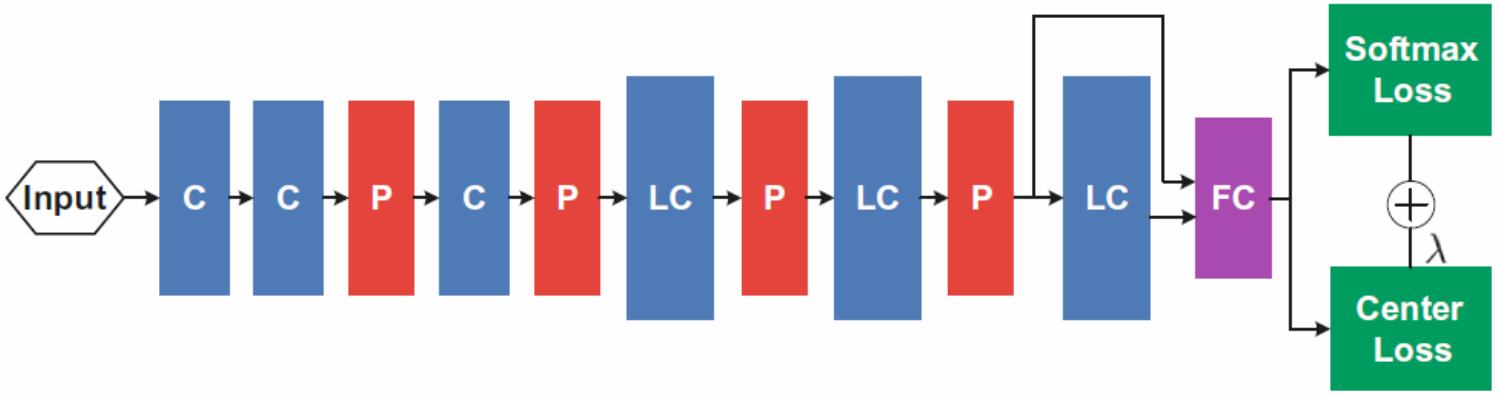
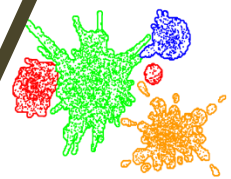


Fig. 4. The CNN architecture using for face recognition experiments. Joint supervision is adopted. The filter sizes in both convolution and local convolution layers are 3×3 with stride 1, followed by PReLU [12] nonlinear units. Weights in three local convolution layers are locally shared in the regions of 4×4, 2×2 and 1×1 respectively. The number of the feature maps are 128 for the convolution layers and 256 for the local convolution layers. The max-pooling grid is 2×2 and the stride is 2. The output of the 4th pooling layer and the 3th local convolution layer are concatenated as the input of the 1st fully connected layer. The output dimension of the fully connected layer is 512.

[Wen2016]

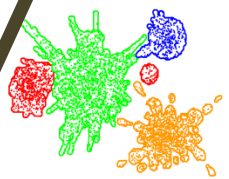


Softmax Loss

- m = mini-batch size
- n = the number of class
- x_i = feature vector in R^d (d is the feature dimension)
- $W = R^{dxn}$, $b = R^n$ (bias)

$$\mathcal{L}_S = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}}$$

[Wen2016]

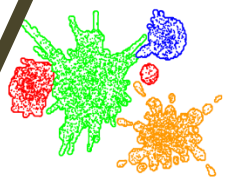


Let's be discriminative by Center Loss

- m = mini-batch size
- c_{yi} = y^{th} class center in d dimension
- x_i = feature vector in R^d (d is the feature dimension)
- But! c_{yi} should be updated as the deep features changed.
- Take the entire training set into account and average the features of every class in each iteration -> impractical

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

[Wen2016]



Let's be discriminative by Center Loss

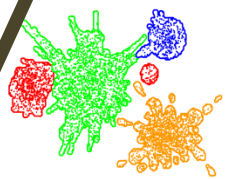
- Two modification
  a. Updating the centers based on mini-batch
  b. Updating center with learning rate alpha (for mislabeled samples)

$$\frac{\partial \mathcal{L}_C}{\partial \boldsymbol{x}_i} = \boldsymbol{x}_i - \boldsymbol{c}_{y_i}$$

$$\Delta \boldsymbol{c}_j = \frac{\sum_{i=1}^{m} \delta(y_i = j) \cdot (\boldsymbol{c}_j - \boldsymbol{x}_i)}{1 + \sum_{i=1}^{m} \delta(y_i = j)}$$

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C$$

[Wen2016]



Let's be discriminative by Center Loss

Algorithm 1 The discriminative feature learning algorithm

Input: Training data $\{\boldsymbol{x}_i\}$. Initialized parameters θ_C in convolution layers. Parameters W and $\{\boldsymbol{c}_j|j = 1, 2, ..., n\}$ in loss layers, respectively. Hyperparameter λ, α and learning rate μ^t. The number of iteration $t \leftarrow 0$.

Output: The parameters θ_C.

1: **while** not converge **do**

2: $t \leftarrow t + 1$.

3: Compute the joint loss by $\mathcal{L}^t = \mathcal{L}_S^t + \mathcal{L}_C^t$.

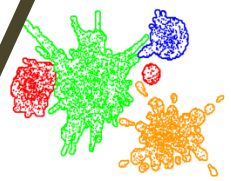
4: Compute the backpropagation error $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t}$ for each i by $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t} = \frac{\partial \mathcal{L}_S^t}{\partial \boldsymbol{x}_i^t} + \lambda \cdot \frac{\partial \mathcal{L}_C^t}{\partial \boldsymbol{x}_i^t}$.

5: Update the parameters W by $W^{t+1} = W^t - \mu^t \cdot \frac{\partial \mathcal{L}^t}{\partial W^t} = W^t - \mu^t \cdot \frac{\partial \mathcal{L}_S^t}{\partial W^t}$.

6: Update the parameters $\boldsymbol{c}_j$ for each j by $\boldsymbol{c}_j^{t+1} = \boldsymbol{c}_j^t - \alpha \cdot \Delta \boldsymbol{c}_j^t$.

7: Update the parameters θ_C by $\theta_C^{t+1} = \theta_C^t - \mu^t \sum_i^m \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{x}_i^t} \cdot \frac{\partial \boldsymbol{x}_i^t}{\partial \theta_C^t}$.

8: **end while**



[Wen2016]

Varying λ with Loss = Soft + λ*Center

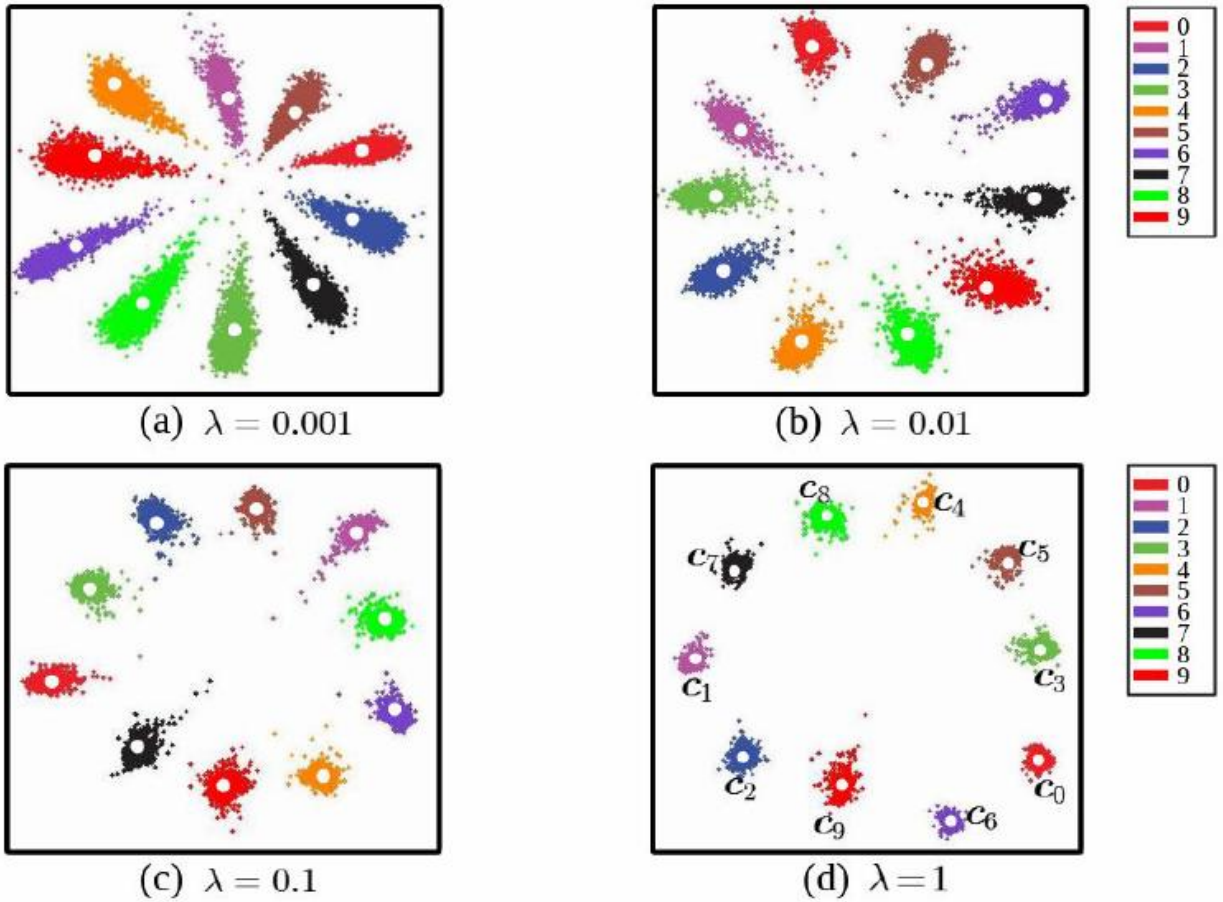
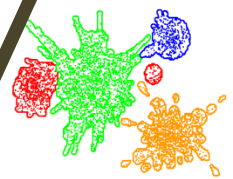


Fig. 3. The distribution of deeply learned features under the joint supervision of softmax loss and center loss. The points with different colors denote features from different classes. Different λ lead to different deep feature distributions ($\alpha = 0.5$). The white dots ($\boldsymbol{c}_0$, $\boldsymbol{c}_1$,...,$\boldsymbol{c}_9$) denote 10 class centers of deep features. **Best viewed in color.**

[Wen2016]

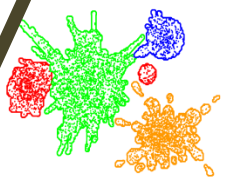


Implementation Details

Training data

- 17,189 unique persons
- 0.7 M images
  a. CASIA-WebFace : 0.49 M, 10,575 persons
  b. CACD2000 : 0.16 M, 2,000 persons
  c. Celebrity+ : 0.20 M, 10,177 persons
- Remove same identities with the testing set!!!
- Data Augmentation : Only horizontally Flipped

[Wen2016]



Implementation Details

Preprocessing

1. Cropped Face Size = (112x96x3)
   a. Subtraction 127.5
   b. Dividing by 128
   c. Every Pixel Values in -1 <= val <= 1
2. Face Detect by MTCNN
3. Use 5 landmarks
   a. Two Eyes
   b. One Nose
   c. Two Mouth Corners
4. Alignment : Similarity Transformation.
   a. Rotation, Translation, Scaling



$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & -b & c \\ b & a & d \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

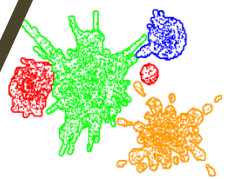
[Wen2016]

Implementation Details

Detailed Settings in CNNs

- Model A : Softmax Loss
- Model B : Softmax Loss & Contrastive Loss
- Model C : Softmax Loss & **Center Loss**
- Batch Size : 256
- GPUs : 2 x Titan X
- Learning rate : Start from 0.1
- Model A, C
  a. Learning rate divided by 10 at the 16K, 24K iterations
  b. Complete at 28K iterations (Roughly costs **14 hours**)
- Model B
  a. Learning rate divided by 10 at the 24K, 36K iterations
  b. Complete at 42K iterations (Roughly costs **22 hours**)

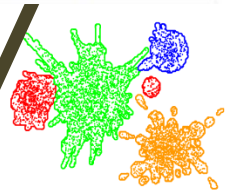
[Wen2016]



Verification Performance (λ=0.003 & α=0.5)

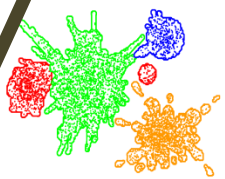
Method	Images	Networks	Acc. on LFW	Acc. on YTF
DeepFace [34]	4M	3	97.35%	91.4%
DeepID-2+ [32]	-	1	98.70%	-
DeepID-2+ [32]	-	25	99.47%	93.2%
FaceNet [27]	200M	1	99.63%	95.1%
Deep FR [25]	2.6M	1	98.95%	97.3%
Baidu [21]	1.3M	1	99.13%	-
model A	0.7M	**1**	97.37%	91.1%
model B	0.7M	**1**	99.10%	93.8%
model C (Proposed)	**0.7M**	**1**	**99.28%**	**94.9%**

[Wen2016]



MegaFace

- Gallery Set : 690,000 persons / 1 milion Images (Distractors)
- Probe set
  a. Facescrub : 530 persons / 100K images
  b. FGNet : 82 persons / 1,002 images (ages varying from 0 to 69)
- Small / Large Protocol
  a. When training db size < 0.5M, < 20K persons
  b. When training db size > 0.5M, > 20K persons
- Face Identification
  a. From 1 vs 10
  b. To 1 vs 1,000,000
- Face Verification
  a. 4 bilion = 4,000,000,000 negative pairs



[Wen2016]

MegaFace : Identification (1M and 10K distractors)

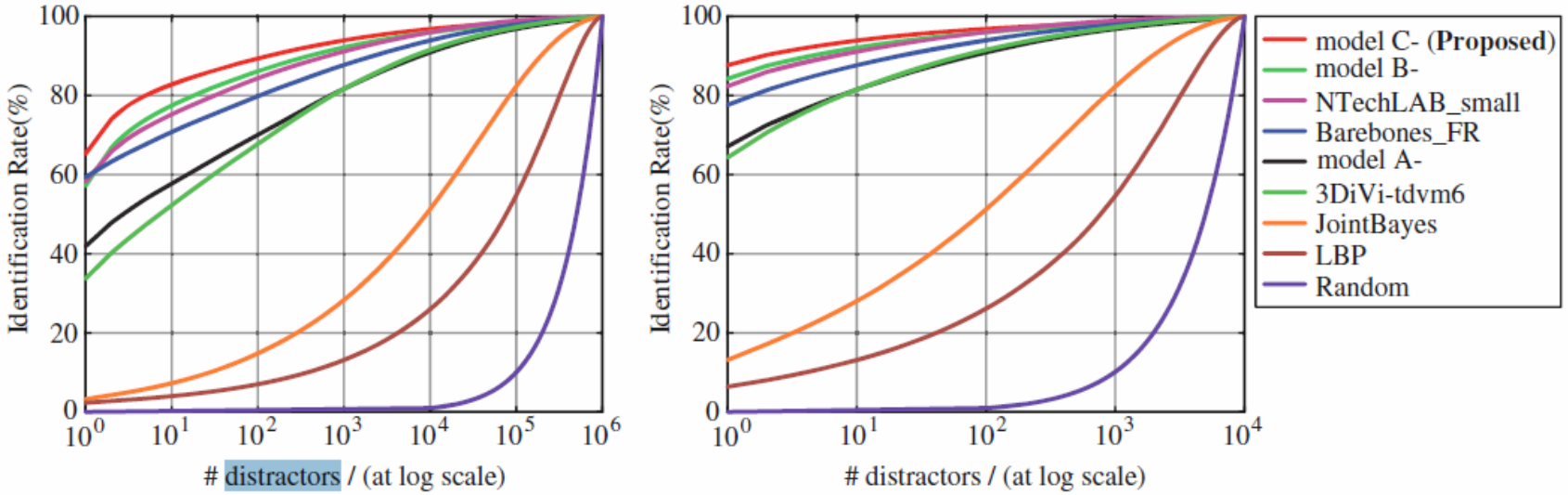
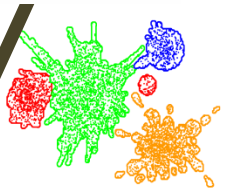


Fig. 8. CMC curves of different methods (under the protocol of small training set) with (a) 1M and (b) 10 K distractors on Set 1. The results of other methods are provided by MegaFace team.

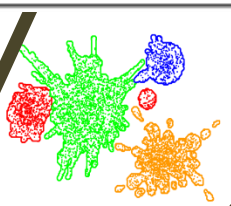
[Wen2016]



MegaFace : Identification (1M distractors)

Method	protocol	Identification Acc. (Set 1)
NTechLAB - facenx_large	large	73.300%
Google - FaceNet v8	large	70.496%
Beijing Faceall Co. - FaceAll_Norm_1600	large	64.803%
Beijing Faceall Co. - FaceAll_1600	large	63.977%
Barebones_FR - cnn	small	59.363%
NTechLAB - facenx_small	small	58.218%
3DiVi Company - tdvm6	small	33.705%
model A-	small	41.863%
model B-	small	57.175%
model C- (Proposed)	**small**	**65.234%**

[Wen2016]



MegaFace : Verification

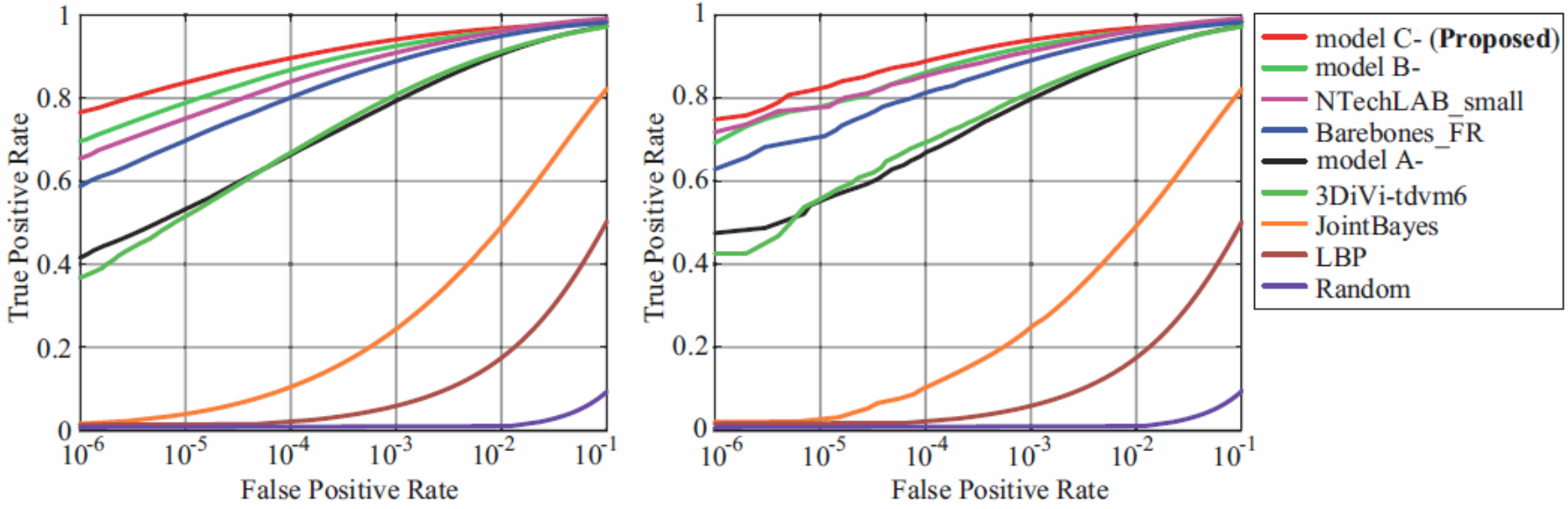
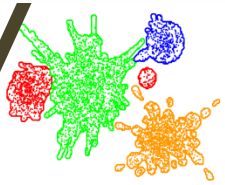


Fig. 9. ROC curves of different methods (under the protocol of small training set) with (a) 1M and (b) 10 K distractors on Set 1. The results of other methods are provided by MegaFace team.

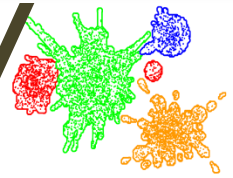
[Wen2016]



MegaFace : Verification (10^{-6} FAR, 1M distractors)

Method	protocol	Verification Acc. (Set 1)
Google - FaceNet v8	large	86.473%
NTechLAB - facenx_large	large	85.081%
Beijing Faceall Co. - FaceAll_Norm_1600	large	67.118%
Beijing Faceall Co. - FaceAll_1600	large	63.960%
Barebones_FR - cnn	small	59.036%
NTechLAB - facenx_small	small	66.366%
3DiVi Company - tdvm6	small	36.927%
model A-	small	41.297%
model B-	small	69.987%
model C- (Proposed)	**small**	**76.516%**

[Wen2016]



Center Loss

- center loss reduce the intra-class variations of the learned features, but the inter-class similarity was not considered.
- propose an island loss to simultaneously compress each cluster and push cluster centers apart as isolated “islands”
- Used for *expression recognition*

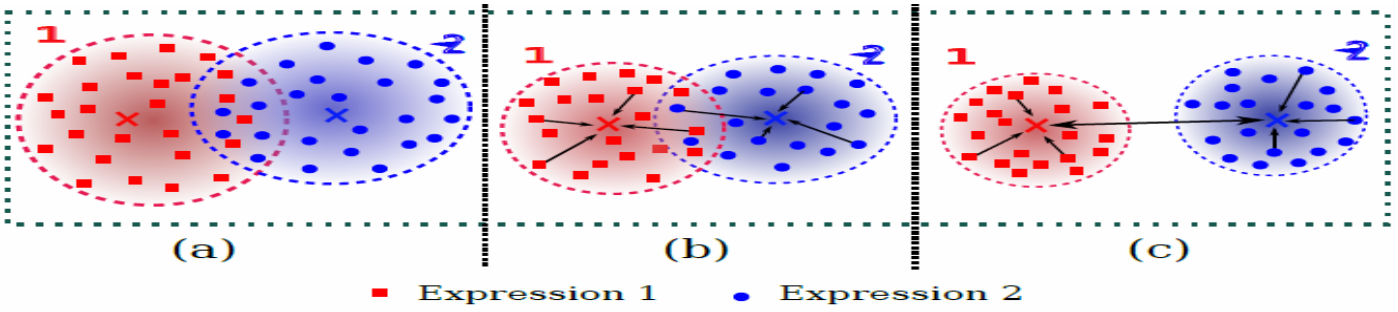
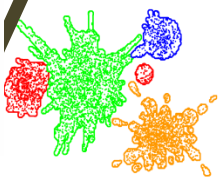


Fig. 1. An illustration of deep features learned by (a) a softmax loss, (b) a softmax loss + a center loss, and (c) a softmax loss + an island loss in the feature space. Inclusion of a center loss pulls the features of the same expression towards their centers denoted by a cross; while the island loss not only compresses the clusters individually, but also pushes the two clusters apart. Best viewed in color.

[Cai2018]



Arhitecture

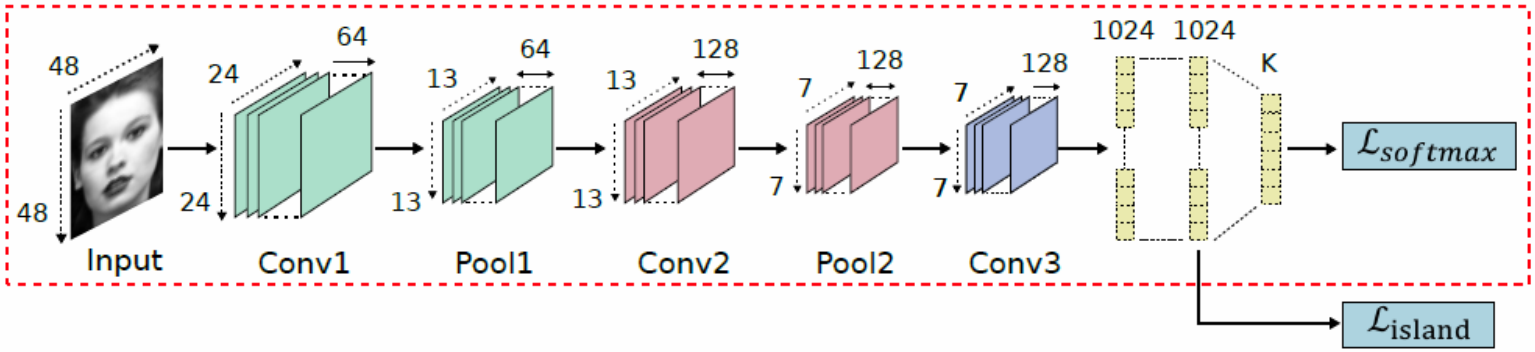
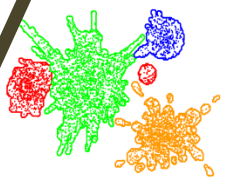


Fig. 2. The proposed IL-CNN for facial expression recognition. An island loss calculated at the second fully-connected layer and the softmax loss calculated at the decision layer are responsible to fine-tuning the CNN parameters. Best viewed in color.



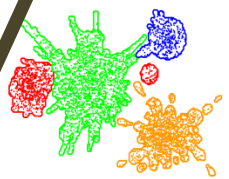
[Cai2018]

Island Loss

$$\mathcal{L}_{IL} = \mathcal{L}_C + \lambda_1 \sum_{\mathbf{c}_j \in \mathcal{N}} \sum_{\substack{\mathbf{c}_k \in \mathcal{N} \\ \mathbf{c}_k \neq \mathbf{c}_j}} \left(\frac{\mathbf{c}_k \cdot \mathbf{c}_j}{\|\mathbf{c}_k\|_2 \|\mathbf{c}_j\|_2} + 1 \right) \quad (6)$$

where $\mathcal{N}$ is the set of expression labels; $\mathbf{c}_k$ and $\mathbf{c}_j$ denote the k^{th} and j^{th} center with L_2 norm $\|\mathbf{c}_k\|_2$ and $\|\mathbf{c}_j\|_2$, respectively; $(\cdot)$ represents the dot product. Specifically, the

- the first term penalizes the distance between the sample and its corresponding center
- the second term penalizes the similarity between expressions.
- the samples of the same expression will get closer to each other

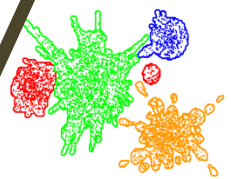


[Cai2018]

Update Cluster Centers

$$\Delta \mathbf{c}_j = \frac{\sum_{i=1}^{m} \delta(y_i, j)(\mathbf{c}_j - \mathbf{x}_i)}{1 + \sum_{i=1}^{m} \delta(y_i, j)} + \frac{\lambda_1}{|\mathcal{N}| - 1} \sum_{\substack{\mathbf{c}_k \in \mathcal{N} \\ \mathbf{c}_k \neq \mathbf{c}_j}} \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|_2 \|\mathbf{c}_j\|_2} - \left(\frac{\mathbf{c}_k \cdot \mathbf{c}_j}{\|\mathbf{c}_k\|_2 \|\mathbf{c}_j\|_2^3} \right) \mathbf{c}_j \tag{9}$$

where $|\mathcal{N}|$ is the total number of expressions.



[Cai2018]

Visualization study

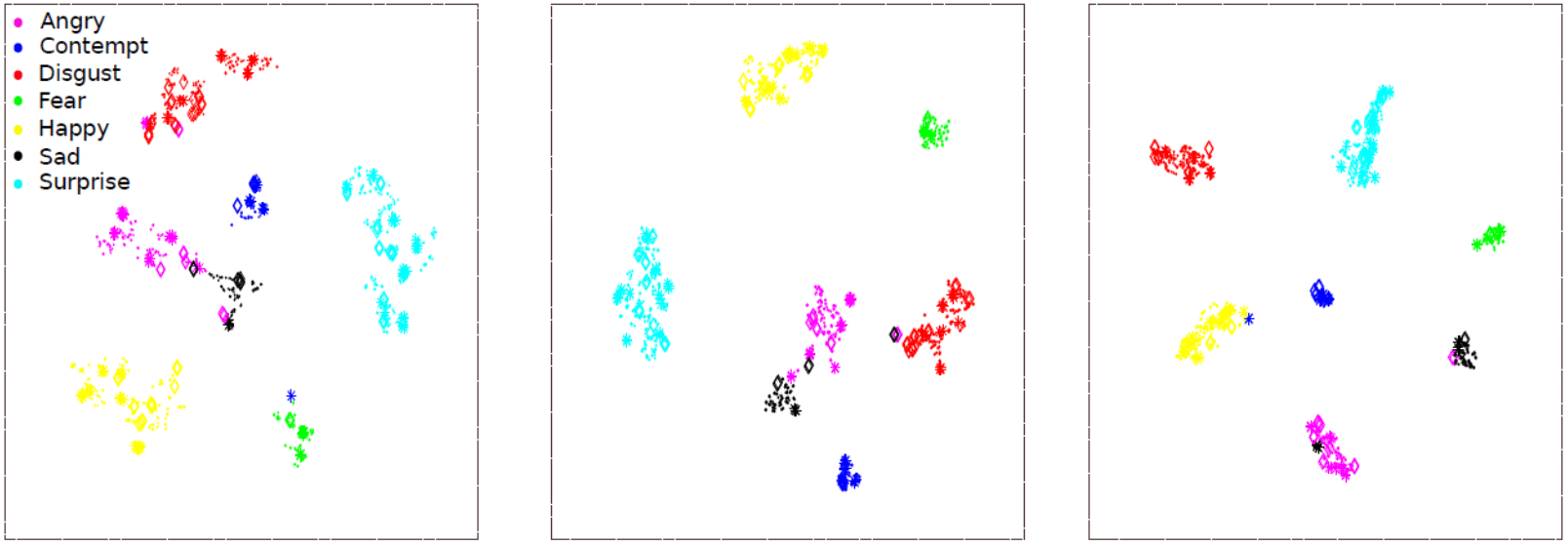
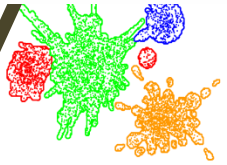


Fig. 3. A visualization study of the deep features learned by CNNs using (a) softmax loss, (b) softmax + center loss, and (c) softmax + island loss on the CK+ database. There are a total of 981 samples including 263 × 3 training data from 8 subsets, 32 × 3 validation data, and 32 × 3 testing data. The dots, stars, and diamonds represent training, validation, and testing data, respectively. Note that the features learned by the IL-CNN are well separated according to expressions. Best viewed in color.

[Cai2018]



Cluster distances

-softmax loss: the cosine distance between the means of each pair of expressions

- center loss / island loss: distance between each pair of the learned centers

TABLE X

AVERAGE BETWEEN-CENTER-DISTANCE ON FOUR DATASETS. THE BIGGER VALUE MEANS THE FURTHER DISTANCE.

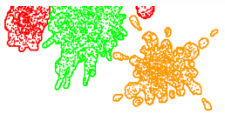
Dataset	CK+	MMI	CASIA	SFEW Val
Softmax loss	0.3418	0.2934	0.3537	0.3012
Center loss	0.3665	0.3165	0.3668	0.3274
Island loss	**0.4396**	**0.5250**	**0.4562**	**0.3778**

TABLE XI

AVERAGE SAMPLE-TO-CENTER DISTANCE ON FOUR DATASETS. THE SMALLER VALUE MEANS THE CLOSER DISTANCE.

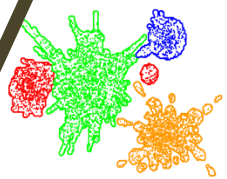
Dataset	CK+	MMI	CASIA	SFEW Val
Softmax loss	0.0301	0.0804	0.0536	0.0797
Center loss	0.0233	0.0634	0.0395	0.0694
Island loss	**0.0194**	**0.0618**	**0.0307**	**0.0689**

[Cai2018]



And so on…

- Angular-SoftMax Loss
- SphereFace: Deep Hypersphere Embedding
- CosFace: Large Margin Cosine Loss
- etc



[Cai2018]

Referinţe bibliografice

Biometrica si problematica generala:

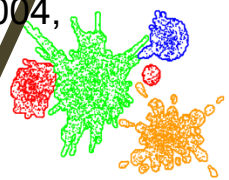
- [Belhumeur] Peter Belhumeur "*Introduction to Biometrics*", Columbia University http://www1.cs.columbia.edu/~belhumeur/courses/biometrics/2010/biometrics.html
- [Sinha] P. Sinha, B. Balas, Y. Ostrovsky, R. Russell, "*Face Recognition by Humans: 19 Results All Computer Vision Researchers Should Know About*", Proceedings of the IEEE, Vol. 94, No. 11, pp. 1948-1962, Noiembrie 2006.

Texture Unit:

- [He] He, D. and Wang, L., "*Texture unit, texture spectrum and texture analysis*," IEEE Trans. on Geoscience and Remote Sensing 28, pp.509-512 Iulie, 1990.

Local Binary Pattern:

- [Ojala1]T. Ojala et al, "*A comparative study of texture measures with classification based on feature distributions*", Pattern Recognition, vol. 29, pp. 51-59, 1996.
- [Ojala2] T. Ojala et al, "*Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 24, pp. 971-981, 2002.
- [Ahonen] Timo Ahonen et al, "*Face Recognition with Local Binary Patterns*", ECCV 2004, LNCS 3021, pp. 469-481, 2004
- [Pietikäinen] Matti Pietikäinen, "*Face Analysis Using Local Binary Patterns*" http://www.ee.oulu.fi/mvg



Referinţe bibliografice

Recunoastere cu metode 3D:

- [Heseltine] Thomas Heseltine, Nick Pears, Jim Austin *"Three-Dimensional Face Recognition Using Surface Space Combinations "*Advanced Computer Architecture Group, Department of Computer Science - University of York, www.cs.york.ac.uk/~tomh

Deep Learning

- Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. CVPR Link - https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering, CVPR Link - https://arxiv.org/pdf/1503.03832v3.pdf
- Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015, September). Deep face recognition. In *BMVC* (Vol. 1, No. 3, p. 6).Link: http://cis.csuohio.edu/~sschung/CIS660/DeepFaceRecognition_parkhi15.pdf
- Wen Y., Zhang K., Li Z., Qiao Y. "A Discriminative Feature Learning Approach for Deep Face Recognition" ECCV. 2016
- Jie Cai, Zibo Meng, Ahmed Shehab Khan, Zhiyuan Li, James O'Reilly, Yan Tong "Island Loss for Learning Discriminative Features in Facial Expression Recognition", FG 2018

