Eigenfaces for Recognition

Instructor: Stephanie Brandl

Andre Jansen, Sourabh Raj | Machine Learning

Structure

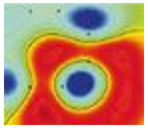
– Introduction

– Our paper

– Our implementation

– Advantages and disadvantages

– Conclusion

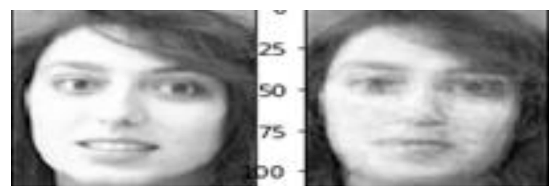


Introduction

– Fast and simple way to face recognition, which performs well under certain circumstances.

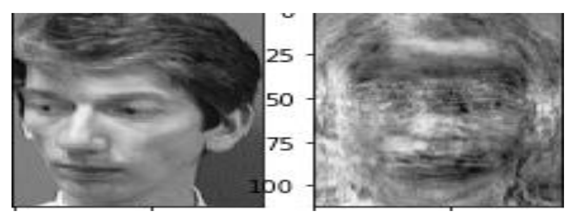
– If trained :

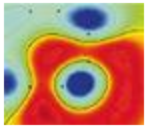
Recognition



– If not trained :

Recognition



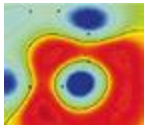


Our Paper – Eigenfaces Recognition Procedure

- Collect a set of characteristic face images, this set should include images with varying expressions

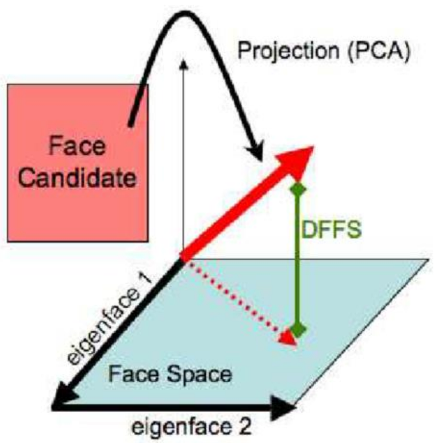
- Calculate the eigenvectors and eigenvalues, and choose K eigenvectors with largest associated eigenvalues.

$$\mathbf{u}_l = \sum_{k=1}^{M} \mathbf{v}_{lk} \mathbf{\Phi}_k,$$



Our Paper – Eigenfaces Recognition Procedure

– For each known individual, calculate the face class vector Ω_k and then choose thresholds, θ, one defining minimum allowable distance from the face class and other from face space.



Our Paper – Eigenfaces Recognition Procedure

- Calculate the pattern vector Ω, the distance to each known class, and the distance to face space.

$$\epsilon_k^2 = \|(\Omega - \Omega_k)\|^2 \qquad \epsilon^2 = \|\Phi - \Phi_f\|^2$$

- Pick one test image and project the image onto face space after centring the and normalizing the image. Reconstructed Non-face pictures can sometimes look like a face.

Technische Universität Berlin

Our Paper – Eigenfaces Recognition Procedure

- If the new image is classified as unknown individual, then eigenfaces may be recalculated by adding the new picture in the set of familiar faces. This enlarges the scope of face recognition

Notes : Calculating eigenfaces can be very expensive for the large set of data so preferably should be done on an offline system. It is sometimes preferable to use SVD as it is computationally efficient.

Issues:

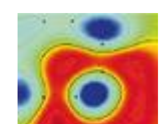
– **Background**: The background of image can have a significant effect on the face recognition accuracy and performance.

– **Scaling**: Size of face/head can have striking effect on the algorithm accuracy.

– **Orientation**: A non-straight face can result into a very wrong recognition.

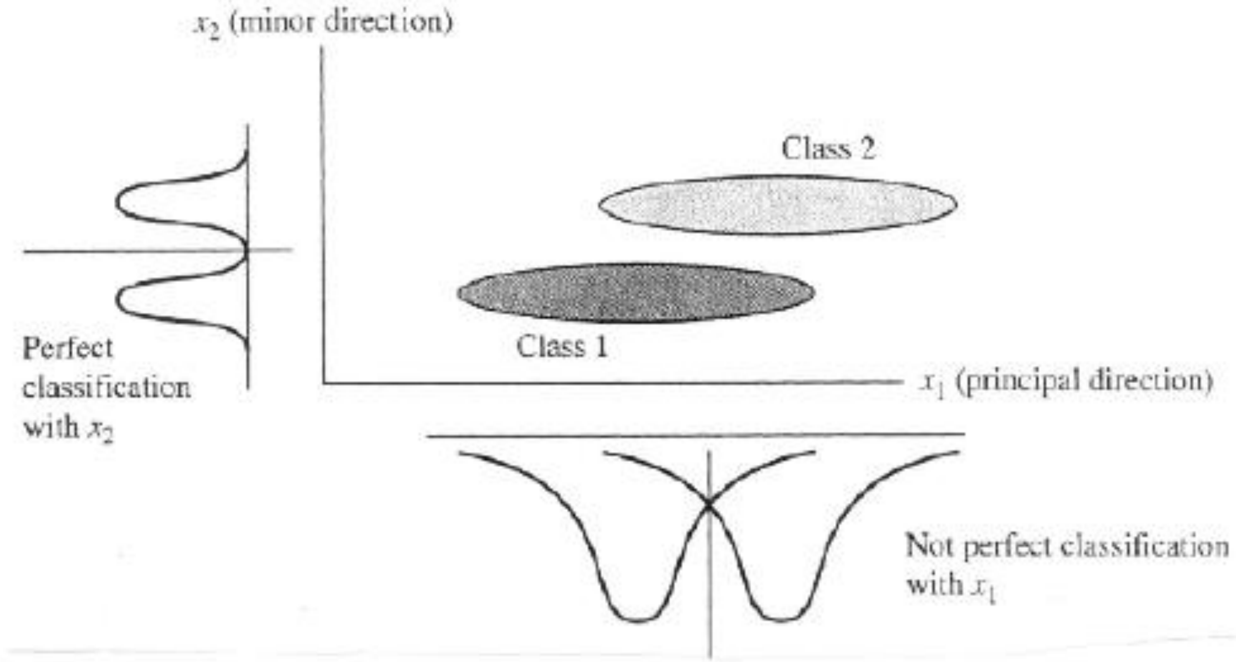
– **Distribution in Face space**: We assume that the distribution is Gaussian, we rather want to characterize it.

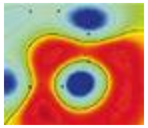
– Multiple view is not supported by the algorithm yet and is prone to misalignment.



Issues: Cont..

PCA is not always an optimal dimensionality-reduction procedure for classification :-





Experiments:

Static Recognition: 2500 Images, 16 Person and using 6 level gaussian distribution pyramid.

Varying lighting	96% correct classification
Head size and scaling	85% correct classification
Head orientation	64% correct classification

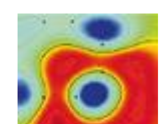
Real-Time Recognition: A camera fixed in a room and a Datacube image processing system.

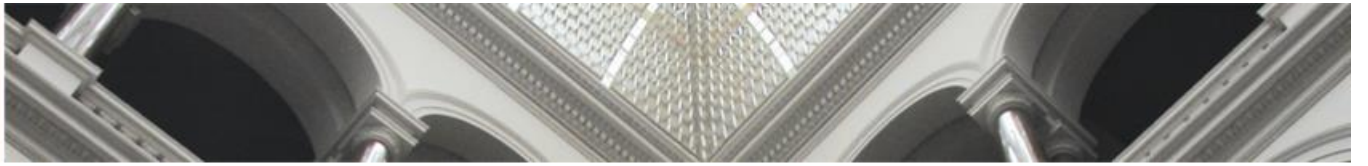
– Detect a moving object, identify the head and when head is found run the algorithm.

Analysis:

– The performance mainly depends upon the relation between pixels, in case of head scaling the correlation between pixels get lost.

– The threshold value also have dramatic effect on the performance.



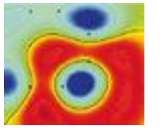


Relation to Neural Network:

Biological Motivation:

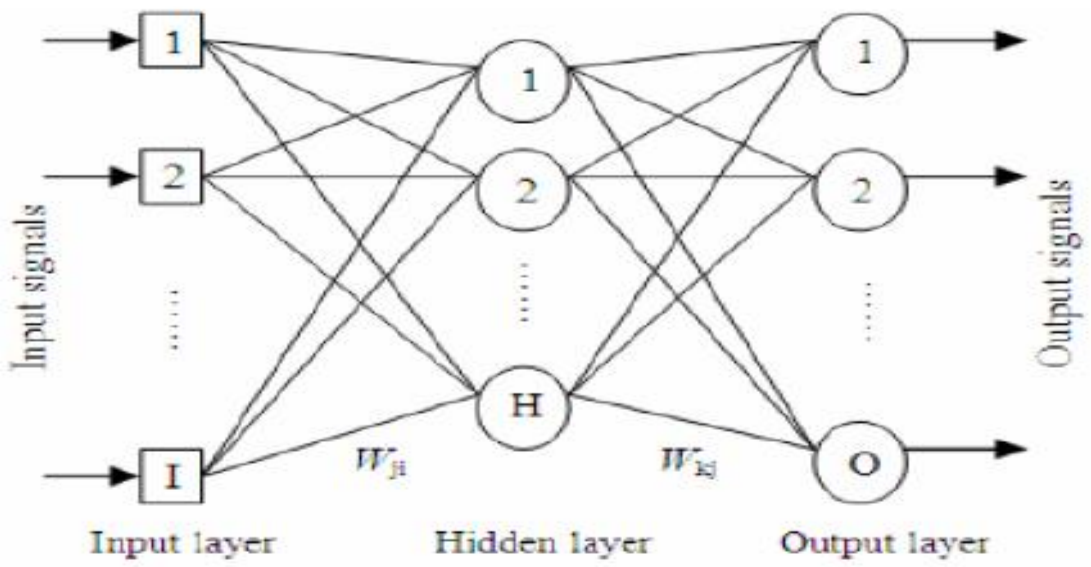
- Early development makes it appear as that biological face recognition mechanism may also be based on some low-level, two-dimensional image processing.

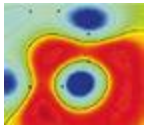
- Eigenfaces approach somewhere qualitatively similar to biological system.
  1. **Change in face:** Even humans have trouble recognising a face after changes.
     - Gradual changes like aging can be dealt with re-calculation of eigenfaces.
     - Quick changes like growing a beard have to proven fooling humans as well.
  2. **Lighting effects:** Recognizing a face in dark is tricky for humans as well.



Relation to Neural Network: Cont..

Neural Networks: Eigenfaces approach can be implemented using parallel computing elements in the form of three layer liner network.



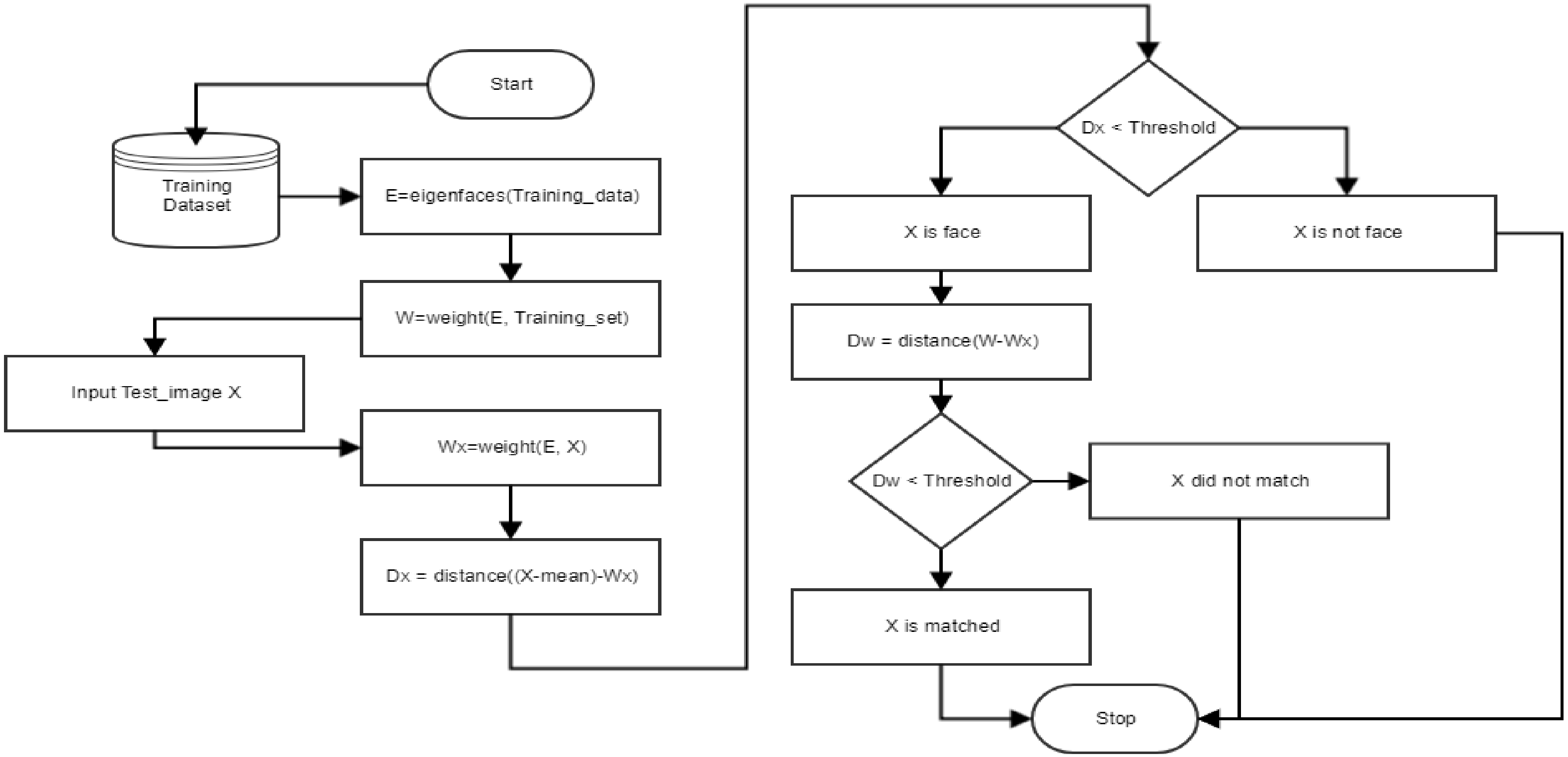


Our Implementation

Our face recognition implementation:

- As described by steps in the paper (shown before)
- Images used are read from training folder, or test folder as needed
- Eigenfaces are calculated using singular value decomposition

Our Implementation – Flow Chart



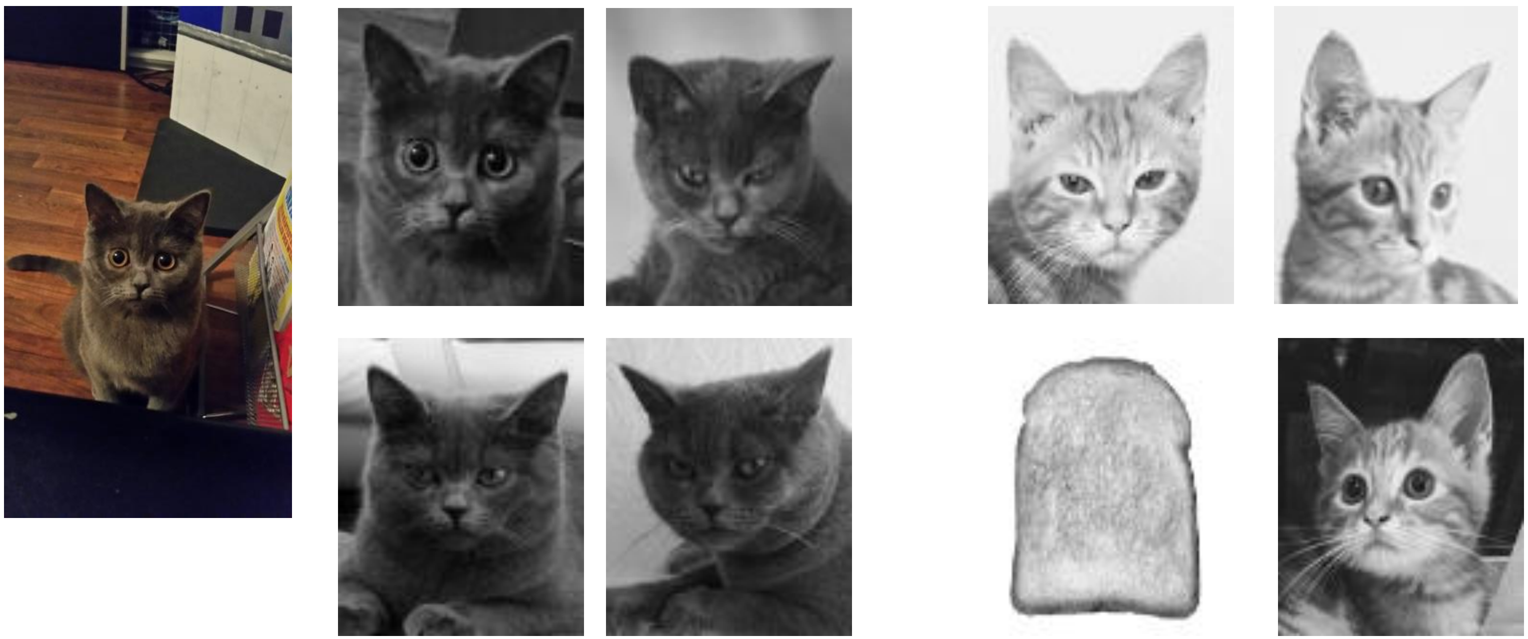
Our Implementation – Training image set

We used following face databases, made available by the Open Source Computer Vision Library (OpenCV):

- AT&T Facedatabase, sometimes “ORL Database of Faces”. Consists of 40 subjects, 10 images each. Same background, sometimes with glasses, eyes closed or smiling. Lighting is not similar. (credit to “AT&T Laboratories, Cambridge)

- The extended Yale Face Database B consists of 28 subjects, with 9 poses and 64 different lighting conditions.

Our Implementation – Training image set



Our Implementation – Testing image set

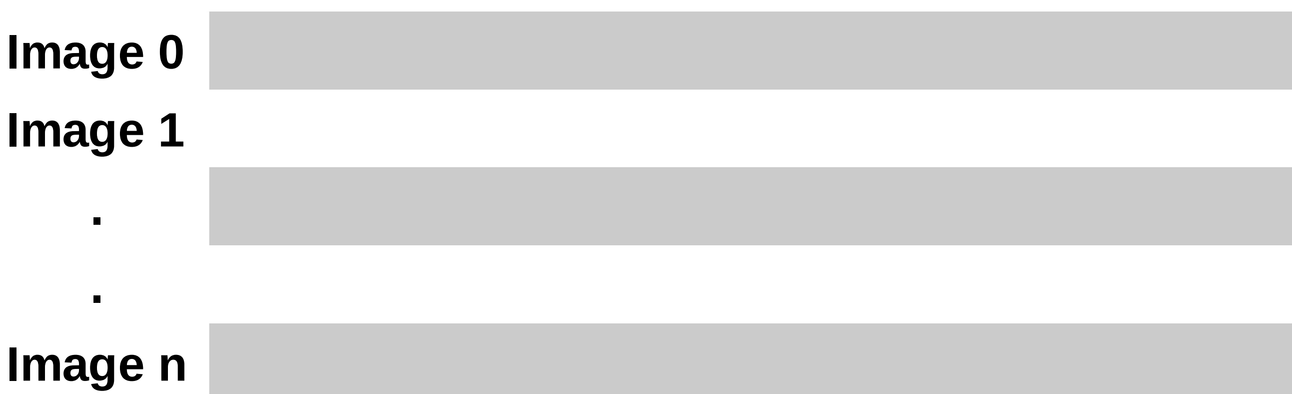
Test image set contains the images which are to be identified by implementation:

- Selected images from training image set

- Additional images to see how implementation performs from the web.

Our Implementation (Data matrix)

- Each row of data matrix consists of flattened image from the training image set
- Data matrix visualization:



Our Implementation (Singular value decomposition)

Principal component analysis with singular value decomposition:

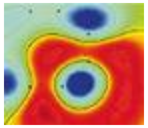
- Computationally more efficient then classic way of eigendecomposition
- The average face has been calculated and subtracted from data matrix before svd
- Function in python for singular value decomposition we used is:
  - np.linalg.svd(D), where D is our data matrix
- Returns eigenvectors as well as singular values

Our Implementation (Determine “k” highest eigen values)

– Performing PCA on the input Data matrix gives us K orthonormal vectors which best describes the distribution of data.

$$\boldsymbol{\lambda}_k = \frac{1}{M}\sum_{n=1}^{M}(\mathbf{u}_k^T\boldsymbol{\Phi}_n)^2$$

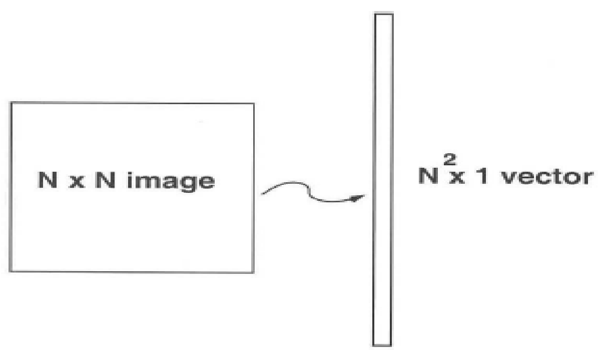
– These K orthonormal vectors are the in ascending order in respective of their magnitude.



Our Implementation (Project to face space)

- Each eigenvector define the subspace of face image, which we call face space.

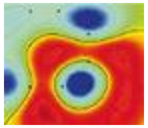
- Each vector describes N by N image and is linear combination of the original face.



Our Implementation (Face identification)

A test image is recognized as a face by the following:

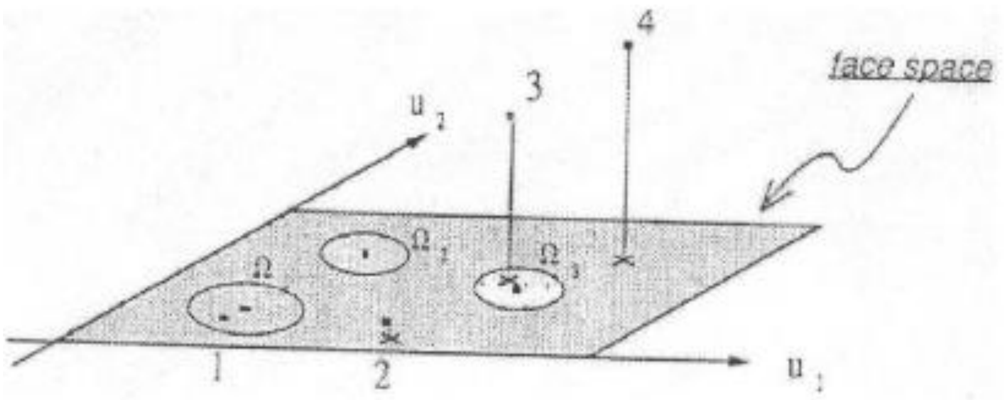
- First the closest match for test image to training image is found
- This is done by calculating the distance between images
- Then by using the determined threshold it is checked whether:
    - The test image is a face
    - The test image has been matched to a face in training image set



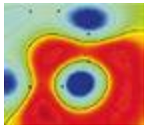
Our Implementation (Face identification): Cont…

Face recognition follows below criteria to match a face.

- IN face *space* AND close to a given face
- IN face space but NOT close to any given face
- NOT in face space AND close to a given face
- NOT in face space and NOT close to any given face

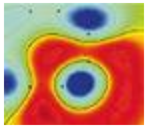


Demo



Advantages

- Does not require complex three-dimensional models
- Relatively simple way of face characterization
- Performs well if provided images meet certain requirements



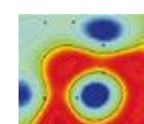
Disadvantages

As shown before, when addressing the issues:

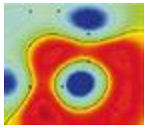
- Does not take into account three-dimensionality of human faces
- Does not work if background varies strongly, lighting is bad etc.

Conclusion

- Eigenfaces algorithm is not a perfect solution to face recognition problem but is quite well fitted. However, In terms of robustness and accuracy, still a lot of work is needed to be done.
- Some suggested enhancement may look like follows:
  - Centring the face using 2-dimensional gaussian window might be used to tackle the background problem by diminishing the background and accentuating the middle face.
  - Rescaling the head size to fit the eigenface size can deal with scaling inaccuracy.
  - An accurate estimation of head orientation may be helpful to align the head in upright view and can benefit the algorithm.
  - Rather than assuming the face space a gaussian distribution, characterization should be done.
  - Face class should be defined for each known person’s characteristic multiple view.



Thank you for listening!



Template (PLEASE KEEP THIS UNTIL END)

Fließtext Blindtext arcilis ipsuntis mil il **Hervorhebung im Fließtext** cupti qui blandig endella borehent, aliciae labores molor sum fugiate cullatusam:

- Kipitium quae quiatus andandunt landio et aut mi
- Neseque volenisci vellacearit harum velestis reicipis as ipsum quodicipsunt velectur
- Sunt alisinvel essi nia quam quid
- Eum dolupta turit, con nobis et volore, opta con corro doluptas dolupis doluptas dolu
- Ventore ntiusam aspelignim
- Volore dolecta turibus, quibea consequo experit aspero velluptas nos conseri orempos
- Ad milliassi solorio molupta pe inciatiis eniet earu

Hervorhebung als Fazit Blindtext suntis mil il moloribus cupti qui blandig endella borehent.