FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

# Computer Vision Project – Summer Term 2025

# Face Recognition

Dr.-Ing. Thomas Köhler
Pattern Recognition Lab, Friedrich-Alexander-Universität (FAU) Erlangen-Nürnberg
e.solutions GmbH, Erlangen
July 7, 2025

# Face Recognition Example Use Cases

Unlock smart phones via face recognition[1]


[1] https://www.apple.com/iphone-x/#face-id

# Face Recognition Example Use Cases

Identification of occupants and personalization of vehicle settings

# Scope of this Project

In the lecture:

- Learn how to design and evaluate the core of current facial recognition systems from a technical point of view
- Overview of modern machine learning methods in this field
- Discussion of their strengths and limitations

In the exercise:

- Implementation of a simple system comprising basic functionality for face verification, identification, and clustering
- Evaluation of face recognition algorithms
- Design own recognition method and participate in a challenge

# Organization

- Exercises will be held virtually (start: CW 28/2025)
  - We will propose a couple of available time slots at the beginning of the week
  - Please drop me an e-mail with your availability
  - We will send you an invitation
  - **Be well prepared (only 15 - 30 minutes per group/week)!**
- Final submission via StudOn
  - Upload your source codes to StudOn **before the submission deadline!**
  - We will schedule a meeting to review your final solution
  - You will present your solution and we will ask questions
  - All exercise partners shall contribute equally to all tasks and shall be able to answer questions on each of the exercises!
  - **No certificate without presentation, review, and answering questions!**
- Additional exercise
  - Required to gain 10 ECTS
  - Will be a challenge with winners (details to be announced) :-)

# Outline

# Introduction

# The Face as a Biometric Marker

Biometric markers for identification[2]:

- Fingerprint
- Iris
- Speech
- Face

Face vs. other markers:

- Face the only marker that can be captured at large distances
- Can work with low-cost hardware

[2] Jain, Anil K., and Stan Z. Li. Handbook of face recognition. New York, Springer, 2011.

# Challenges

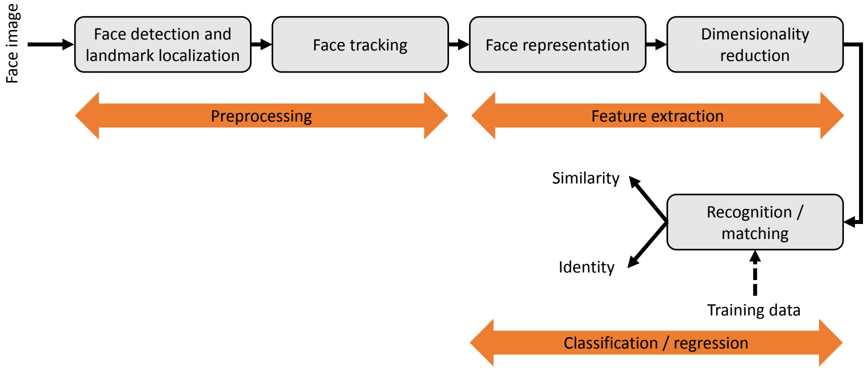
Intrinsic conditions:

- High intra-subject variation: expressions, changes in facial hair or pose, aging
- Low inter-subject variation: similar skin or hair color, similar eye glasses

Extrinsic conditions:

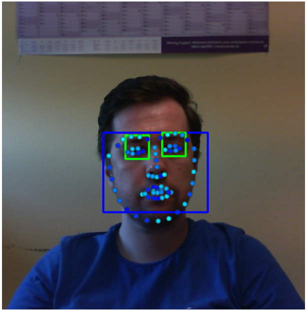
- Varying illumination: images captured at day and at night
- Image quality: facial data processed with different image or video codecs

# Face Recognition Pipeline



# Face Detection, Landmark Localization, and Tracking

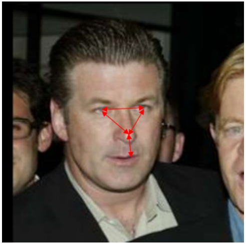
- Detection of face region
- Detection of eye regions
- Extracting landmarks to model pose and facial expression
- Alignment using detected landmarks
  - Compensate for different head poses
  - Eyes, nose, and mouth at predefined positions
- Track bounding box and landmarks in video data



# Face Representation

## Geometric approach:

- Distances, areas, or angles between salient points (eyes, nose, mouth)
- Obtained by feature detection algorithms
- Requires hand-crafted features



## Data-driven approach:

- Image intensities form raw features
- Learn suitable representation from exemplars
- Requires training from large datasets

$$\begin{pmatrix} 124 \\ 122 \\ 116 \\ 130 \\ 132 \\ \dots \end{pmatrix}$$

# Face Recognition Problem Statements

## Verification (one-to-one matching):

- Given one probe image and one gallery image
- Check if both images show the same identity

## Identification (one-to-many matching):

- Given one probe image and a set of gallery images with identity labels
- Retrieve identity of probe image

## Clustering (many-to-many matching):

- Given a set of unlabeled face images
- Cluster according to the identities captured in the data

# Face Representation

# Data-Driven Face Representations

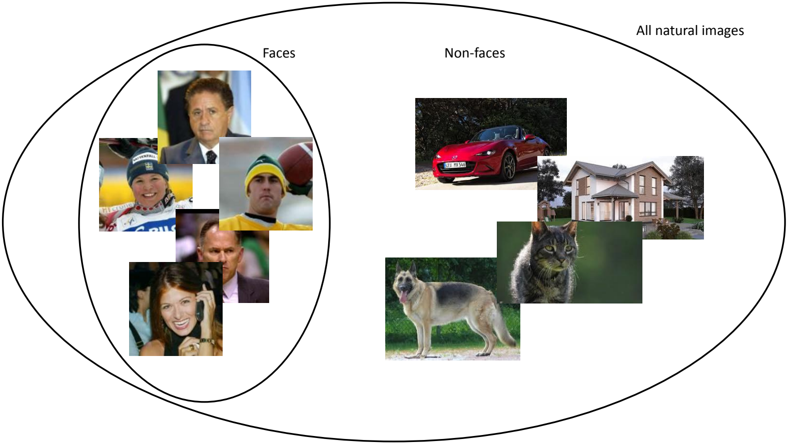
## How to represent faces in digital images?

$$\begin{pmatrix} 124 \\ 122 \\ 116 \\ 130 \\ 132 \\ \dots \end{pmatrix}$$

- Grayscale image with $M \times N$ pixels and $b$ bits per pixel (e. g. $64 \times 64$, 8 bit) $\rightarrow (2^b)^{M \cdot N}$ different images (e. g. $256^{4096} \gg 10^{1000}$)
- Only a small fraction of images corresponds to valid faces

# Manifold of Face Images

Faces are a small subset of the natural image manifold



# Learning Face Representations: Eigenfaces

- Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, $\boldsymbol{x}_i \in \mathbb{R}^D$ be a set of $n$ face images with mean:

$$\boldsymbol{\mu}_x = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \tag{1}$$

and covariance:

$$\boldsymbol{\Sigma}_x = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu}_x)(\boldsymbol{x}_i - \boldsymbol{\mu}_x)^\top \tag{2}$$

- Project each $\boldsymbol{x}_i$ to a $M$-dimensional space ($M \ll D$):

$$\boldsymbol{y}_i = \boldsymbol{W}\boldsymbol{x}_i \in \mathbb{R}^M \tag{3}$$

- Seek linear transform $\boldsymbol{W} \in \mathbb{R}^{M \times D}$ that maximizes the variance of $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$:

$$\begin{aligned}\boldsymbol{\Sigma}_y &= \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{y}_i - \boldsymbol{\mu}_y)(\boldsymbol{y}_i - \boldsymbol{\mu}_y)^\top \\ &= \boldsymbol{W}^\top \boldsymbol{\Sigma}_x \boldsymbol{W}\end{aligned} \tag{4}$$

$\rightarrow$ Principal Component Analysis (Principal components $\equiv$ Eigenfaces)

# Learning Face Representations: Fisherfaces

- Exploit two constraints (one unique face $\equiv$ one class):
  - Between-class scatter shall be maximum
  - Within-class scatter shall be minimum
- Seek the transform $\boldsymbol{W} \in \mathbb{R}^{M \times D}$ maximizing Fisher's linear discriminant:

$$J(\boldsymbol{W}) = \frac{\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\text{inter}} \boldsymbol{W}}{\boldsymbol{W}^\top \boldsymbol{\Sigma}_{\text{intra}} \boldsymbol{W}} \tag{5}$$

Intra-class and inter-class scatter for $c$ classes:

$$\boldsymbol{\Sigma}_{\text{intra}} = \sum_{i=1}^{c} \frac{1}{|\mathscr{X}_i|} \sum_{\boldsymbol{x}_k \in \mathscr{X}_i} (\boldsymbol{x}_k - \mu_i)(\boldsymbol{x}_k - \mu_i)^\top \tag{6}$$

$$\boldsymbol{\Sigma}_{\text{inter}} = \sum_{i=1}^{c} (\mu_i - \mu)(\mu_i - \mu)^\top \tag{7}$$

$\mu_i$ is the mean of all faces $\mathscr{X}_i$ in the $i$-th class and $\mu$ is the overall mean face

# Learning Face Representations: Deep Features

Limitations of the methods discussed so far:

- Determine face representations under a linear model
- Limited robustness against pose or illumination variations

Extension:

- Use a non-linear face representation $f_\theta(\boldsymbol{x})$
- The transform $f_\theta(\boldsymbol{x})$ is implemented by a deep neural network
- Provides a face representation (aka. embedding)
- Parameters $\theta$ are learned from exemplars

# Recap: Convolutional Neural Networks (CNNs)

Neural network:

- Computation graphs comprising neurons
- Propagation of input signals through the network to obtain outputs

Network design with input, output, and hidden layers:

- Convolutional layer: convolution of input neuron activation with filter kernel
- Pooling layer: fusion of clusters of input neuron activation
- Locally/fully connected layer: weighted sum of input neuron activation

$\theta$: Parameters of all layers in the network

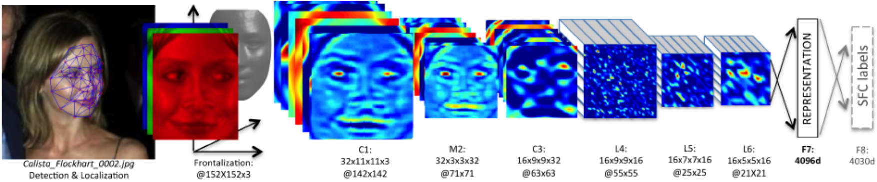
# Face Representation via Classification

- Design neural network to classify face images according to their identity to one out of $c \geq 2$ classes
- Fully connected output layer $\left(\boldsymbol{W}, \boldsymbol{b}\right)$ to model probability distribution:

$$p_i = \frac{\exp\left(\boldsymbol{w}_i^\top \boldsymbol{x} + b_i\right)}{\sum_{j=1}^{c} \exp\left(\boldsymbol{w}_j^\top \boldsymbol{x} + b_j\right)}, \qquad i = 1, \ldots, c \tag{8}$$

- Train network parameters on face exemplars by minimizing misclassification error (e.g. cross entropy loss)
- Activation of fully connected layer $\equiv$ face representation (aka. embedding)

# Example: Facebooks DeepFace (trained on 4M faces)[3]




- Face frontalization for preprocessing
- Eight layer network: convolutional (C1, C3), max-pooling (M2), locally connected (L4, L4, L6), and fully connected layers (F7, F8)
- Feature maps describe face from high-level to low-level
- Activation of F7 fully connected layer used as face representation

[3] Taigman, Yaniv, et al. "Deepface: Closing the gap to human-level performance in face verification." Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2014.

# Face Representation via Contrastive Learning

Discussion of the classification-based approach:

- Classification requires class labels (identities)
- Models discriminative features for face recognition only implicitly
- Learned features are not necessarily optimal

Extension to regression-based / contrastive losses:

- Can be used for pre-training or fine-tuning
- No full labeling of dataset required

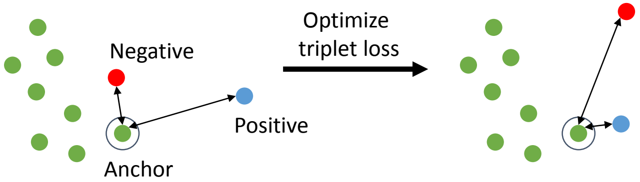
# Triplet Loss

- Let $\boldsymbol{x}_i^a$ be an anchor face, $\boldsymbol{x}_i^p$ a face of the same subject (positive) and $\boldsymbol{x}_i^n$ be a face of different subject (negative)
- Ensure for all triplets $(\boldsymbol{x}_i^a, \boldsymbol{x}_i^p, \boldsymbol{x}_i^n)$ with margin $\alpha$:

$$\underbrace{||f_\theta(\boldsymbol{x}_i^a) - f_\theta(\boldsymbol{x}_i^p)||_2^2}_{\text{anchor-to-positive distance}} + \alpha < \underbrace{||f_\theta(\boldsymbol{x}_i^a) - f_\theta(\boldsymbol{x}_i^n)||_2^2}_{\text{anchor-to-negative distance}} \tag{9}$$

- Penalize distances of positive and negative samples w.r.t. the anchor

$$\mathscr{L}_{\text{triplet}}(\theta) = \sum_{i=1}^{n} ||f_\theta(\boldsymbol{x}_i^a) - f_\theta(\boldsymbol{x}_i^p)||_2^2 - ||f_\theta(\boldsymbol{x}_i^a) - f_\theta(\boldsymbol{x}_i^n)||_2^2 + \alpha \tag{10}$$



# Practical Considerations on Deep Features

## Learning:

- Learn $f_\theta(\boldsymbol{x})$ on large datasets (Facebook: 4M faces, Google: 200M faces)
- Computationally very demanding
- Implemented on graphics processing units (GPU)

## Inference:

- Determine face representation $f_\theta(\boldsymbol{x})$ by forward pass
- Efficient to compute using additional optimizations (weight quantization)
- Use face representation within lightweight classification or clustering models
- Can be implemented on embedded devices

# Selected Topics in Face Recognition

# Face Verification

- Given two face images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, is the image pair of the same person?
- Define a suitable distance measure $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$, e.g., cosine distance:

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{||f_\theta(\boldsymbol{x}_1) - f_\theta(\boldsymbol{x}_2)||_2^2}{||f_\theta(\boldsymbol{x}_1))||_2^2 + ||f_\theta(\boldsymbol{x}_2)||_2^2} \tag{11}$$

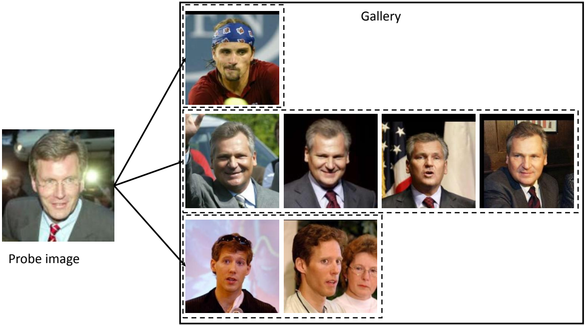
- Verification using the distance measure and threshold $\tau$:

$$v(\boldsymbol{x}_1, \boldsymbol{x}_2) = \begin{cases} \text{same} & \text{if } d(\boldsymbol{x}_1, \boldsymbol{x}_2) \leq \tau \\ \text{not same} & \text{if } d(\boldsymbol{x}_1, \boldsymbol{x}_2) > \tau \end{cases} \tag{12}$$

- Alternatively, use similarity measure $s(\boldsymbol{x}_1, \boldsymbol{x}_2) = -d(\boldsymbol{x}_1, \boldsymbol{x}_2)$

# Face Identification

- Given a probe image $\boldsymbol{x}_{\text{probe}}$ of unknown identity
- Determine identity $I(\boldsymbol{x}_{\text{probe}})$ from labeled images in a gallery




## Two protocols:

- Closed-set: all possible identities of probes are contained in the gallery
- Open-set: identities of some probes are missing in the gallery

# Face Identification with Closed-Set Protocol

## Repeated face verification using k-nearest neighbors (k-NN) classifier:

- For the probe image $\boldsymbol{x}_{\text{probe}}$, find the $k$ closest gallery images $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ according to a distance $d(\boldsymbol{x}_{\text{probe}}, \cdot)$
- Identity of $\boldsymbol{x}_{\text{probe}}$ is the majority (mode) in $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$:

$$I(\boldsymbol{x}_{\text{probe}}) = \text{mode}(I(\boldsymbol{x}_1), \ldots, I(\boldsymbol{x}_k)) \tag{13}$$

- Alternatively, we can consider similarities $s(\boldsymbol{x}_{\text{probe}}, \cdot) = -d(\boldsymbol{x}_{\text{probe}}, \cdot)$

## Other discriminative classification models:

- Support vector machine (SVM)
- Random forests
- Boosting methods

## Face Identification with Open-Set Protocol

Handle open space using thresholded nearest neighbors:

- Extract best matching gallery image $\boldsymbol{x}_{\text{match}}$ with minimum distance to $\boldsymbol{x}_{\text{probe}}$
- Possibly assign "unknown" identity to $\boldsymbol{x}_{\text{probe}}$:

$$I(\boldsymbol{x}_{\text{probe}}) = \begin{cases} \text{mode}(I(\boldsymbol{x}_1), \ldots, I(\boldsymbol{x}_k)) & \text{if } d(\boldsymbol{x}_{\text{probe}}, \boldsymbol{x}_{\text{match}}) \leq \tau \\ \text{unknwon} & \text{if } d(\boldsymbol{x}_{\text{probe}}, \boldsymbol{x}_{\text{match}}) > \tau \end{cases} \tag{14}$$

$\tau$ is the face verification threshold

# Learning Open-Set Models using Known Unknowns

Extend the training setup of open-set models:

- Training set with KCs and known unknown classes (KUCs)

$$\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{T} \tag{15}$$

- Labels $y_i$ can encode any KC ($\mathcal{C}_\mathrm{K} = y_1, y_2, \ldots, y_n$) or unknowns $u$

$$\mathcal{C} = \mathcal{C}_\mathrm{K} \cup u \tag{16}$$

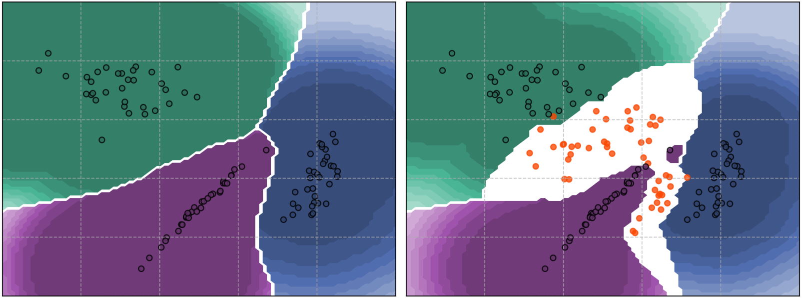
- Learn a likelihood function $h(\boldsymbol{x}; \mathcal{T})$ from the entire training set $\mathcal{T}$

Decision function to predict a label $\hat{y}$ with maximum likelihood:

$$\begin{cases} \hat{y} & \max_{y \in \mathcal{C}} h(\boldsymbol{x}; \mathcal{T}) \geq \tau \text{ and } \hat{y} \neq u \\ u & \text{otherwise} \end{cases} \tag{17}$$

# Pseudo Labeling for Open-Set Learning

Idea: introduce pseudo labels for KUCs and learn from modified training set



Open-set decision boundary without KUCs (left) and with consideration of KUCs during training (right)[4]

[4] Koch, T., Riess, C., & Köhler, T. (2023). LORD: Leveraging Open-Set Recognition with Unknown Data. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4386-4396).

# Pseudo Labeling for Open-Set Learning

Single pseudo label (SPL):

- Assumption: all KUC samples form a large background class
- KUCs are treated as a single class with pseudo-label $u$ resulting in the label set:

$$\mathscr{C} = \{\underbrace{y_1, y_2, \ldots, y_n}_{n \text{ KCs}}, u\} \tag{18}$$

- Predicts unknowns directly independent of a decision threshold
- Can be used with any open-set / closed-set backbone to learn likelihood function

# Pseudo Labeling for Open-Set Learning

## Multi pseudo label (MPL):

- Assumption: every KUC sample models a different class
- KUCs are treated as separate classes with a single sample per pseudo label resulting in the label set:

$$\mathscr{C} = \{\underbrace{y_1, y_2, \ldots, y_n}_{n \text{ KCs}}, \underbrace{y_{n+1}, y_{n+2}, \ldots y_{n+m}}_{m \text{ KUCs}}\} \tag{19}$$

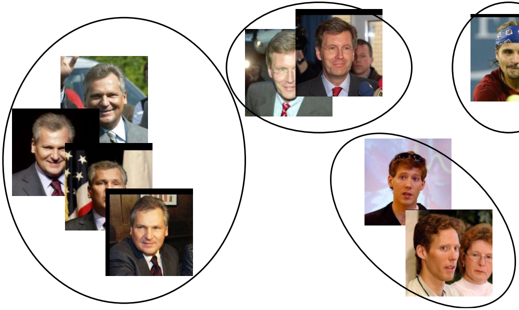
- Prediction of any pseudo label is mapped to the unknown class
- Can be used with any open-set / closed-set backbone to learn likelihood function but limited due to computational complexity

# Face Clustering

- Given $n$ face images $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, cluster them into $k \leq n$ clusters $\mathscr{C}_1, \dots, \mathscr{C}_k$
- Minimize with cluster centers $\boldsymbol{\mu}_i$ and distance measure $d(\cdot, \cdot)$:

$$(\mathscr{C}_1, \dots, \mathscr{C}_k) = \mathrm{argmin}_{\mathscr{C}_1, \dots, \mathscr{C}_k} \sum_{i=1}^{k} \sum_{\boldsymbol{x} \in \mathscr{C}_i} d(\boldsymbol{x}, \boldsymbol{\mu}_i) \tag{20}$$

- $k$-means clustering: $d(\boldsymbol{x}, \boldsymbol{\mu}_i) = ||\boldsymbol{x} - \boldsymbol{\mu}_i||_2^2$ and $\boldsymbol{\mu}_i \equiv$ cluster mean



## Clustering using $k$-Means Algorithm

Iterative algorithm:

1. Assignment: assign each face to the cluster with closest center

$$\mathscr{C}_i^t = \left\{ \boldsymbol{x} : ||\boldsymbol{x} - \mu_i||_2^2 \leq ||\boldsymbol{x} - \mu_j||_2^2 \text{ for all } i \neq j \right\} \tag{21}$$

2. Update: re-calculate cluster centers from current assignment $\mathscr{C}_1^t, \ldots, \mathscr{C}_k^t$

$$\mu_i^{t+1} = \frac{1}{|\mathscr{C}_i^t|} \sum_{\boldsymbol{x} \in \mathscr{C}_i^t} \boldsymbol{x} \tag{22}$$

# Evaluating Face Recognition Systems

Face identification with closed-set protocol:

- Rank of a probe image $\boldsymbol{x}_{\text{probe}}$ for gallery $\mathscr{G}$ with true matching image $\boldsymbol{x}_{\text{match}}$:

$$\text{Rank}(\boldsymbol{x}_{\text{probe}}) = \left| \left\{ \boldsymbol{x}' \in \mathscr{G} \ : \ s(\boldsymbol{x}', \boldsymbol{x}_{\text{probe}}) \geq s(\boldsymbol{x}_{\text{match}}, \boldsymbol{x}_{\text{probe}}) \right\} \right| \tag{23}$$

  $\text{Rank}(\boldsymbol{x}_{\text{probe}}) = 1$ if $\boldsymbol{x}_{\text{probe}}$ is correctly associated with $\boldsymbol{x}_{\text{match}}$

- Rank-$k$ Identification rate on a test set $\mathscr{T}$:

$$\text{IR}(r) = \frac{\left| \left\{ \boldsymbol{x}' \in \mathscr{T} \ : \ \text{Rank}(\boldsymbol{x}') \geq r \right\} \right|}{\left| \mathscr{T} \right|} \tag{24}$$

  For $r = 1$ it is equivalent to the accuracy

# Evaluating Face Recognition Systems

Face identification with open-set protocol:

- Trade-off between true identifications and unknowns that are incorrectly detected as knowns (false alarms) depending on similarity threshold $\theta$
- Detection and identification rate for test set of knowns $\mathscr{K}$:

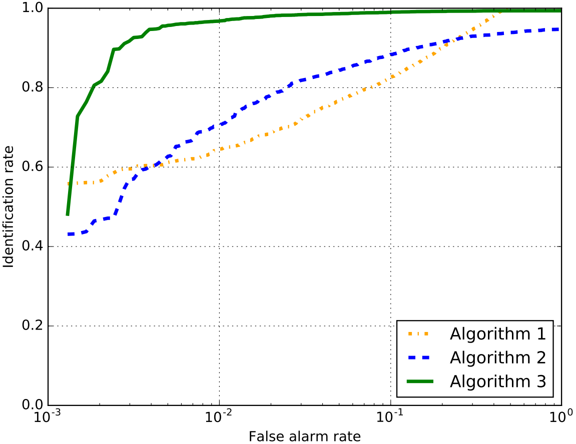
$$\mathrm{DIR}(\theta) = \frac{\left| \{ \boldsymbol{x}' \in \mathscr{K} \; : \; s(\boldsymbol{x}', \boldsymbol{x}_{\text{gallery}}) \geq \theta \text{ and } \mathrm{Rank}(\boldsymbol{x}') = 1 \} \right|}{\left| \mathscr{K} \right|} \tag{25}$$

- False alarm rate for complementary test set of unknowns $\mathscr{U}$:

$$\mathrm{FAR}(\theta) = \frac{\left| \{ \boldsymbol{x}' \in \mathscr{U} \; : \; s(\boldsymbol{x}', \boldsymbol{x}_{\text{gallery}}) \geq \theta \text{ for any } \boldsymbol{x}_{\text{gallery}} \in \mathscr{G} \} \right|}{\left| \mathscr{U} \right|} \tag{26}$$

# DIR Curve for Comparison of Face Recognition Algorithms

Depict $\mathrm{DIR}(\theta)$ at different $\mathrm{FAR}(\theta)$ with semi-logarithmic axes:



# Summary

# Take Home Messages

- Widespread application domains of face recognition
- Still a hard problem under uncontrolled conditions (e. g. difficult poses)
- Face representation is a key component for modern face recognition systems
  - Learned on large training datasets
  - Different methodologies: Eigenfaces, Fisherfaces, deep features
- Recognition tasks (verification, identification, clustering) solved by common machine learning algorithms
  - Based on suitable face representation
  - Today also applicable on embedded devices

# Further Readings

Overview on face image analysis and recognition techniques:

Anil K. Jain and Stan Z. Li. "Handbook of face recognition". Springer, 2011

Eigenfaces, Fisherfaces, and other classical methods:

Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection." IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 1997, 711-720.

Deep learning based methods:

- Yaniv Taigman *et al.* "Deepface: Closing the gap to human-level performance in face verification." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015
- Yandong Wen *et al.* "A discriminative feature learning approach for deep face recognition." European Conference on Computer Vision, 2016.

Thanks for listening.
**Any questions?**