National Research University Higher School of Economics
Nizhny Novgorod

Deep Convolutional Neural Networks and Maximum-Likelihood Principle in Approximate Nearest Neighbor Search

Andrey V. Savchenko
Laboratory of Algorithms and Technologies for Network Analysis (LATNA)
Email: avsavchenko@hse.ru

2017

Outline

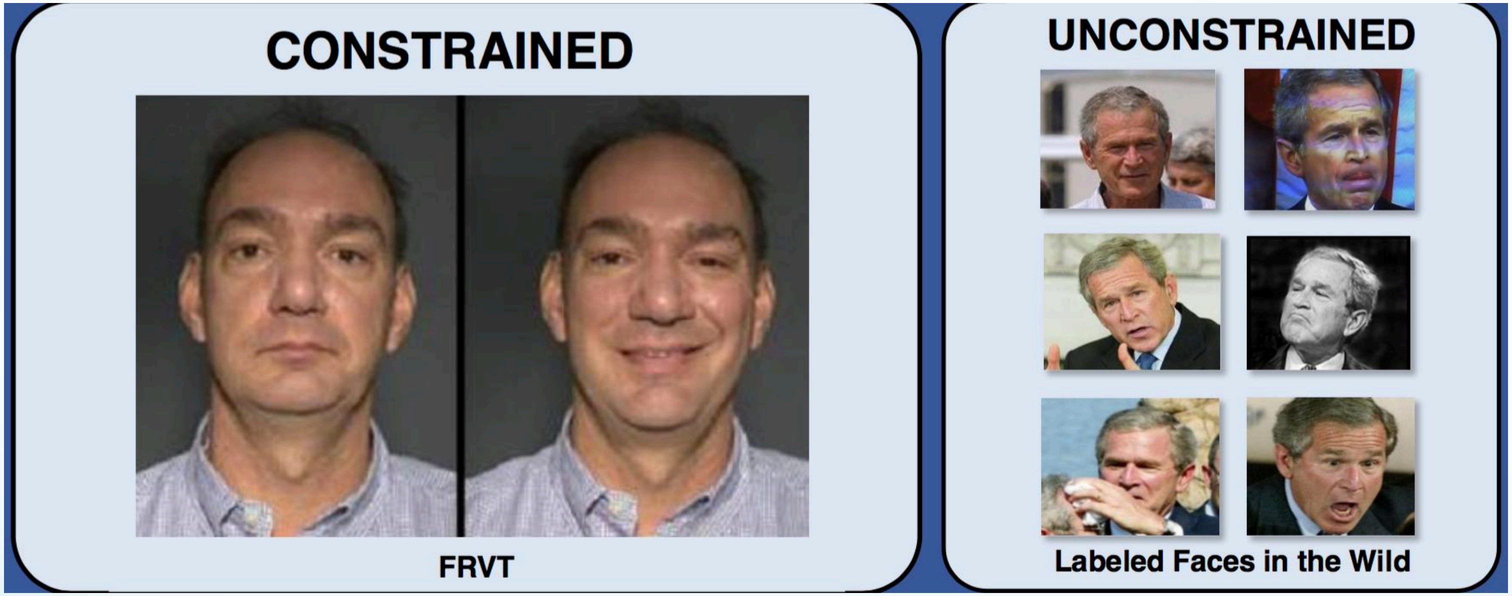
1. Image recognition and small sample size (SSS) problem
2. Proposed approximate nearest neighbor algorithm
3. Experimental results in unconstrained face recognition
4. Conclusion and future work

Image recognition problem

It is required to **assign** the query object X to one of $C>1$ **classes** specified by the **model** objects $\{X_r\}$. Class label $c(r)$ of the r-th model is known (supervised learning).

Small-sample-size (SSS) problem: $C \approx R$. In the worst case, one sample per class ($C=R$)

Examples	Face recognition with one sample per class



Conventional approach	**Nearest neighbor** (NN) classifier Recent trends: recognition as sequential verification (binary classification)
Key problem	Performance of exhaustive search is inappropriate for large databases (thousands of classes)

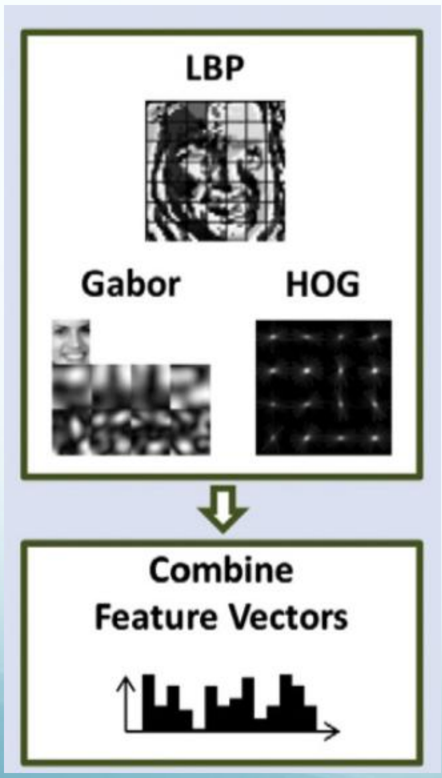
Feature Extraction. Deep Neural Networks (DNN)

Nearest neighbor classifier for small sample size problem

$$\nu = \operatorname*{argmin}_{r \in \{1,\dots, R\}} \rho(X, X_r)$$

Facial images X and X_r are described by M-dimensional feature vectors

Progress in unconstrained face recognition

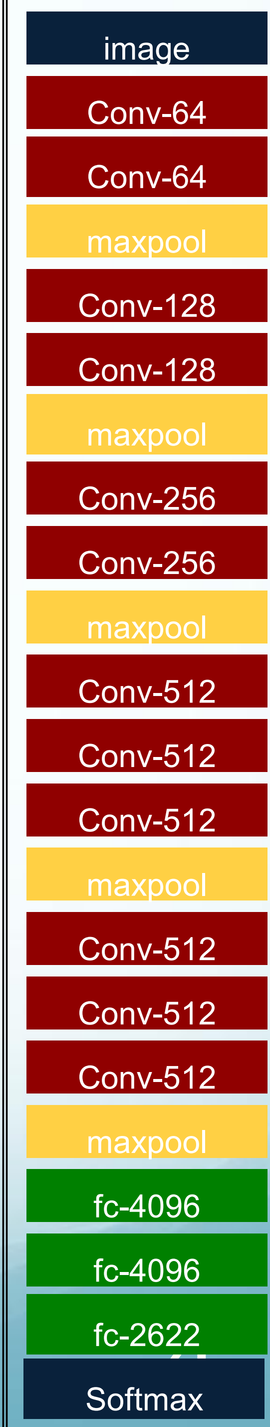


Conventional features (HOG+LBP+Gabor).
Ortiz E.G., Becker B.C. Face recognition for web-scale datasets. Computer Vision and Image Understanding 2014
Accuracy 50-80%

DNN bottleneck features.
Oxford Visual Geometry Group:
Parkhi O.M., Vedaldi A., Zisserman A. Deep face recognition. Proc. of the British Machine Vision. 2015.
Accuracy 85-97%

Key problem

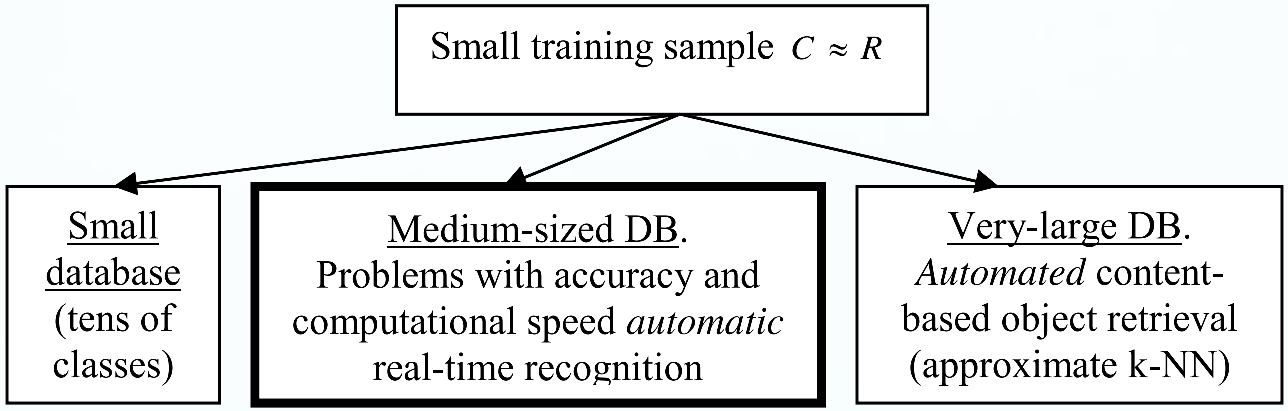
Insufficient runtime complexity of DNN approach



Real-time recognition with large database

k-NN rule requires the brute force of the whole database: time complexity O(*RM*).

Multi-objective optimization: $\overline{\alpha} \to \min \quad \bar{t} \to \min$



- Parallel computing
- Simplification of dissimilarity measure.
- Approximate nearest neighbor (ANN) methods:

1. **ANN** library (*Arya, etc. Journal of the ACM, 1998*): kd-trees. Only Minkowski distances are supported.
2. **Hashing** Techniques, LSH (Locality-Sensitive Hashing) (*Gionis, Indyk, Motwani, R. Proc. of VLDB, 1999*). Applications in **Google Correlate** (*Vanderkam, Schonberger, Rowley, Kumar, Nearest Neighbor Search in Google Correlate. Tech. report, 2013*)
3. **FLANN** library (*Muja, Lowe. Proc. of VISAPP, 2009*)
4. **NonMetricSpaceLib** (*Boytsov, Bilegsaikhan. Proc. of SISAP, LNCS, 2013*)

Maximum Likelihood (ML). Proposed approach

Idea: on each step the next instance is chosen to maximize likelihood of the previously calculated distances

$$r_{k+1} = \underset{\nu \in \{1,\dots,R\} - \{r_1,\dots,r_k\}}{\operatorname{argmax}} \prod_{i=1}^{k} f\left(\rho\left(X, X_{r_i}\right) \middle| W_\nu\right)$$

The known assumption about **normal distribution** of dissimilarity measure, if the number of features *M* is high and the distance between images is defined as an average distance between corresponding features (Burghouts et al. NIPS 2008; P'kalska & Duin. Proc. of ICPR 2000).

$$f\left(\rho\left(X, X_{r_i}\right) \middle| W_\nu\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\left(\rho\left(X, X_{r_i}\right) - \rho_{\nu, r_i}\right)^2 / \sigma^2\right]$$

We additionally assume the identical variance σ for all classes. Final rule:

$$\boxed{r_{k+1} = \underset{\mu \in \{1,\dots,R\} - \{r_1,\dots,r_k\}}{\operatorname{argmin}} \sum_{i=1}^{k} \left(\rho\left(X, X_{r_i}\right) - \rho_{\mu, r_i}\right)^2}$$

Range query. Search procedure is terminated, if:

$$\rho\left(X, X_{r_{k+1}}\right) < \rho_0$$

But! Quadratic memory complexity → Pivot-based techniques

Proposed algorithm

1. Preliminary step
   1. Randomly choose $P << R$ pivots (instances) $\left\{ X_{r_1}, \dots, X_{r_p} \right\}$
   2. Calculate PR distances between all instances and pivots
2. PROPOSED RECOGNITION PROCEDURE
   1. Calculate P distances from the input image X to all pivots
   2. Repeat for each non-pivot instance v
      - 2.1 Initialize log-likelihoods $\varphi_{\Sigma;\nu} = \sum_{p=1}^{P} \left(\rho\left(\mathbf{x}, \mathbf{x}_{r_p}\right) - \rho_{\nu,p} \right)^2$
   3. Partial sort the array $\{\varphi_{\Sigma;\nu}\}$ in ascending order and (E_{max}-P) extract smallest elements
   4. Repeat for $\nu \in \{1, \dots, R\} - \{r_1, \dots, r_P\}$
      - 4.1 Return v-th instance, if distance to X is less than a threshold
   5. Return the closest checked image

The worst-case runtime complexity: $O(E_{\max} \cdot M + R \log E_{\max})$

Experimental study

The proposed approach is compared with the following methods

1. **Support Vector Machine (SVM)**
2. **Brute force** implementation of **1-NN**
3. **ANN Methods:**

- **Randomized kd-tree** (*Silpa-Anan C., Hartley R., CVPR, 2008*)
- **Perm-sort** (*Gonzalez E.C.et al., IEEE Trans. on PAMI, 2008*)
- **ML-ANN** (*Savchenko A.V., Pattern Recognition. 2017*) – specially designed for dissimilarities between probabilistic densities

Hardware

MacBook Pro 2015 laptop (2.2 GHz Intel Core i7, 16 Gb RAM)

Feature extraction (Caffe framework)

1. VGGNet: Oxford VGG face model (Parkhi et al. Proc. of the British Machine Vision 2015). M=4096 non-negative DNN features from fc7 layer
2. LCNN: Lightened CNN model, version C (Wu X.et al. Proc. of CVPR 2015). M= 256 DNN features from eltwise_fc2 layer

Dissimilarity measures:

1. L2 (Euclidean) metric
2. Chi-squared distance for VGGNet features

Experimental setup

1. Face identification

C=1680 persons.

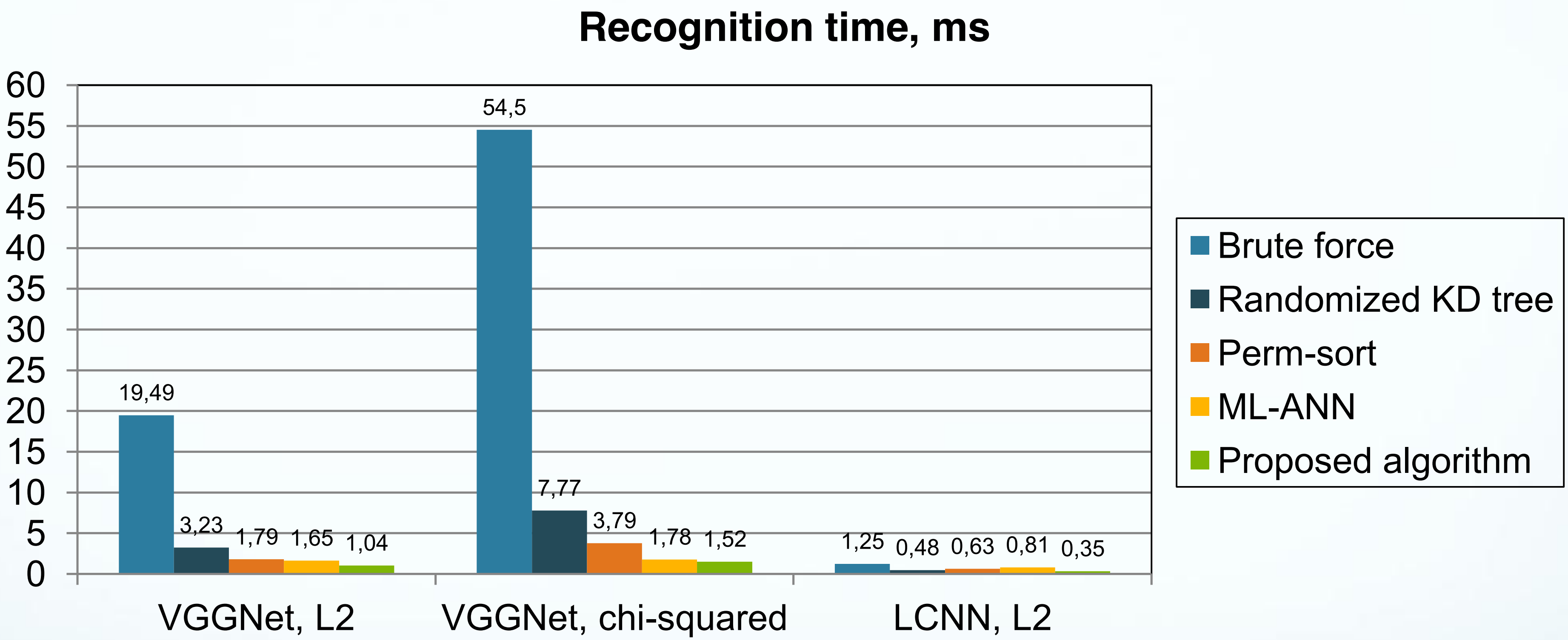
Training set: R=4570 photos. Validation set: 4464 photos, both from LFW (Labeled Faces in the Wild) dataset

2. Still-to-video recognition

C=1589 persons

Training set: 4732 photos from LFW. Validation set: 122756 frames from YTF (YouTube Faces) dataset

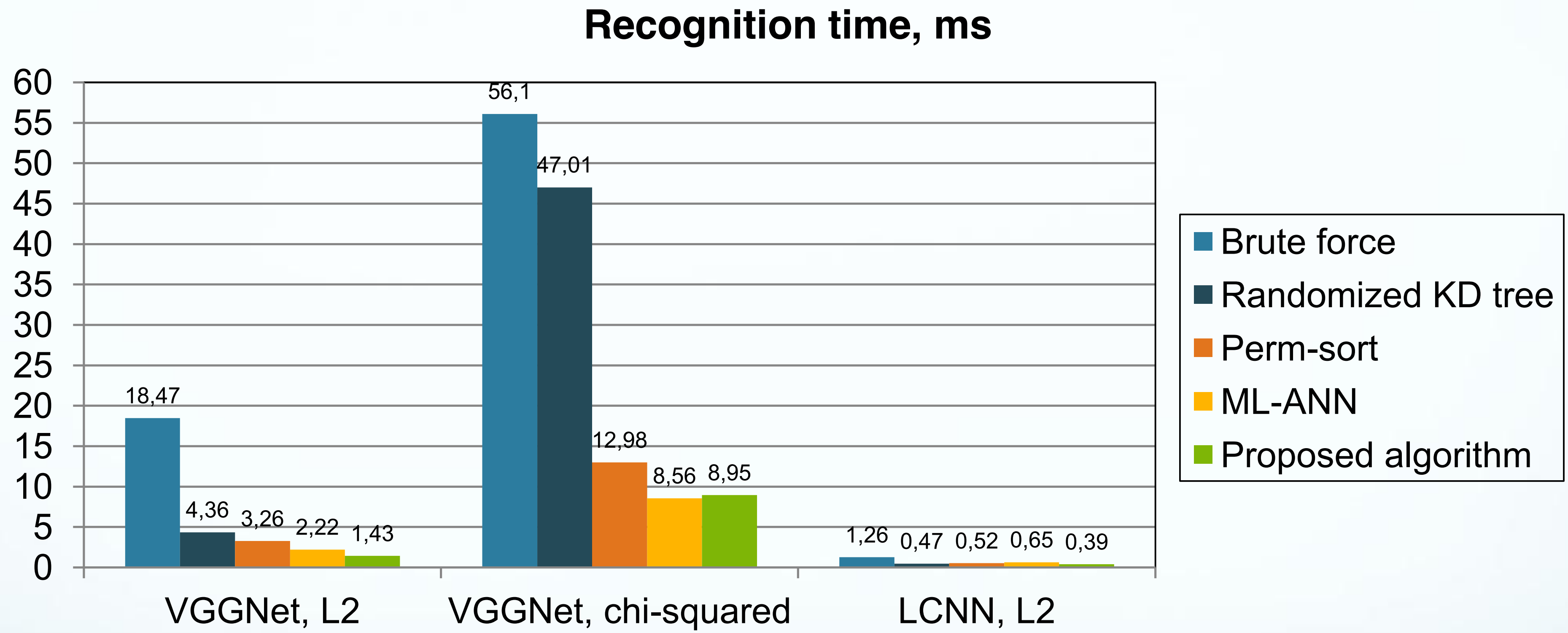
Experimental results (1a). LFW



Error rate, %

	VGGNet, L2	VGGNet, chi-squared	LCNN, L2
SVM	49.6	-	28.9
1-NN (Brute force)	10.7	10.2	9.6
Randomized KD tree	11.0	10.6	9.9
Perm-sort	10.9	10.6	9.9
ML-ANN	10.9	10.6	10.5
Proposed algorithm	10.9	10.5	9.9

Experimental results (1b). YTF



Error rate, %

	VGGNet, L2	VGGNet, chi-squared	LCNN, L2
SVM	87.1	-	82.5
1-NN (Brute force)	59.6	59.4	53.1
Randomized KD tree	59.9	60.7	53.5
Perm-sort	59.7	59.8	53.5
ML-ANN	59.7	59.4	53.6
Proposed algorithm	59.7	59.4	53.7

Conclusion

1. Our ANN algorithm is based on the pivot selection techniques and maximum likelihood method.
2. We experimentally demonstrated that the proposed approach can be efficiently applied with DNN (bottleneck) features in face recognition.
3. Preliminary results for very-large Casia-WebFaces database (10,000 persons, 500,000 images) also showed the superiority of our algorithm: it is 23 and 10.5 and 3.7 times faster, than brute force and perm-sort, respectively.

Future work

1. More accurate assumptions about probability distribution of distances between DNN (non-equal variance, Weibull distribution, etc.)
2. Experimental study of video-based recognition (YouTube Faces)

Current issues of unconstrained face recognition methods:

1. Accuracy is still too low: 45% for Casia WebFaces database.
2. DNN models are trained and tested with images of celebrities (smiling, beautiful, young). What about more realistic images?

Thank you for listening!
Questions?