Face Recognition

Presenters: Shikhar Malhotra, Pranav Bhat, Pranav Sodhani, Atishay Aggarwal, Ameya Kabre

Definition

- **Face recognition system** is a computer application capable of identifying or verifying a person from a digital image

Successful systems in place

- **Apple** uses advanced deep learning techniques to bring facial recognition to iPhone; Only uses local data which doesn't require storing of faceprints on company servers
- Systems like **Google's FaceNet** and **Facebook's DeepFace** have made their way into web platforms, making it easier for users to tag photos and search for people


facebook
deepface
WHAT'S THE
BIG DEAL?

Deep Face Recognition

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman

http://www.robots.ox.ac.uk:5000/~vgg/publications/2015/Parkhi15/parkhi15.pdf

Objectives

- Building a large scale dataset (2.6M images over 2.6K people), assembled by a combination of humans and automation
- Propose a Convolutional Neural Network which can compete with state of the art methods and Internet giants such as Google and Facebook

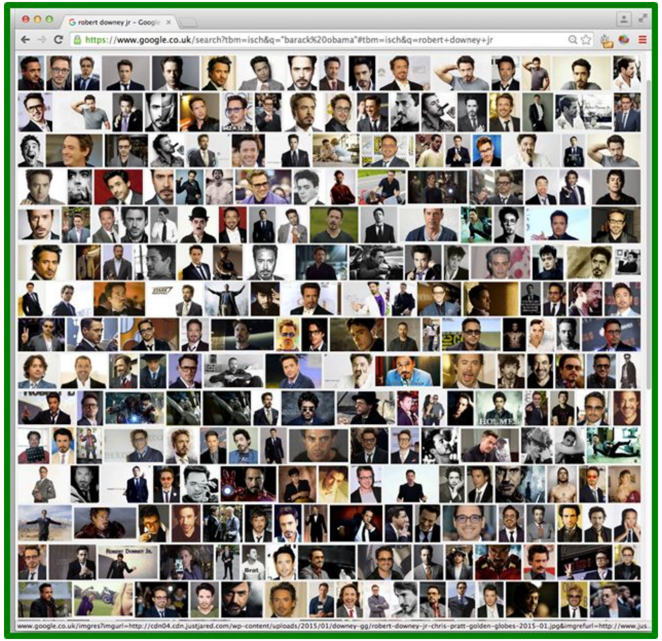
Dataset Collection

1. Bootstrapping and filtering list of candidate identities

- Focus on celebrities and politicians, easily available on internet
- Internet Movie Data Base (IMDB) celebrity list
- 5000 identities reduced to 3,250, by setting 90% purity bar for 200 images per candidate
- Lack of purity due to image scarcity

Dataset Collection

1. Bootstrapping and filtering list of candidate identities



Robert Downey Jr.

Dataset Collection

2. Enhancing dataset by collecting more images

- 2000 images per identity (2,622 celebrity names)
- Searching by appending keyword “actor”

Dataset Collection

3. Improving purity with automatic filter

- Remove erroneous images from each set using a trained classifier
- Linear SVM ranks 2000 images, top 1000 retained

4. Removing near duplicates

- Images differing in colour balance or with text superimposed are removed
- Clustering images and retaining one image per cluster

Dataset Collection

5. Manual filtering

- Multi-way CNN is built to discriminate between 2,622 face identities
- Ranked images displayed in blocks of 200, purity greater than 95%

Dataset Statistics after each stage

No.	Aim	Mode	# Persons	# images /person	Total # images	Anno. effort
1	Candidate list generation	Auto	5000	200	1,000,000	-
2	Collecting more images	Manual	2622	2,000	5,244,000	4 days
3	Rank image sets	Auto	2622	1000	2,622,000	-
4	Near duplicate removal	Auto	2622	623	1,635,159	-
5	Manual filtering	Manual	2622	375	**982,803**	10 days

Network Architecture and Training

- The face recognition problem was modelled as a N-way classification problem.
- The authors used a deep convolutional neural net, to associate with each image a score vector (1024 Dimensions, unit distance)
- These score vectors were compared to ground truth class identity by calculating empirical *soft-max log loss*.

Network Architecture and Training

- The score vectors were improved using a “triplet embedding” scheme.
- Learn a projection which is distinctive and compact, achieving dimensionality reduction at the same time
- The projection W’ is trained to minimise the empirical triplet loss -

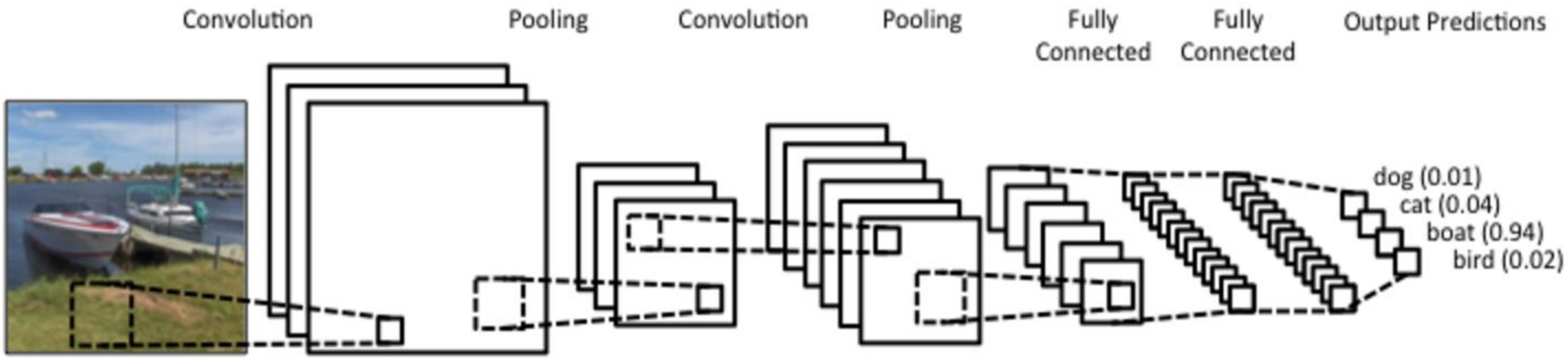
$$\sum_{(a,p,n)\in T} \max\{0, \alpha - \|\mathbf{x}_a - \mathbf{x}_n\|_2^2 + \|\mathbf{x}_a - \mathbf{x}_p\|_2^2\}$$

Network Architecture and Training

- CNN architecture A has 11 blocks, first 8 are set to be convolutional and the last 3 blocks are called Fully Connected (FC)
- First two FC layers have 4096 dimensional outputs, while the last FC layer has 1024 dimensions
- B and D networks have 2 to 5 additional convolution layers respectively
- Input is face image of size 224x224

Network Architecture and Training

- Goal: To find the network parameters which can minimize the average prediction log loss after the softmax layer
- Weights of filters chosen by random sampling from a Gaussian distribution with zero mean and 10^{-2} deviation



Datasets and Evaluation protocols

- Evaluation is performed on existing benchmark datasets
- Labelled Faces in the Wild (LFW): 13,233 images with 5,749 identities
- Youtube Faces (YTF): 3,425 videos of 1,595 people
- Verification accuracy and Equal Error Rate (EER): error rate at the ROC point where FP and FN rates are equal

Implementation

- MATLAB toolbox MatConvNet linked against NVIDIA CuDNN libraries to accelerate training
- When face transformation is used, 2D similarity transformation is applied

Performance Evaluation on LFW: Triplet-loss

No.	Network Config.	Dataset	Face Align Training	Face Align Testing	Embedding	100%-EER
1	A	Curated	No	No	No	92.83
2	A	Full	No	No	No	95.80
3	A	Full	No	Yes	No	96.70
4	**B**	**Full**	**No**	**Yes**	**No**	**97.72**
5	B	Full	Yes	Yes	No	97.07
6	D	Full	No	Yes	No	96.60
7	**B**	**Full**	**No**	**Yes**	**Yes**	**99.13**

Conclusion

- Proposed a procedure to obtain a large dataset with small label noise and involving minimum manual annotation
- Proved that a deep CNN without any embellishments and with appropriate training, can achieve results comparable to the state of the art

Fisher Vector Faces in the Wild

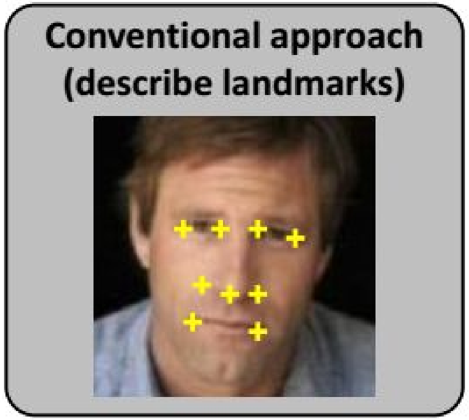
Karen Simonyan, Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman

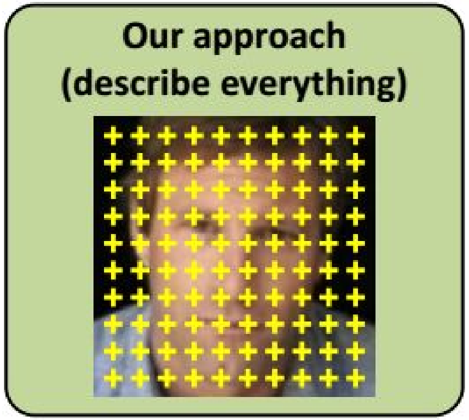
Visual Geometry Group, University of Oxford

https://www.robots.ox.ac.uk/~vgg/publications/2013/Simonyan13/simonyan13.pdf

Key Points

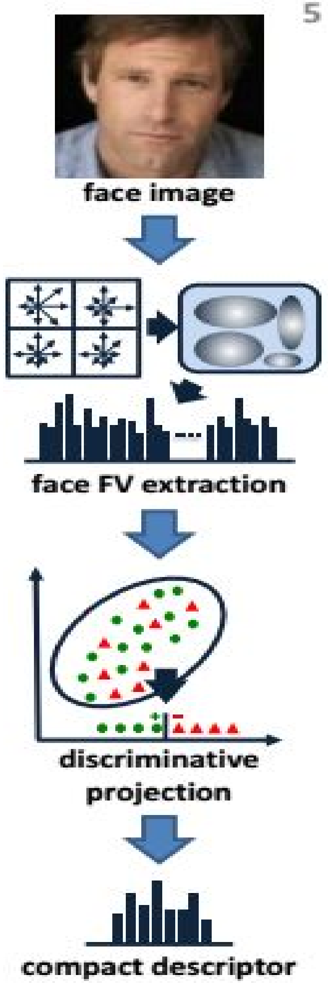
- Dense sampling
- Relevant face parts learnt automatically
- Compact and Discriminative





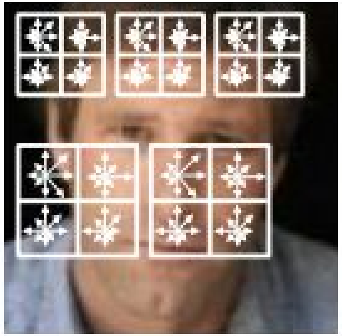
Process Overview

- Input: Face Image
- Deploy SIFT to extract features
- 26K 128-dimensional vectors
- Non-linear FV encoding
- Dimensionality reduction - non convex formulation



I. Dense SIFT

face image → set of local features



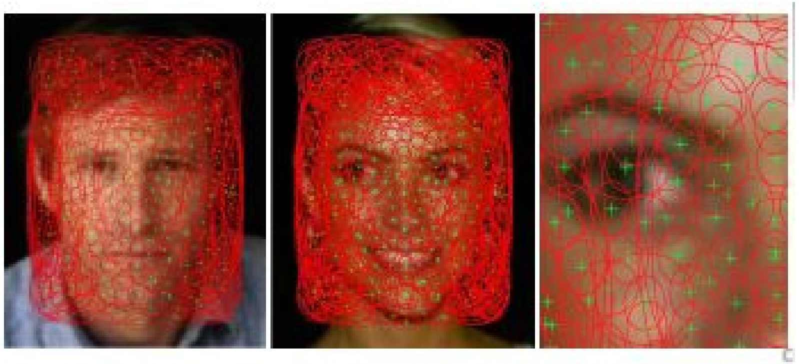
- SIFT - Scale Invariant Feature Transform
  - Scale-invariant
  - Rotation-invariant
  - Translation-invariant
- Scale-Space grid
- 24x24 window
- 1 pixel stride
- 5 scales
- 128-dim vectors -> PCA -> 64-dim
- 26K 64-dim feature vectors

II. Fisher Vector Encoding

set of local features → high-dim Fisher vector

- Describes a set of local features in a single vector
- Uses diagonal covariance GMM as a codebook
- GMM can be seen as a face model

ellipses – means & variances of GMM's (x,y) components



Issue: Spatial Information

- FV does not capture distribution of features in spatial domain
- Spatial pyramid coding - image divided into cells - FVs of all cells stacked
- Dimensionality increased with number of cells
- Solution - Augment the visual features with their spatial coordinates. => $[S_{xy}, x/w - 0.5; y/h - 0.5]$

III. Dimensionality Reduction

- Linear projection is used
- Goal: Find the projection matrix W
- Non convex formulation
- Reduces dimensionality drastically, thus can be used with large scale datasets.
- Dual Benefit - speed and accuracy

Implementation

1. Face Alignment and Extraction

- Viola Jones detector run on the image -> face detection
- 9 facial landmark positions identified
- Similarity transform applied to transform the face to a canonical frame.
- Extract a 160 x 125 face region around the landmarks for further processing.

2. Face descriptor Computation

- Publicly available packages used for FV encoding, SIFT
- Dimensionality reduction performed in matlab
- It takes few hours for computation on a single core machine

3. Diagonal Metric Learning

- Linear SVM is used
- Features = vectors of squared differences between corresponding components of two FVs
- Learning is basically performed to extract semantic face attributes as facial features which could be used for identification, etc

4. Horizontal Flipping

- The test set is augmented.
- Horizontal reflections of 2 compared images are taken.
- The distances between the 4 possible combinations of the original and reflected images is averaged.

Evaluation

- Labelled faces in the Wild dataset used
- 13233 images of 5749 people - considered benchmark.
- Divided into 10 disjoint splits
- 600 predefined image pairs: 300 positive pairs (same person), 300 negative pairs (different people)
- 10 fold cross validation

Training the Data

- PCA projections for SIFT
- Gaussian mixture models
- Discriminative Fisher vector projections
- All these are trained independently for each fold

Evaluation Metrics

- Receiving Operating Characteristic Equal Error Rate (ROC-EER)
- Gives the accuracy at the ROC operating point, where false positives and false negatives rates are equal
- Reflects quality of ranking obtained by scoring image pairs
- Different stages of the proposed framework can be compared.

- Final classification performance is reported in terms of classification accuracy
- Classification accuracy = percentage of image pairs classified correctly

Evaluation Protocols

- The LFW specifies 2 protocols:
- Restricted setting - predefined image pairs for each split used for training
- Unrestricted setting - identities within each split are given, an arbitrary number is formed for positive and negative training pairs

Experiments

- Unrestricted setting and unaligned LFW images.
- The parameters SIFT density, GMM size, effect of spatial augmentation, dimensionality reduction, distance function and horizontal flipping.
- The results are summarized in the following slide

SIFT density	GMM Size	Spatial Aug.	Desc. Dim.	Distance Function	Hor. Flip.	ROC-EER,%
2 pix	256		32768	diag. metric		89.0
2 pix	256	✓	33792	diag. metric		89.8
2 pix	512	✓	67584	diag. metric		90.6
1 pix	512	✓	67584	diag. metric		90.9
1 pix	512	✓	128	low-rank PCA-whitening		78.6
1 pix	512	✓	128	low-rank Mah. metric		91.4
1 pix	512	✓	256	low-rank Mah. metric		91.0
1 pix	512	✓	128	low-rank Mah. metric	✓	92.0
1 pix	512	✓	2×128	low-rank joint metric-sim.		92.2
1 pix	512	✓	2×128	low-rank joint metric-sim.	✓	93.1

Table 1: **Framework parameters**: The effect of different FV computation parameters and distance functions on ROC-EER. All experiments done in the unrestricted setting.

Observations

- Performance increases with:

1. Denser Sampling
2. More clusters in GMM
3. Spatial augmentation (with minor increase in dimensionality)
4. Dimensionality reduction
5. Horizontal Flipping

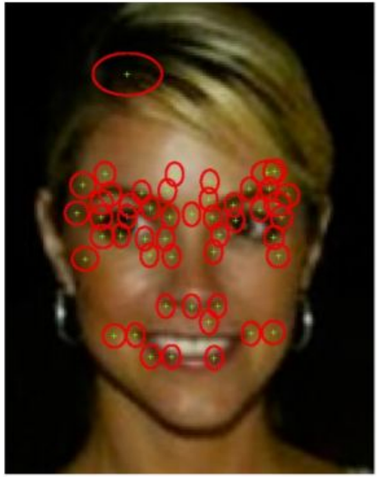
- Projection to higher dimensions - overfitting

Model Visualisation

- Model can capture face specific features
- Each GMM component corresponds to a part of the Fisher Vector and to a group of columns in the projection matrix.
- Certain Gaussians are important and can be found by computing the energy of the corresponding column group

Learnt Model Visualisation

all Gaussians



important (top-50 Gaussians)

irrelevant (bottom-50 Gaussians)

Gaussian ranking (for visualisation):

GMM component → FV sub-vector → W sub-matrix → its energy

dimensionality reduction projection $W =$

1st Gaussian	2nd Gaussian		512th Gaussian

Results

For unrestricted setting:

- 93.03% face verification accuracy
- Almost equal to state of the art (93.18%) that uses landmark detection
- Author’s algorithm:
    - Sampled the features densely instead
    - 10 fold cross validation

Results

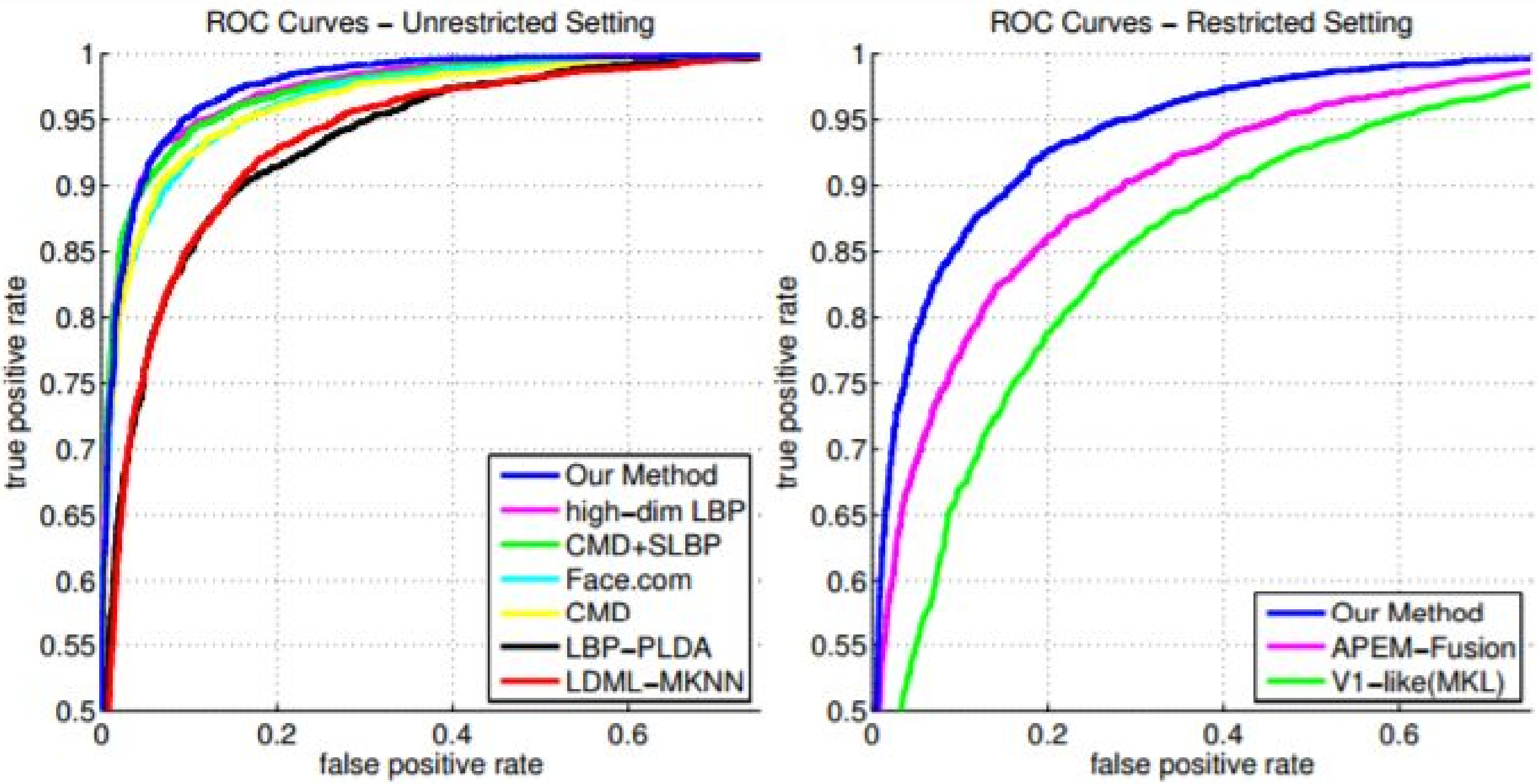
For restricted setting:

- Centred 150 x 150 crops of LFW dataset used for training.
- Training data insufficient for dimensionality reduction learning, thus a diagonal metric function using SVM learnt
- Verification accuracy of 87.47%
- 3.4% greater than the existing best.

Results

- Even though some methods use GMMs for dense feature clustering, they do not use Fisher Vector, keeping all extracted features for matching - limitation.

- Dimensionality of Fisher Vector does not depend upon the number of features it encodes.



ROC Curves – Unrestricted Setting
true positive rate
false positive rate
Our Method
high–dim LBP
CMD+SLBP
Face.com
CMD
LBP–PLDA
LDML–MKNN
ROC Curves – Restricted Setting
true positive rate
false positive rate
Our Method
APEM–Fusion
V1–like(MKL)

Conclusion

- Use of dense features avoids applying landmark detectors
- Huge dimensionality reduction
- Effective and efficient face descriptor computation, thus can be used for large datasets
- Future work - Handle multi-feature image representations for which a framework is already in place

Thank You for

Listening...

any questions?