POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

POLITECNICO MILANO 1863
NECST laboratory

Quantum Eigenfaces:

Linear Feature Mapping and Nearest Neighbor Classification with Outlier Detection

Armando Bellante, William Bonvini, Stefano Vanerio, Stefano Zanero

IEEE International Conference on Quantum Computing and Engineering, Bellevue, WA, 21st September 2023

Overview

Quantum Eigenfaces-based classification w/ provable running time

Motivation:

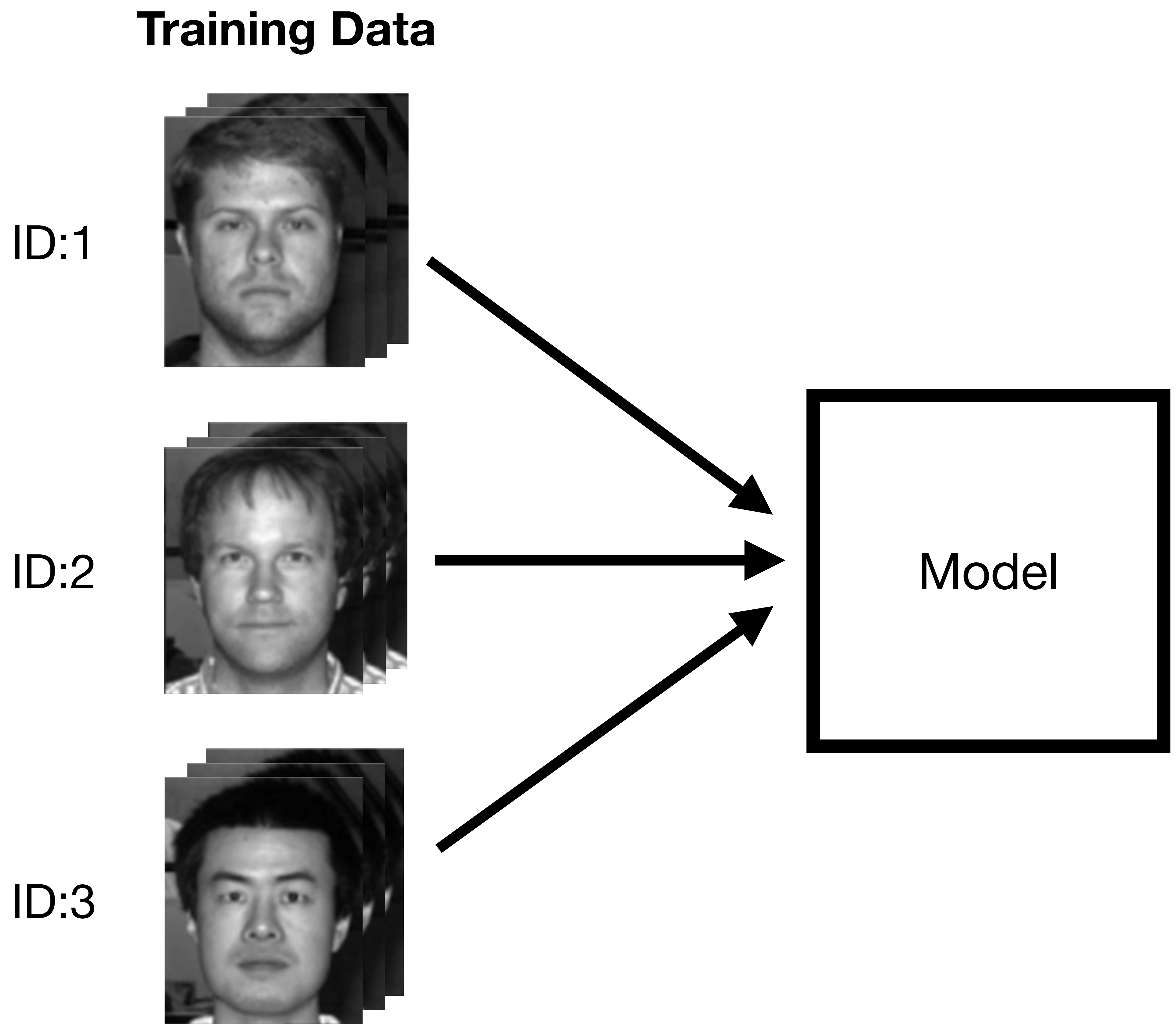
- Eigenfaces is a milestone in Computer Vision and Machine Learning;
- More generally, it is a linear feature mapping + nearest neighbor/centroid classification with outlier detection.

[1] Turk, Matthew, and Alex Pentland. "Eigenfaces for recognition." Journal of cognitive neuroscience 3.1 (1991): 71-86.
[2] Turk, Matthew A., and Alex P. Pentland. "Face recognition using eigenfaces." Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition. IEEE Computer Society, 1991.
Related work on nearest neighbor/centroid:
Seth Lloyd et al. - arXiv:1307.0411 (2013)
Nathan Wiebe et al. - Quantum Information & Computation 15.3-4 (2015): 316-356

Eigenfaces

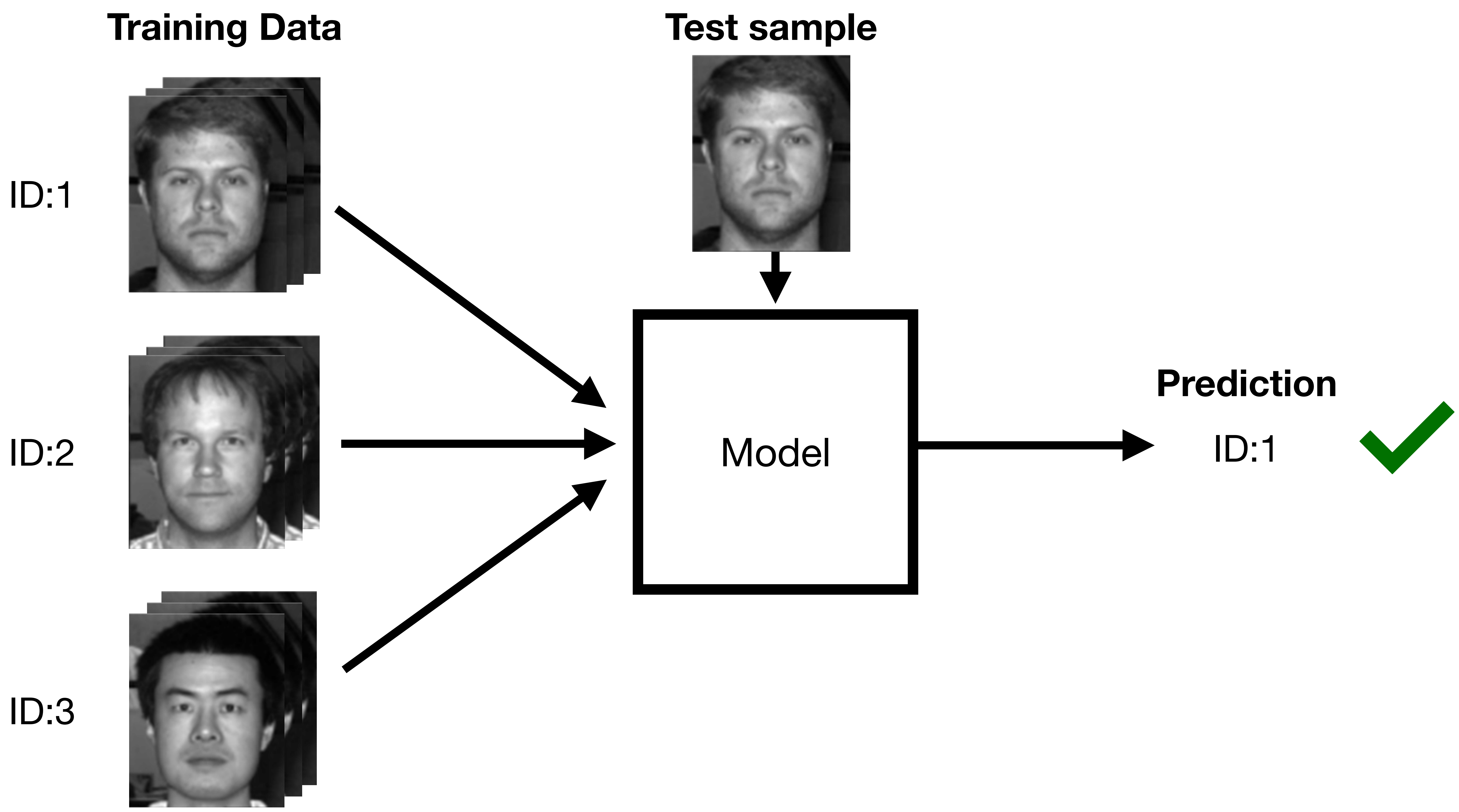
Eigenfaces for face recognition

Algorithm's overview



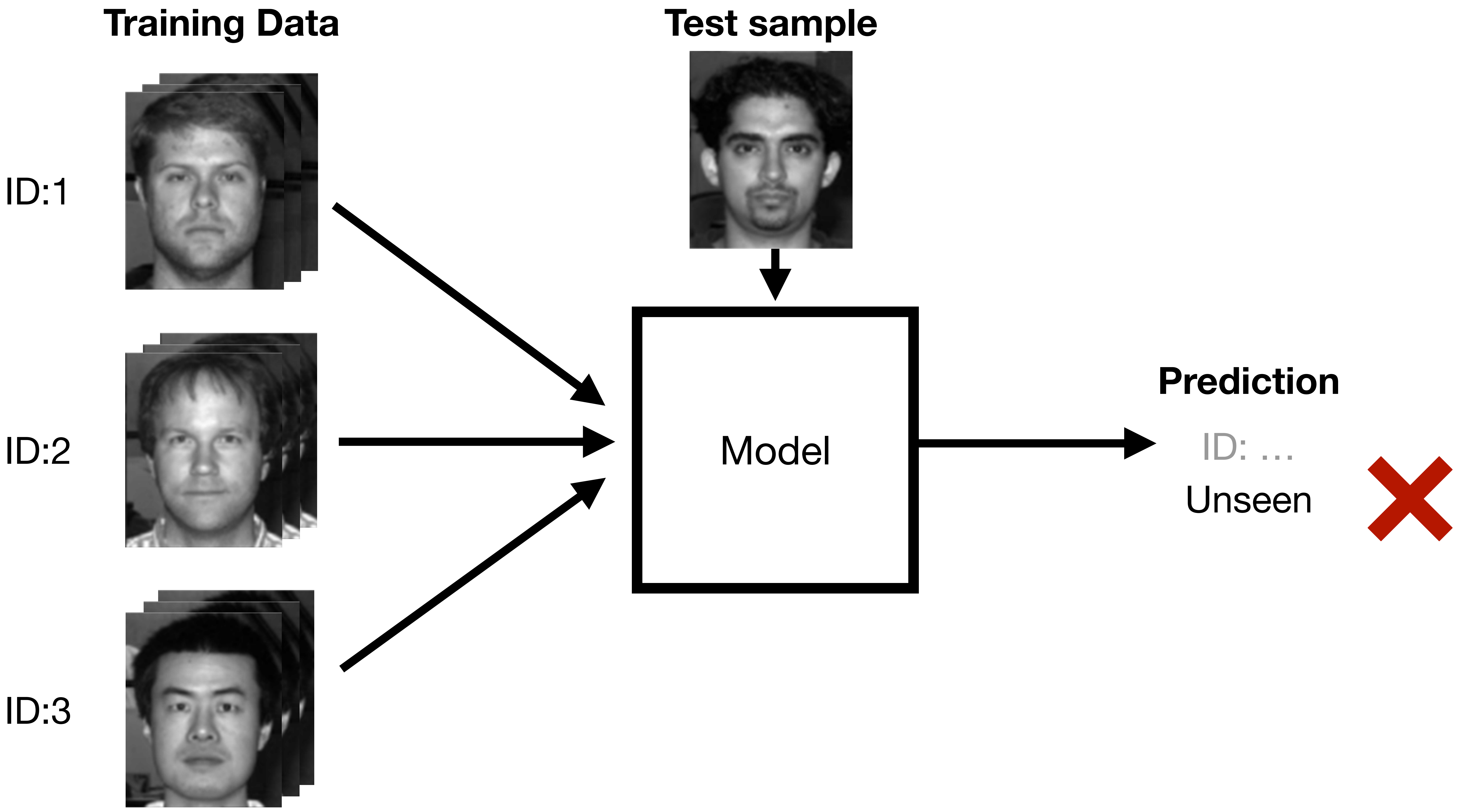
Eigenfaces for face recognition

Algorithm's overview



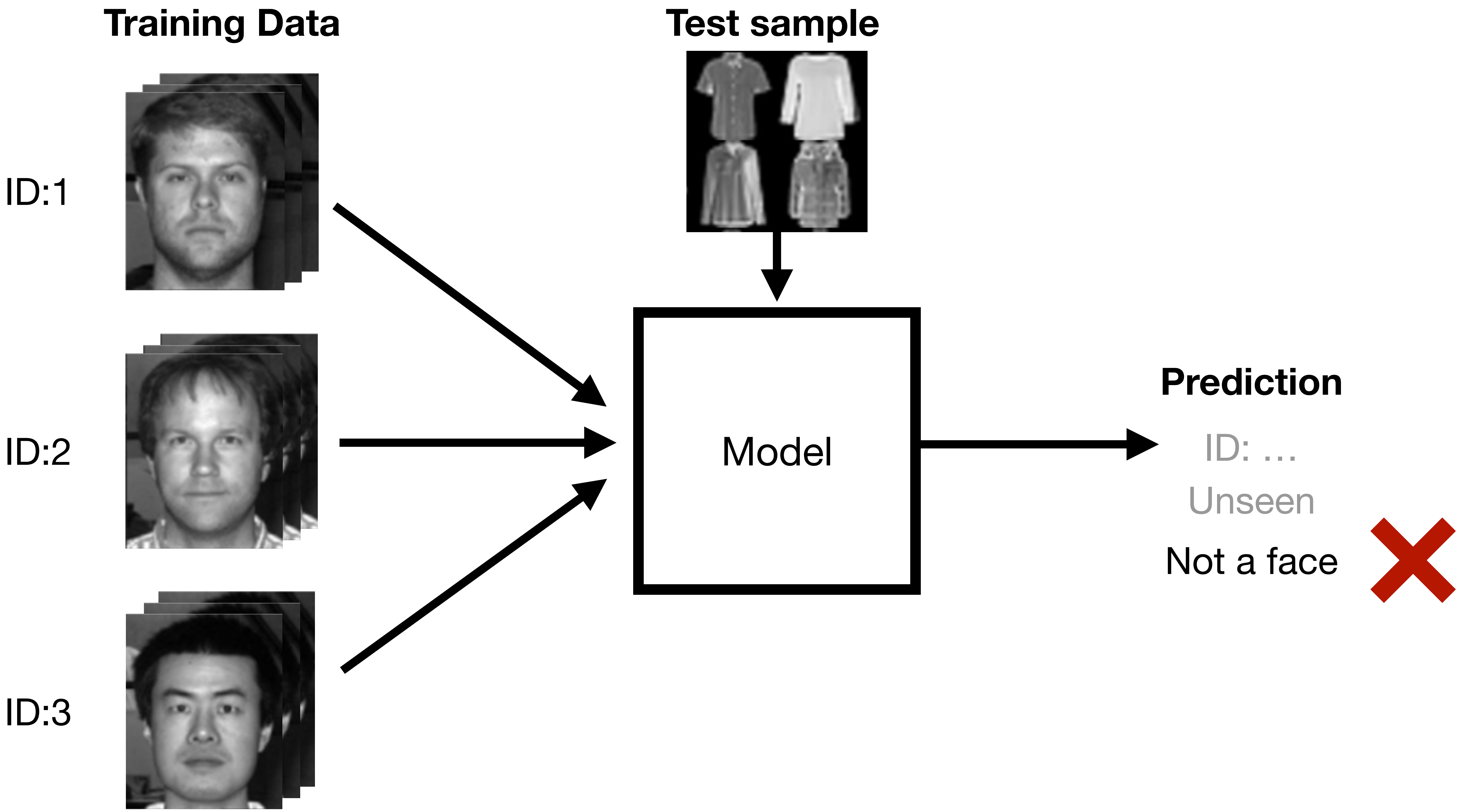
Eigenfaces for face recognition

Algorithm’s overview



Eigenfaces for face recognition

Algorithm's overview



Eigenfaces for face recognition

Training the model

Training Data

128x128 = 16384

$\vec{x}_i \in [0,255]^{16384}$

Faces will not be distributed uniformly in this space

Eigenfaces for face recognition

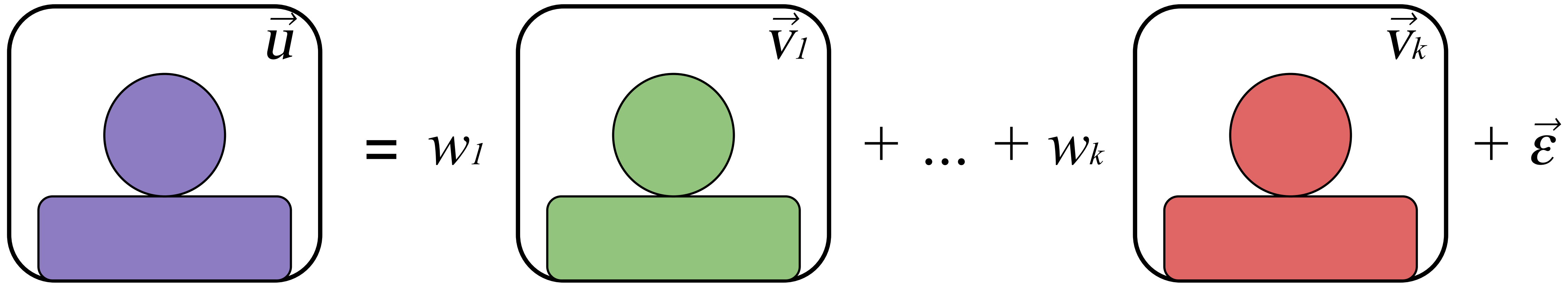
Training the model

Training Data

128x128 = 16384

$\vec{x}_i \in [0,255]^{16384}$

Faces will not be distributed uniformly in this space



We can express a face as a combination of some elementary faces

Eigenfaces for face recognition

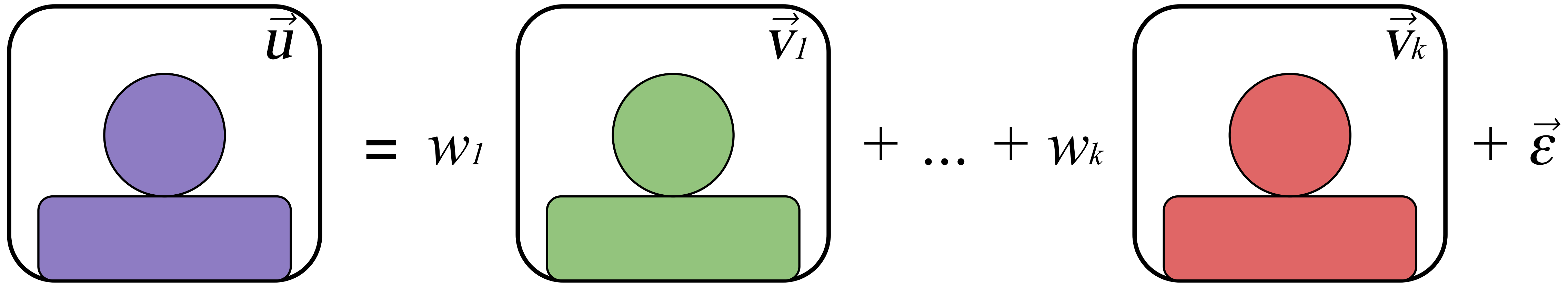
Training the model

Training Data

128x128 = 16384

$\vec{x}_i \in [0,255]^{16384}$

Faces will not be distributed uniformly in this space



We can express a face as a combination of some elementary faces

$$\vec{u}_i = \vec{x}_i - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$$

Center the data

$$\xrightarrow{\text{Select top-k Principal Components}} \{\vec{v}_1, \cdots, \vec{v}_k\}$$

$$\vec{u}_i = \sum_{j\in[k]} w_j\vec{v}_j + \vec{\epsilon} = V^T\overrightarrow{w}_i + \vec{\epsilon}$$

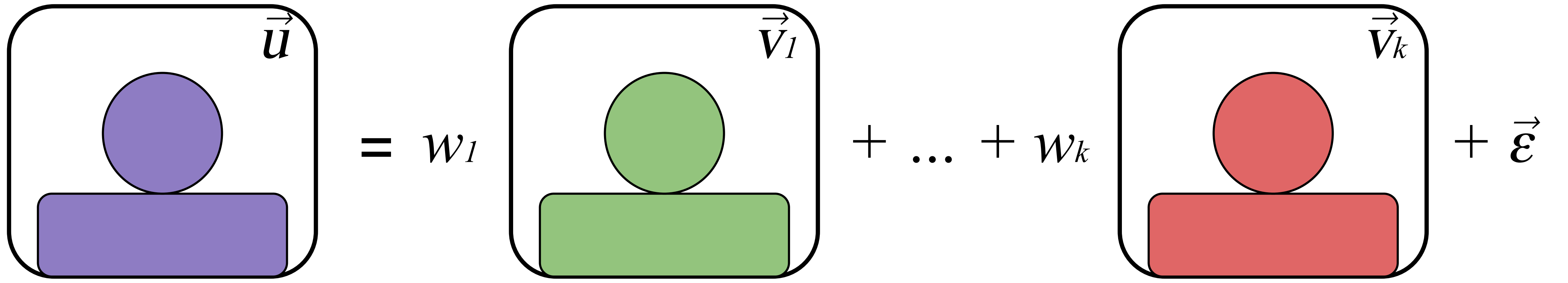
Eigenfaces for face recognition

Training the model

Training Data

128x128 = 16384

$\vec{x}_i \in [0{,}255]^{16384}$



We can express a face as a combination of some elementary faces

Faces will not be distributed uniformly in this space

$$\vec{u}_i = \vec{x}_i - \frac{1}{N}\sum_{j\in[N]}\vec{x}_j \xrightarrow{\text{Select top-k Principal Components}} \{\vec{v}_1, \cdots, \vec{v}_k\}$$

Center the data

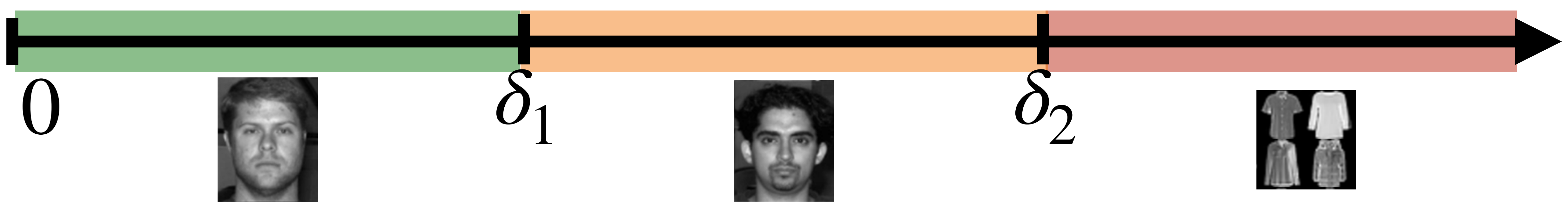
$$\vec{u}_i = \sum_{j\in[k]} w_j\vec{v}_j + \vec{\epsilon} = V^T\underbrace{\vec{w}_i}_{\vec{c}_j \in \mathbb{R}^k} + \vec{\epsilon}$$

Eigenfaces for face recognition

Classification

Algorithm 1 Eigenfaces-based classification

1: Center the data point:
$$\vec{u} = \vec{x} - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$$
2: Compute the weights vector:
$$\vec{w} = V\vec{u}$$
3: Select the minimum distance between w and the stored weights:
$$d = \min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$$
4: Save the index of the closest weights vector:
$$j^* = \arg\min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$$
5: Output: if $d \leq \delta_1$: *same class of* $\vec{c}_{j^*}$, *output* y_{j^*};
if $\delta_1 < d < \delta_2$: *similar element, output* -1.
if $d \geq \delta_2$: *not a similar element, output* -2.

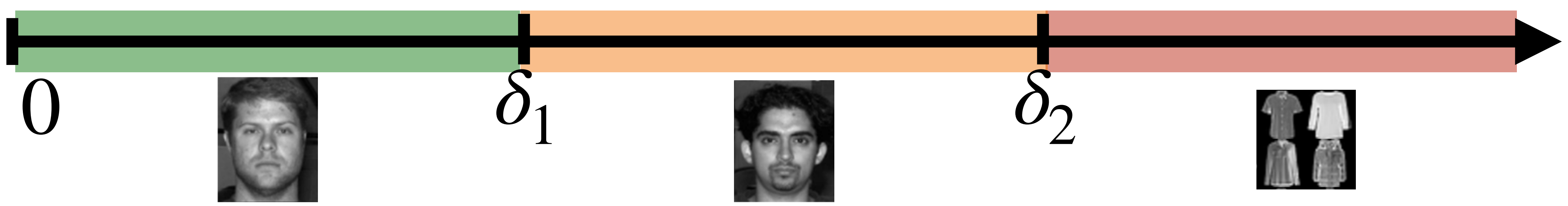


Eigenfaces for face recognition

Classification

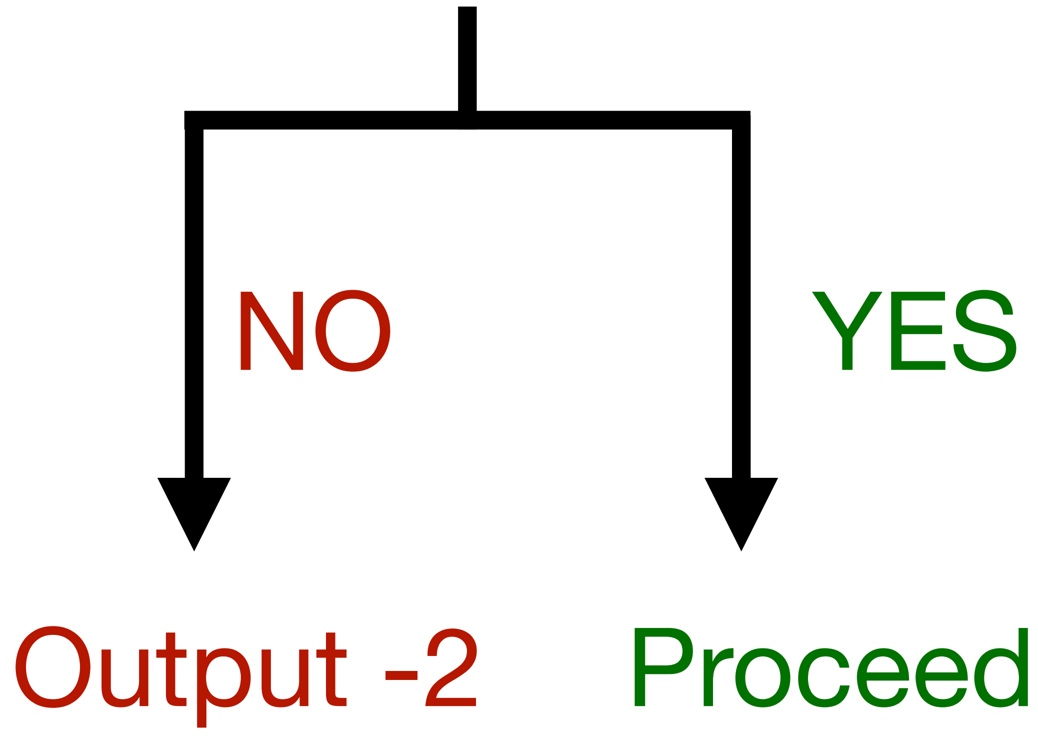
Algorithm 1 Eigenfaces-based classification

1: Center the data point:
   $\vec{u} = \vec{x} - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$
2: Compute the weights vector:
   $\vec{w} = V\vec{u}$
3: Select the minimum distance between w and the stored weights:
   $d = \min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$
4: Save the index of the closest weights vector:
   $j^* = \arg\min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$
5: Output: if $d \leq \delta_1$: *same class of* $\vec{c}_{j^*}$, *output* y_{j^*};
   if $\delta_1 < d < \delta_2$: *similar element, output* -1.
   if $d \geq \delta_2$: *not a similar element, output* -2.



Norm-based Outlier detection

$$\frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} > \gamma$$

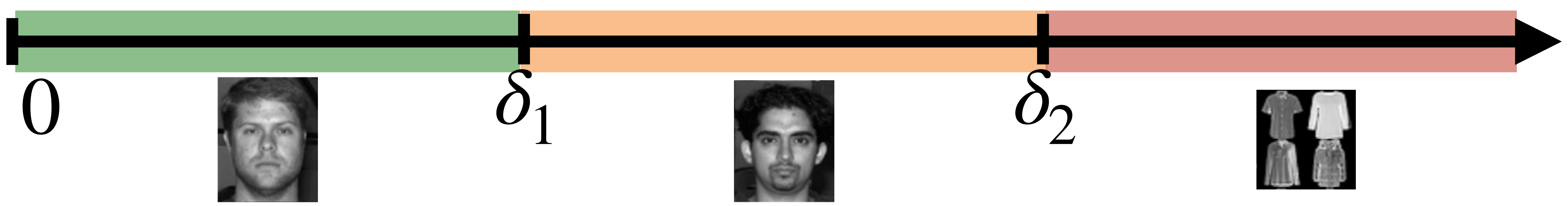


Eigenfaces for face recognition

Classification

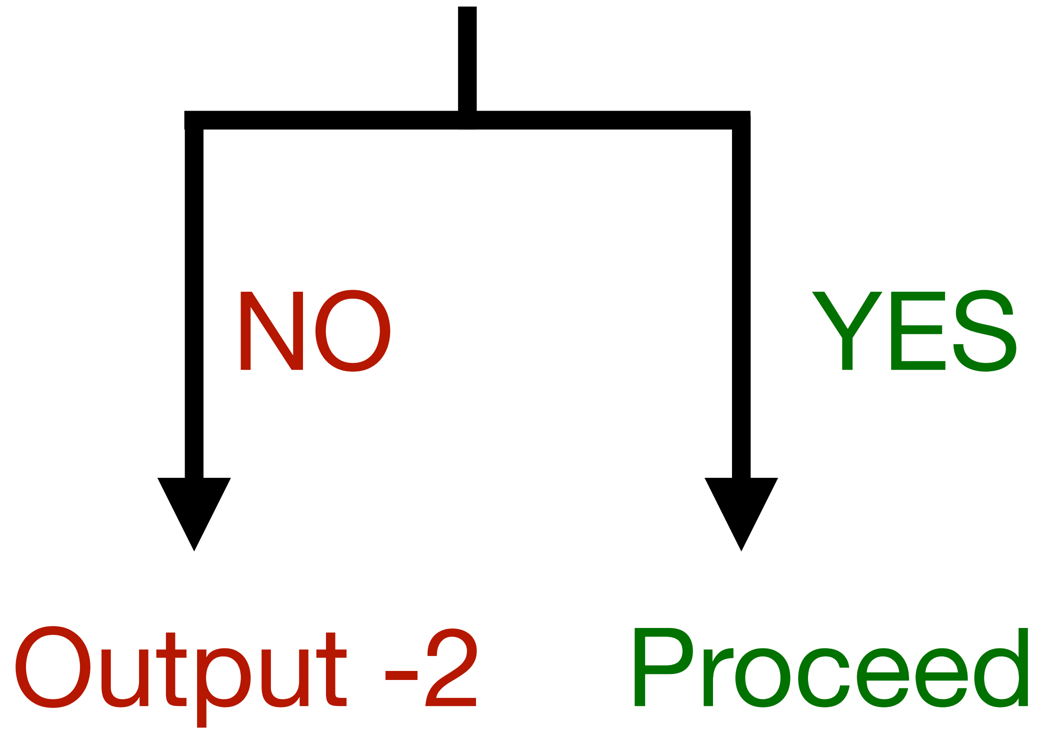
Algorithm 1 Eigenfaces-based classification

1: Center the data point:
$$\vec{u} = \vec{x} - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$$
2: Compute the weights vector:
$$\vec{w} = V\vec{u}$$
3: Select the minimum distance between w and the stored weights:
$$d = \min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$$
4: Save the index of the closest weights vector:
$$j^* = \arg\min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$$
5: Output: if $d \leq \delta_1$: *same class of* $\vec{c}_{j^*}$, *output* y_{j^*};
if $\delta_1 < d < \delta_2$: *similar element, output* -1.
if $d \geq \delta_2$: *not a similar element, output* -2.



Norm-based Outlier detection

$$\frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} > \gamma$$



Textbook implementation $O(mk + pk)$

Quantum algorithm

Quantum Eigenfaces-based classification

Input data

Algorithm 1 Eigenfaces-based classification

1: Center the data point:
$\vec{u} = \vec{x} - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$
2: Compute the weights vector:
$\vec{w} = V\vec{u}$
3: Select the minimum distance between w and the stored weights:
$d = \min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$
4: Save the index of the closest weights vector:
$j^* = \arg\min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$
5: Output: if $d \leq \delta_1$: *same class of* $\vec{c}_{j^*}$, *output* y_{j^*};
if $\delta_1 < d < \delta_2$: *similar element, output* -1.
if $d \geq \delta_2$: *not a similar element, output* -2.

Norm-based Outlier detection

$$\frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} > \gamma$$

Main idea:

Encode $\{\vec{c}_1, \cdots, \vec{c}_p\}$, $\vec{u}$ in quantum states and V in a unitary.

Data access

Required oracles

Quantum access to the centroids/neighbors and their norms

$$U_C : |i\rangle|0\rangle \mapsto |i\rangle|\vec{c}_i\rangle$$

$$U_C' : |i\rangle|0\rangle \mapsto |i\rangle|\|\vec{c}_i\|_2\rangle$$

Quantum access to the (mean centered) test data

$$U_u : |0\rangle \mapsto |\vec{u}\rangle$$

(α, q, ϵ_0)-Block-encoding access to V

$$U_V = \begin{bmatrix} \frac{\overline{V}}{\alpha} & \cdot \\ \cdot & \cdot \end{bmatrix}$$

s.t. $\|V - \overline{V}\|_2 \leq \epsilon_0$

$$\vec{x} = \begin{bmatrix} x_1 \\ \cdots \\ x_N \end{bmatrix} \longmapsto |\vec{x}\rangle = \frac{1}{\|\vec{x}\|_2} \sum_{i \in [N]} x_i |i\rangle$$

Data access

Quantum Random Access Memory

Assumption of a QRAM: Device that performs queries
$U_{QRAM} : |i, 0\rangle \mapsto |i, x_i\rangle$ in superposition, with x_i binary encoded, in time $\widetilde{O}(1)$.

Store classical data

i	x_i
i + 1	x_{i+1}
i + 2	x_{i+2}
i + 3	x_{i+3}
i + 4	x_{i+4}
i + 5	x_{i+5}

Perform quantum queries

$$U_{QRAM} : |i\rangle|0\rangle \mapsto |i\rangle|x_i\rangle$$

$$U_{QRAM} : \sum_{i=0}^{k} \alpha_i |i\rangle|0\rangle \mapsto \sum_{i=0}^{k} \alpha_i |i\rangle|x_i\rangle$$

Vittorio Giovannetti et al. - *Physical Review A*, *78*(5), 052310 - *ITCS 2008*
Connor T. Hang et al. - *PRX Quantum*, *2*(2), 020311 - 2021

Data access

Implementing the oracles via QRAM

Assumption of a QRAM: Device that performs queries
$U_{QRAM} : |i,0\rangle \mapsto |i, x_i\rangle$ in superposition, with x_i binary encoded, in time $\widetilde{O}(1)$.

Store classical data

i	x_i
i + 1	x_{i+1}
i + 2	x_{i+2}
i + 3	x_{i+3}
i + 4	x_{i+4}
i + 5	x_{i+5}

For a matrix $A \in \mathbb{R}^{n \times m}$

- Preparation time/space: $O(nm \log(nm))$
- Query time: $O(\text{polylog}(nm, 1/\epsilon_0)) = \widetilde{O}(1)$

$$U_A : |i\rangle|0\rangle \mapsto |i\rangle|\vec{a}_i\rangle$$

$$U_{A'} : |i\rangle|0\rangle \mapsto |i\rangle|\|\vec{a}_i\|_2\rangle$$

$(\mu(A), \lceil \log(n + m + 1) \rceil, \epsilon_0)$-Block-encoding of A

Iordanis Kerenidis and Anupam Prakash - *ITCS 2017*
Iordanis Kerenidis and Anupam Prakash - *Physical Review A* 101.2 (2020): 022316
Shantanav Chakraborty et al. - ICALP 2019

Quantum Eigenfaces-based classification

Linear mapping

Algorithm 1 Eigenfaces-based classification

1: Center the data point:
$\vec{u} = \vec{x} - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$
2: Compute the weights vector:
$\vec{w} = V\vec{u}$
3: Select the minimum distance between w and the stored weights:
$d = \min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$
4: Save the index of the closest weights vector:
$j^* = \arg\min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$
5: Output: if $d \leq \delta_1$: *same class of* $\vec{c}_{j^*}$, *output* y_{j^*};
if $\delta_1 < d < \delta_2$: *similar element, output* -1.
if $d \geq \delta_2$: *not a similar element, output* -2.

Norm-based Outlier detection

$$\frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} > \gamma$$

Main idea:

Encode $\{\vec{c}_1, \cdots, \vec{c}_p\}$, $\vec{u}$ in quantum states and V in a unitary.

Algorithm 2 Quantum Eigenfaces-based classification

1: Prepare a state proportional to $\vec{w}$:
$|\vec{w}\rangle = \frac{V\vec{u}}{\|V\vec{u}\|}$

Quantum Eigenfaces-based classification

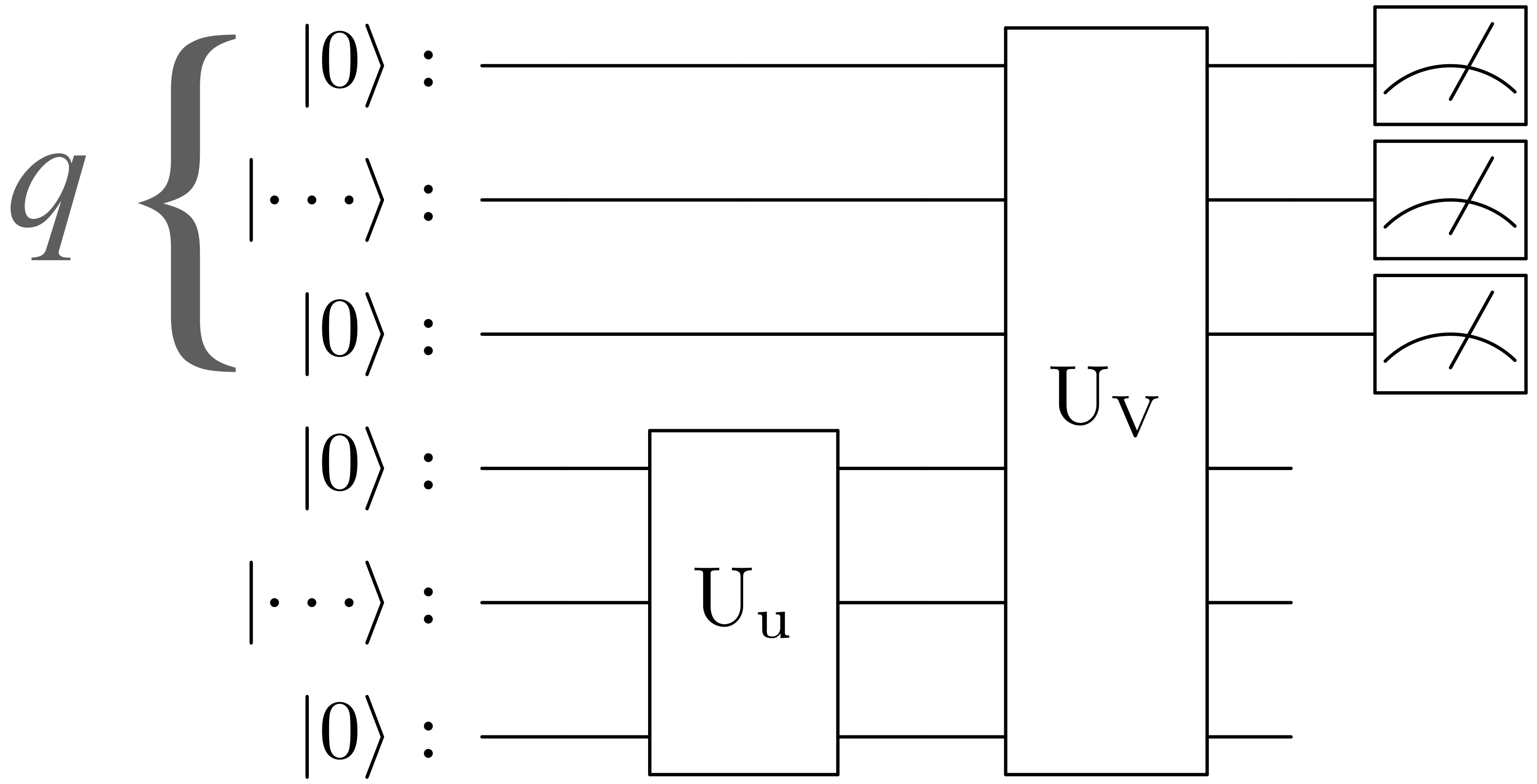
Linear mapping

State preparation for $\vec{u}$

$$U_u : |0\rangle \mapsto |\vec{u}\rangle$$

(α, q, ϵ_0)-Block-encoding of V

$$U_V = \begin{bmatrix} \frac{V}{\alpha} & \cdot \\ \cdot & \cdot \end{bmatrix}$$



$$P(0) = \left(\frac{\|\overline{w}\|_2}{\alpha \|\vec{u}\|_2} \right)^2$$

$$\begin{bmatrix} \frac{V}{\alpha} & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} \vec{u} \\ \cdot \end{bmatrix} = |0\rangle^{\otimes q} \frac{V}{\alpha} |\vec{u}\rangle + |0^{\perp}\rangle$$

Main idea:

Encode $\{\vec{c}_1, \cdots, \vec{c}_p\}$, $\vec{u}$ in quantum states and V in a unitary.

Algorithm 2 Quantum Eigenfaces-based classification

1: Prepare a state proportional to $\vec{w}$:
$|\vec{w}\rangle = \frac{V\vec{u}}{\|V\vec{u}\|}$

Quantum Eigenfaces-based classification

Distance estimation

Algorithm 1 Eigenfaces-based classification

1: Center the data point:
$$\vec{u} = \vec{x} - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$$
2: Compute the weights vector:
$$\vec{w} = V\vec{u}$$
3: Select the minimum distance between w and the stored weights:
$$d = \min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$$
4: Save the index of the closest weights vector:
$$j^* = \arg\min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$$
5: Output: if $d \leq \delta_1$: *same class of* $\vec{c}_{j^*}$, *output* y_{j^*};
if $\delta_1 < d < \delta_2$: *similar element, output* -1.
if $d \geq \delta_2$: *not a similar element, output* -2.

Main idea:

Encode $\{\vec{c}_1, \cdots, \vec{c}_p\}$, $\vec{u}$ in quantum states and V in a unitary.

Algorithm 2 Quantum Eigenfaces-based classification

1: Prepare a state proportional to $\vec{w}$:
$$|\vec{w}\rangle = \frac{V\vec{u}}{\|V\vec{u}\|}$$
2: Perform Euclidean distance estimation in superposition:
$$|\varphi\rangle = \frac{1}{\sqrt{p}}\sum_{j\in[p]} |j\rangle \left|\|\vec{w} - \vec{c}_j\|_2^2\right\rangle$$

Norm-based Outlier detection

$$\frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} > \gamma$$

Quantum Eigenfaces-based classification

Distance estimation

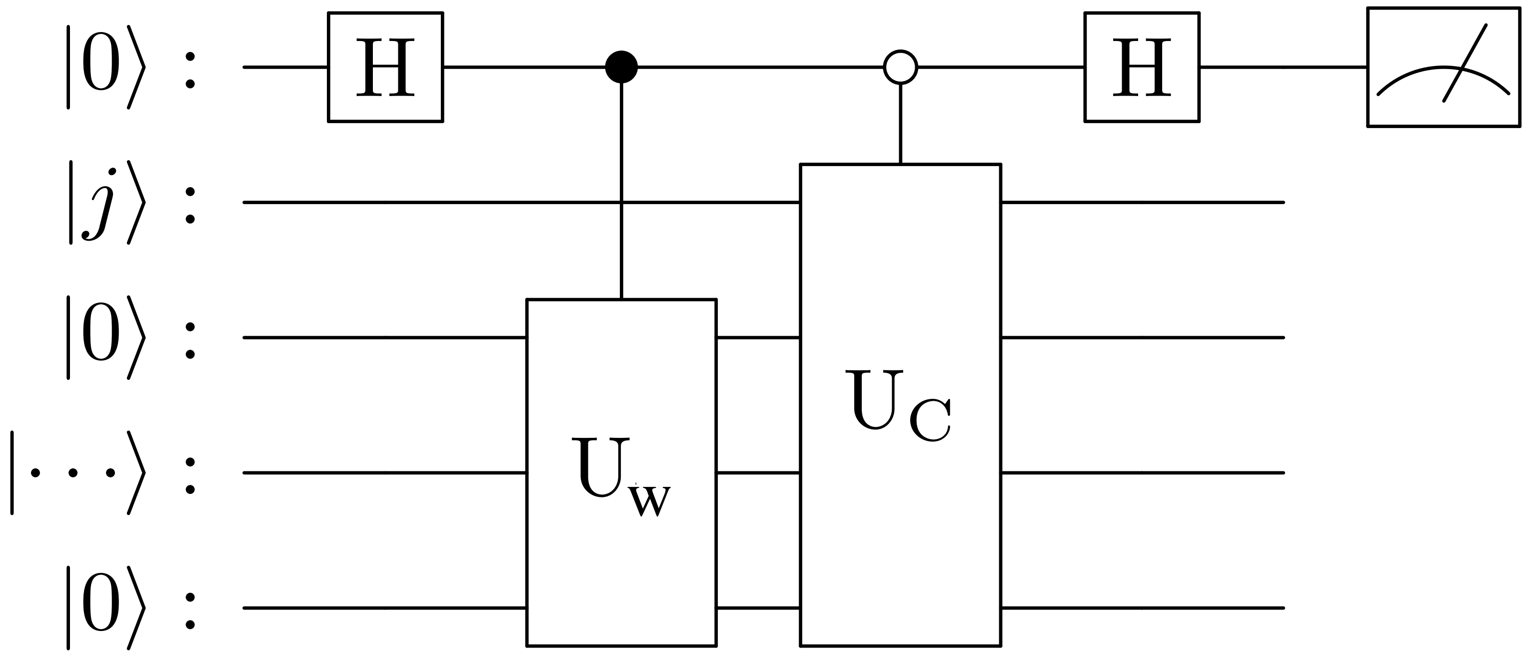
State preparation for $\vec{w}$

$$U_w : |0\rangle \mapsto |\vec{w}\rangle$$

Quantum access to the centroids/ neighbors and their norms

$$U_C : |i\rangle|0\rangle \mapsto |i\rangle|\vec{c}_i\rangle$$

$$U'_C : |i\rangle|0\rangle \mapsto |i\rangle|\|\vec{c}_i\|_2\rangle$$



$$P(1) = \frac{1 - \langle \vec{w} | \vec{c}_i \rangle}{2}$$

$$\|\vec{w} - \vec{c}_j\|_2^2 = \|\vec{w}\|_2^2 + \|\vec{c}_j\|_2^2 - 2\|\vec{w}\|_2\|\vec{c}_j\|_2\langle \vec{w} | \vec{c}_j \rangle$$

Main idea:

Encode $\{\vec{c}_1, \cdots, \vec{c}_p\}$, $\vec{u}$ in quantum states and V in a unitary.

Algorithm 2 Quantum Eigenfaces-based classification

1: Prepare a state proportional to $\vec{w}$:
$$|\vec{w}\rangle = \frac{V\vec{u}}{\|V\vec{u}\|}$$
2: Perform Euclidean distance estimation in superposition:
$$|\varphi\rangle = \frac{1}{\sqrt{p}} \sum_{j\in[p]} |j\rangle \left|\|\vec{w} - \vec{c}_j\|_2^2\right\rangle$$

Quantum Eigenfaces-based classification

Finding the minimum

Algorithm 1 Eigenfaces-based classification

1: Center the data point:
$\vec{u} = \vec{x} - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$

2: Compute the weights vector:
$\vec{w} = V\vec{u}$

3: Select the minimum distance between w and the stored weights:
$d = \min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$

4: Save the index of the closest weights vector:
$j^* = \arg\min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$

5: Output: if $d \leq \delta_1$: *same class of* $\vec{c}_{j^*}$, *output* y_{j^*};
if $\delta_1 < d < \delta_2$: *similar element, output* -1.
if $d \geq \delta_2$: *not a similar element, output* -2.

Main idea:

Encode $\{\vec{c}_1, \cdots, \vec{c}_p\}$, $\vec{u}$ in quantum states and V in a unitary.

Algorithm 2 Quantum Eigenfaces-based classification

1: Prepare a state proportional to $\vec{w}$:
$|\vec{w}\rangle = \frac{V\vec{u}}{\|V\vec{u}\|}$

2: Perform Euclidean distance estimation in superposition:
$|\varphi\rangle = \frac{1}{\sqrt{p}}\sum_{j\in[p]} |j\rangle \left|\|\vec{w} - \vec{c}_j\|_2^2\right\rangle$

3: Find the minimum distance and index using a variant of Grover's search

4: Output: if $\overline{d} \leq \delta_1$: *same class of* $\vec{c}_{j^*}$, *output* y_{j^*};
if $\delta_1 < \overline{d} < \delta_2$: *similar element, output* -1.
if $\overline{d} \geq \delta_2$: *not a similar element, output* -2.

Norm-based Outlier detection

$$\frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} > \gamma$$

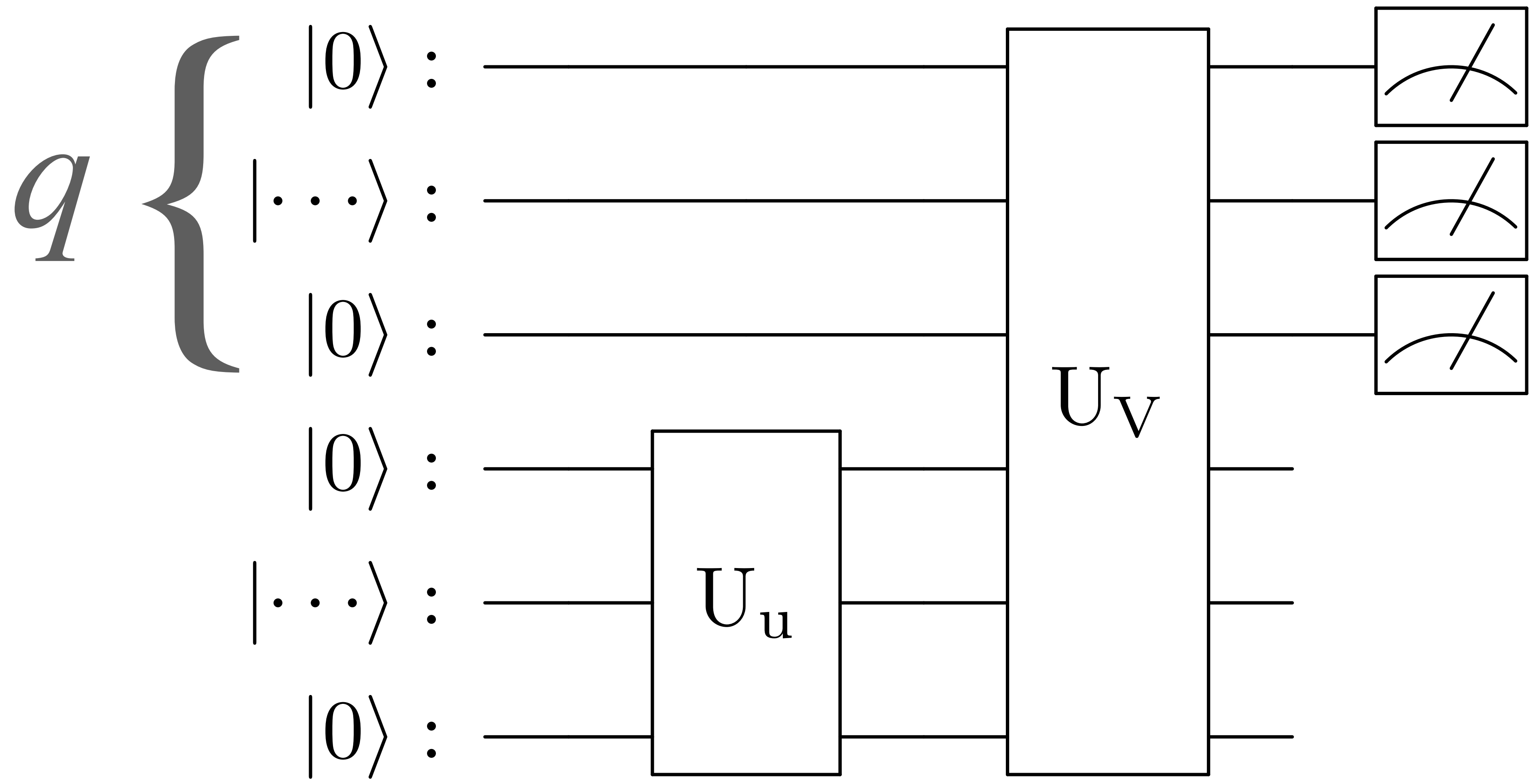
Norm-based outlier detection

Block-encoding + amplitude estimation

State preparation for $\vec{u}$ $\quad$ (α, q, ϵ_0)-Block-encoding of V

$$U_u : |0\rangle \mapsto |\vec{u}\rangle \qquad U_V = \begin{bmatrix} \frac{\overline{V}}{\alpha} & \cdot \\ \cdot & \cdot \end{bmatrix}$$

Estimate $\frac{\|\vec{w}\|}{\|\vec{u}\|}$ and compare it to γ.



$$P(0) = \left(\frac{\|\overline{w}\|_2}{\alpha \|\vec{u}\|_2} \right)^2$$

Norm-based outlier detection

Block-encoding + amplitude estimation

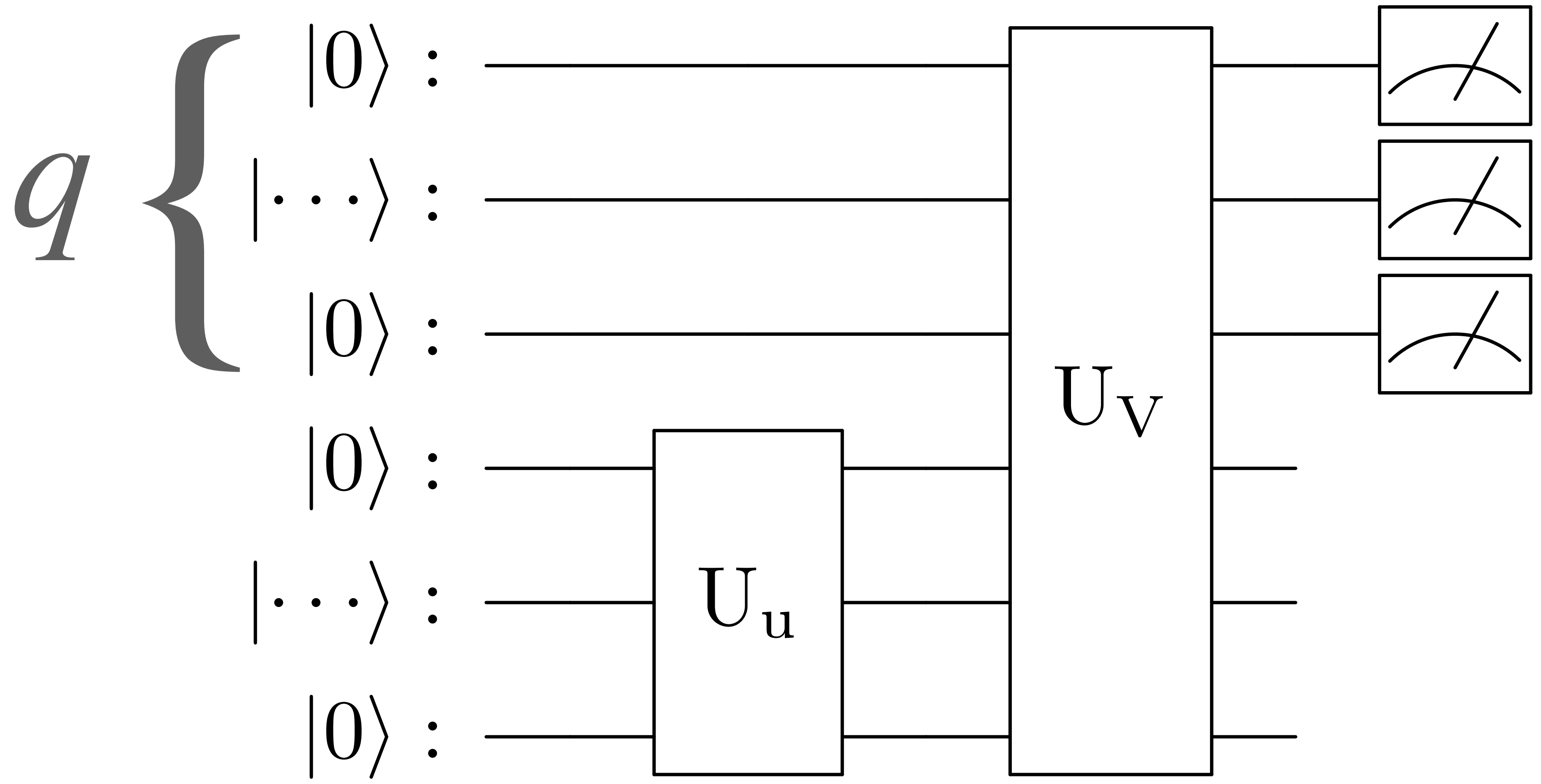
State preparation for $\vec{u}$

$$U_u : |0\rangle \mapsto |\vec{u}\rangle$$

(α, q, ϵ_0)-Block-encoding of V

$$U_V = \begin{bmatrix} \frac{\overline{V}}{\alpha} & \cdot \\ \cdot & \cdot \end{bmatrix}$$

Estimate $\frac{\|\vec{w}\|}{\|\vec{u}\|}$ and compare it to γ.



$$P(0) = \left(\frac{\|\overline{w}\|_2}{\alpha\|\vec{u}\|_2}\right)^2$$

Estimate: $\left| \bar{t} - \frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} \right| \leq \xi$ $\quad O\left(\frac{\alpha}{\xi}\right)$

QRAM: $\widetilde{O}\left(\frac{\mu(V)}{\xi}\right) = \widetilde{O}\left(\frac{\sqrt{k}}{\xi}\right)$

Quantum Eigenfaces-based classification

Resulting complexity

Algorithm 1 Eigenfaces-based classification

1: Center the data point:
$\vec{u} = \vec{x} - \frac{1}{N}\sum_{j\in[N]} \vec{x}_j$
2: Compute the weights vector:
$\vec{w} = V\vec{u}$
3: Select the minimum distance between w and the stored weights:
$d = \min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$
4: Save the index of the closest weights vector:
$j^* = \arg\min_{j\in[p]} \|\vec{w} - \vec{c}_j\|_2^2$
5: Output: if $d \leq \delta_1$: *same class of* $\vec{c}_{j^*}$*, output* y_{j^*}*;*
if $\delta_1 < d < \delta_2$: *similar element, output* -1.
if $d \geq \delta_2$: *not a similar element, output* -2.

Main idea:

Encode $\{\vec{c}_1, \cdots, \vec{c}_p\}, \vec{u}$ in quantum states and V in a unitary.

Algorithm 2 Quantum Eigenfaces-based classification

1: Prepare a state proportional to $\vec{w}$:
$|\vec{w}\rangle = \frac{V\vec{u}}{\|V\vec{u}\|}$
2: Perform Euclidean distance estimation in superposition:
$|\varphi\rangle = \frac{1}{\sqrt{p}}\sum_{j\in[p]} |j\rangle \left|\|\vec{w} - \vec{c}_j\|_2^2\right\rangle$
3: Find the minimum distance and index using a variant of Grover's search
4: Output: if $\overline{d} \leq \delta_1$: *same class of* $\vec{c}_{j^*}$*, output* y_{j^*}*;*
if $\delta_1 < \overline{d} < \delta_2$: *similar element, output* -1.
if $\overline{d} \geq \delta_2$: *not a similar element, output* -2.

Norm-based Outlier detection

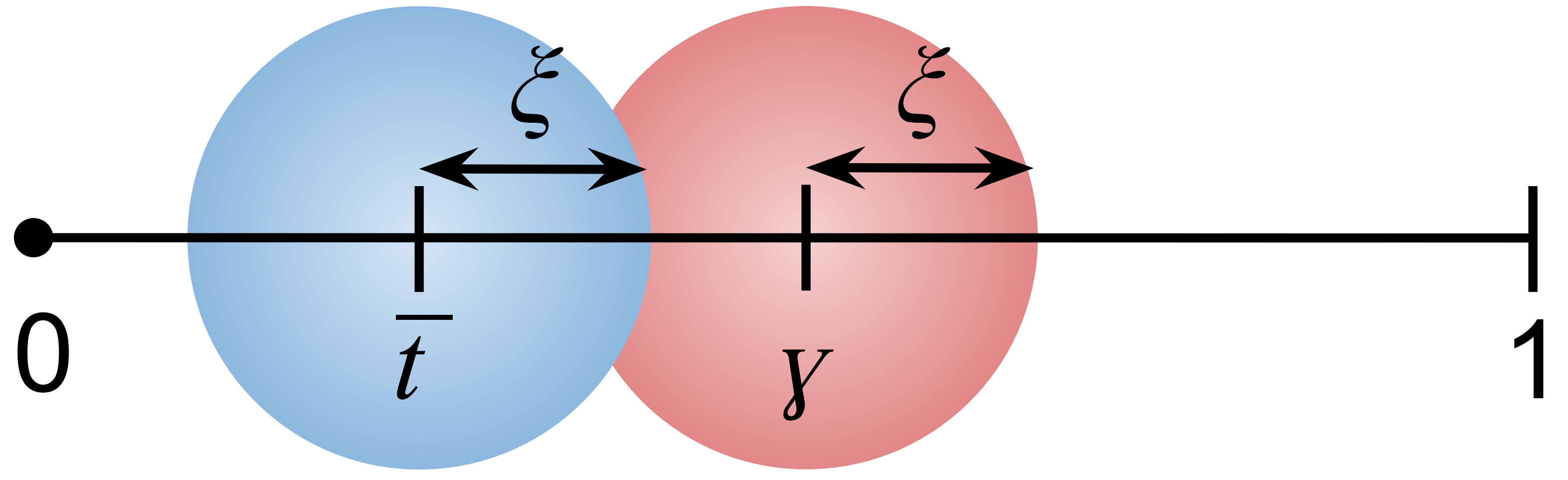
$$\frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} > \gamma \qquad O(mk + pk)$$

$$\widetilde{O}\left(\sqrt{p}\mu(A)\frac{\max_{i\in[p]}(\|\vec{c}_i\|_2)\|\vec{u}\|_2}{\epsilon}\right)$$

Errors and uncertainties

Expected time vs heuristic choice

Estimate: $\left| \bar{t} - \frac{\|\vec{w}\|_2}{\|\vec{u}\|_2} \right| \leq \xi$



Estimate: $\left| \bar{d} - \|\vec{w} - \vec{c}_{j*}\|_2^2 \right| \leq \epsilon$

Numerical experiments

Goal and methodology

Testing the allowable error and comparing running times

$$O(mk + pk) \qquad \widetilde{O}\left(\sqrt{p}\mu(A)\frac{\max_{i\in[p]}(\|\vec{c}_i\|_2)\|\vec{u}\|_2}{\epsilon}\right)$$

We run the classical algorithm, introducing uniformly sampled noise $[-\epsilon, +\epsilon]$ in the distance estimation and norm ratio estimation ($[-0.01, +0.01]$)

https://github.com/WilliamBonvini/quantum-eigenfaces

Datasets

Two tasks: Face recognition and image classification

Face recognition

ORL:

Ext Yale B:

Image classification

MNIST +
Fashion MNIST:

Datasets

Two tasks: Face recognition and image classification

Face recognition

ORL:  112×92

Ext Yale B:  192×168

Image classification

MNIST + Fashion MNIST:  28×28

 28×28

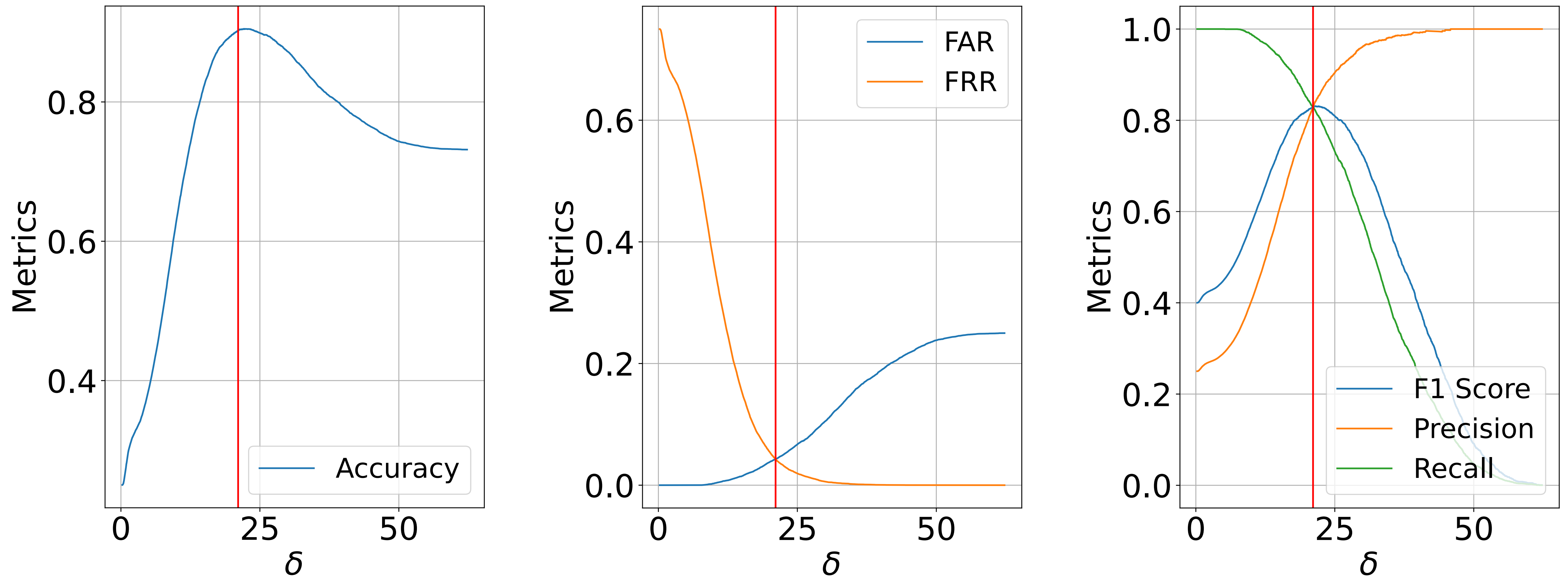
Dataset	# Labels	m	Outliers	#Training (p)	k	#Validation	#Test
ORL	40	10304	No	288	70	36	36
			Yes			54	54
YALE	38	32256	No	322	80	69	70
			Yes			149	134
MNIST	10	784	No	49000	60	10500	10500
			Yes			14000	14000

Datasets

Image classification - Hyperparameter tuning δ

Image classification

MNIST + Fashion MNIST:

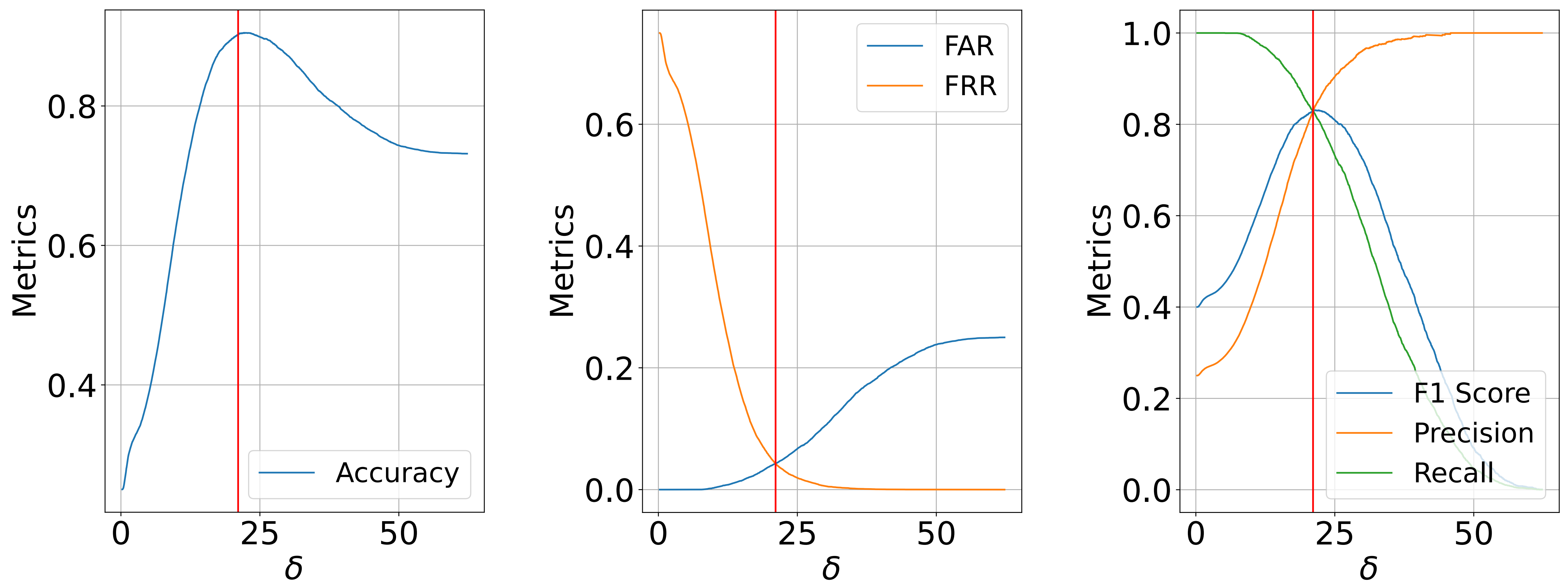


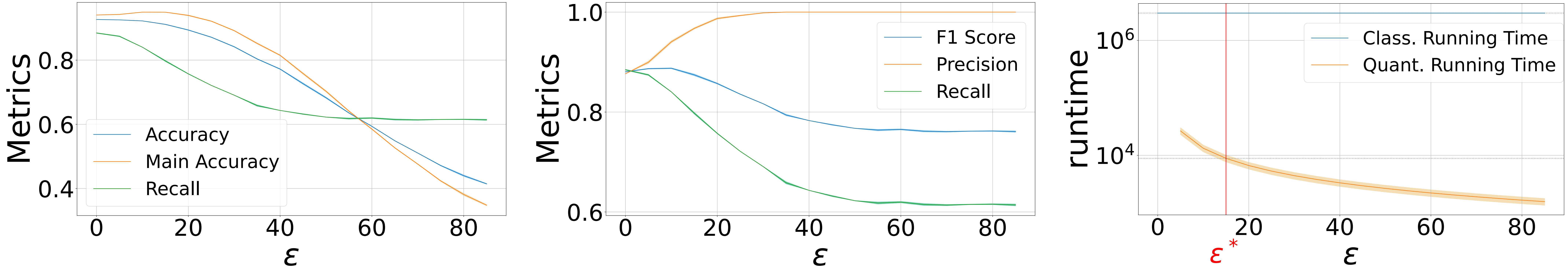
Datasets

Image classification - Performances with ϵ

Image classification

MNIST + Fashion MNIST:





Results

Summary of the numerical experiments

Dataset	δ	Outliers	γ	ϵ	Run. time	Acc.	Main Acc.	Recall
ORL	∞	No	-	0	$7.41 \cdot 10^5$	-	0.944	-
	74.38	Yes	-	0	$7.41 \cdot 10^5$	0.870	0.833	0.944
	74.38	Yes	0.75	0	$7.41 \cdot 10^5$	0.870	0.833	0.944
	74.38	Yes	0.75	15	$3.12 \cdot 10^3$	0.849	0.829	0.888
YALE	∞	No	-	0	$2.61 \cdot 10^6$	-	0.986	-
	232.0	Yes	-	0	$2.61 \cdot 10^6$	0.888	0.900	0.875
	232.0	Yes	0.94	0	$2.61 \cdot 10^6$	0.940	0.900	0.984
	232.0	Yes	0.94	100	$2.84 \cdot 10^3$	0.910	0.866	0.959
MNIST	∞	No	-	0	$2.99 \cdot 10^6$	-	0.975	-
	23.52	Yes	-	0	$2.99 \cdot 10^6$	0.906	0.940	0.803
	22.34	Yes	0.75	0	$2.99 \cdot 10^6$	0.927	0.940	0.885
	22.34	Yes	0.75	15	$6.66 \cdot 10^3$	0.913	0.949	0.804

Results

Summary of the numerical experiments

Dataset	δ	Outliers	γ	ϵ	Run. time	Acc.	Main Acc.	Recall
ORL	∞	No	-	0	$7.41 \cdot 10^5$	-	0.944	-
	74.38	Yes	-	0	$7.41 \cdot 10^5$	0.870	0.833	0.944
	74.38	Yes	0.75	0	$7.41 \cdot 10^5$	0.870	0.833	0.944
	74.38	Yes	0.75	15	$3.12 \cdot 10^3$	0.849	0.829	0.888
YALE	∞	No	-	0	$2.61 \cdot 10^6$	-	0.986	-
	232.0	Yes	-	0	$2.61 \cdot 10^6$	0.888	0.900	0.875
	232.0	Yes	0.94	0	$2.61 \cdot 10^6$	0.940	0.900	0.984
	232.0	Yes	0.94	100	$2.84 \cdot 10^3$	0.910	0.866	0.959
MNIST	∞	No	-	0	$2.99 \cdot 10^6$	-	0.975	-
	23.52	Yes	-	0	$2.99 \cdot 10^6$	0.906	0.940	0.803
	22.34	Yes	0.75	0	$2.99 \cdot 10^6$	0.927	0.940	0.885
	22.34	Yes	0.75	15	$6.66 \cdot 10^3$	0.913	0.949	0.804

Thanks!

armando.bellante@polimi.it
@ikiga1