KENTECH
Korea Institute of Energy Technology

Advanced Computer Vision
Week 11

Nov. 15, 2022

Seokju Lee

Parts of slides are by Prof. In So Kweon and Prof. Shree Nayar

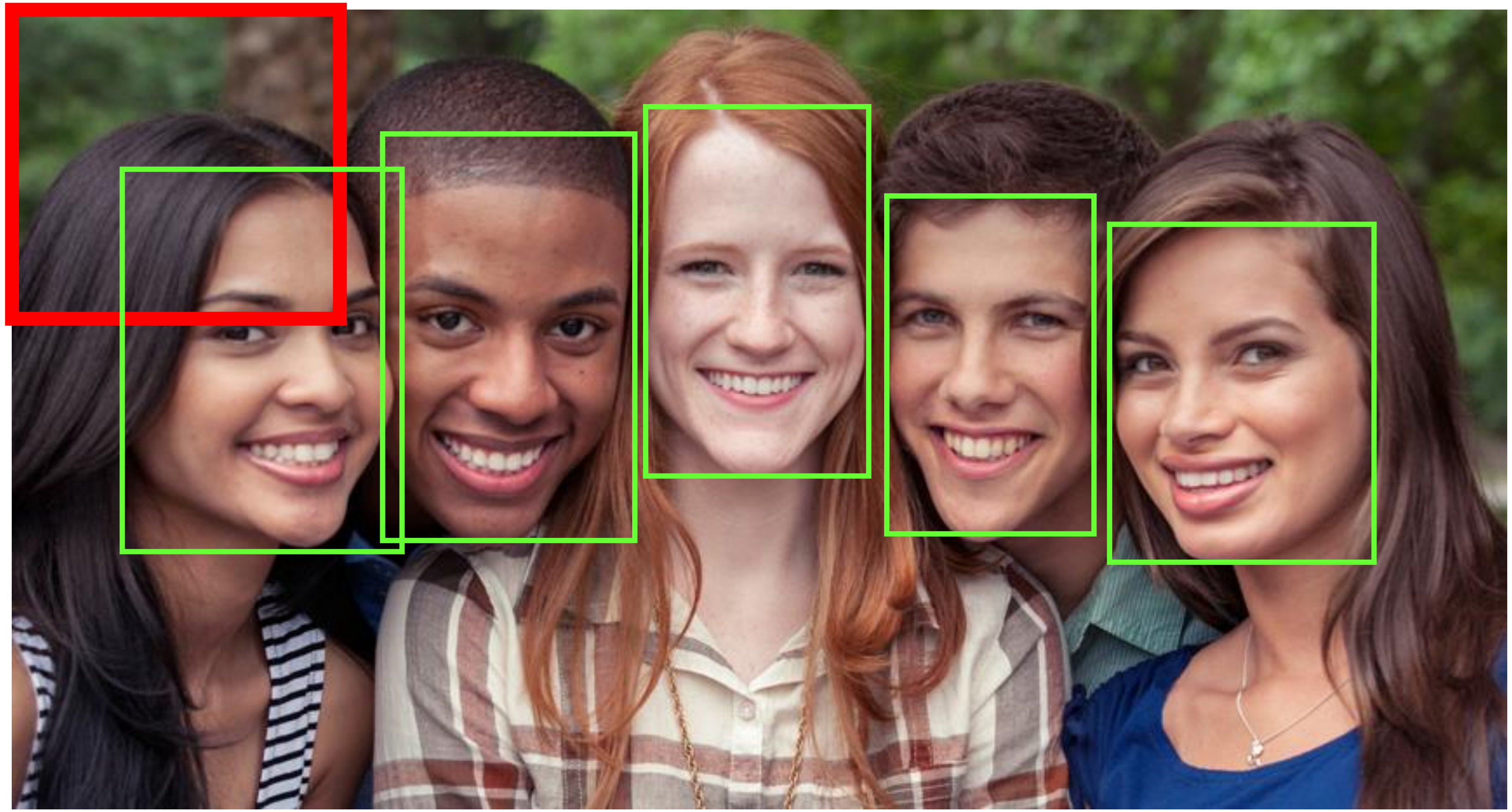
KENTECH
Korea Institute of Energy Technology

Object Recognition in the Past: Face Detection

Face Detection

Slide windows of **different** sizes across image.

At each location, **decide** the sample whether it is face or not.



Face Detection Framework

For each window:

Extract features → $\begin{bmatrix} \mathbf{f} \end{bmatrix}$ → Classify feature vector → **Yes / No**

Features: Which features represent faces well?

Classifier: How to construct a face model and efficiently classify features as face or not?

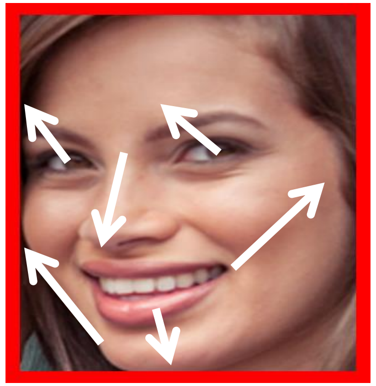
Parts of slides are by Prof. In So Kweon and Prof. Shree Nayar

KENTECH
Korea Institute of Energy Technology

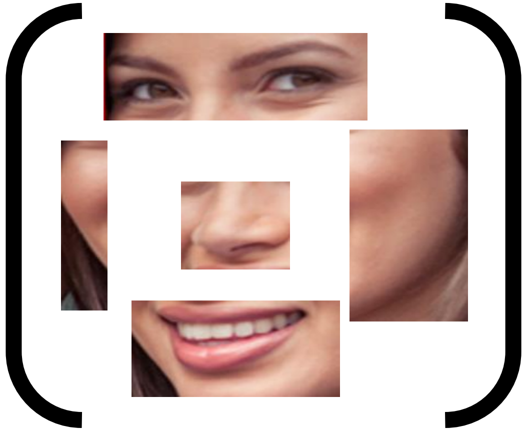
Features for Face Detection

What are Good Features?

Interest points (e.g., edges, corners, SIFT)?



Facial components (e.g., templates)?

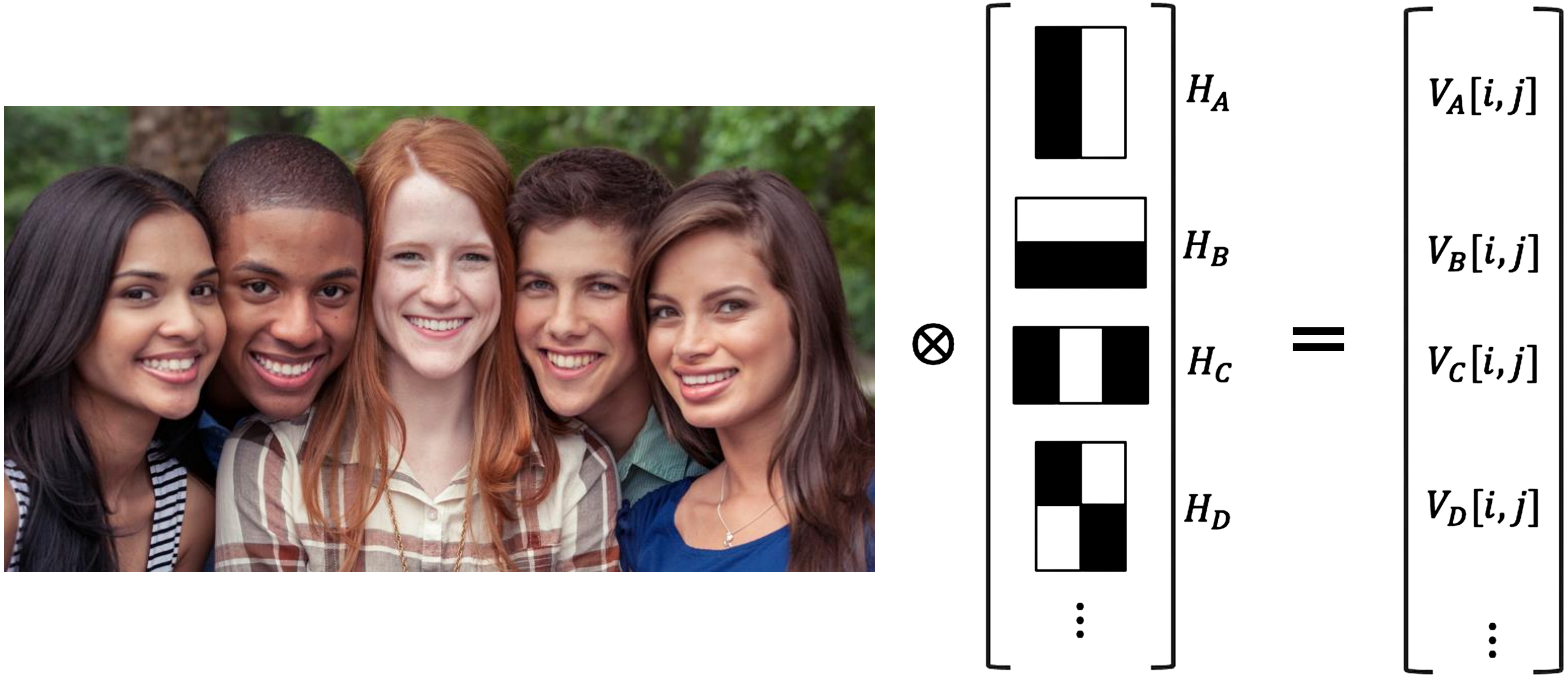


+ **Must be extremely <u>fast</u> to compute!**

(Need to process millions of windows in an image)

Haar Features

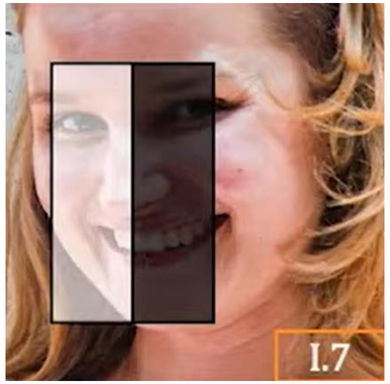
Set of correlation responses to **Haar** filters



Haar filters Haar features

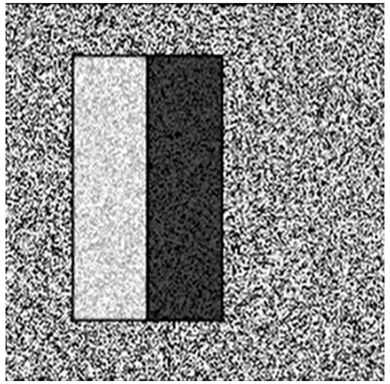
Discriminative Ability of Haar Features

Haar features are **sensitive** to **directionality** of patterns!

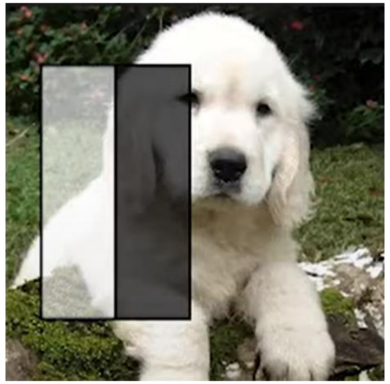


$V_A = 64$

$V_A = 16$



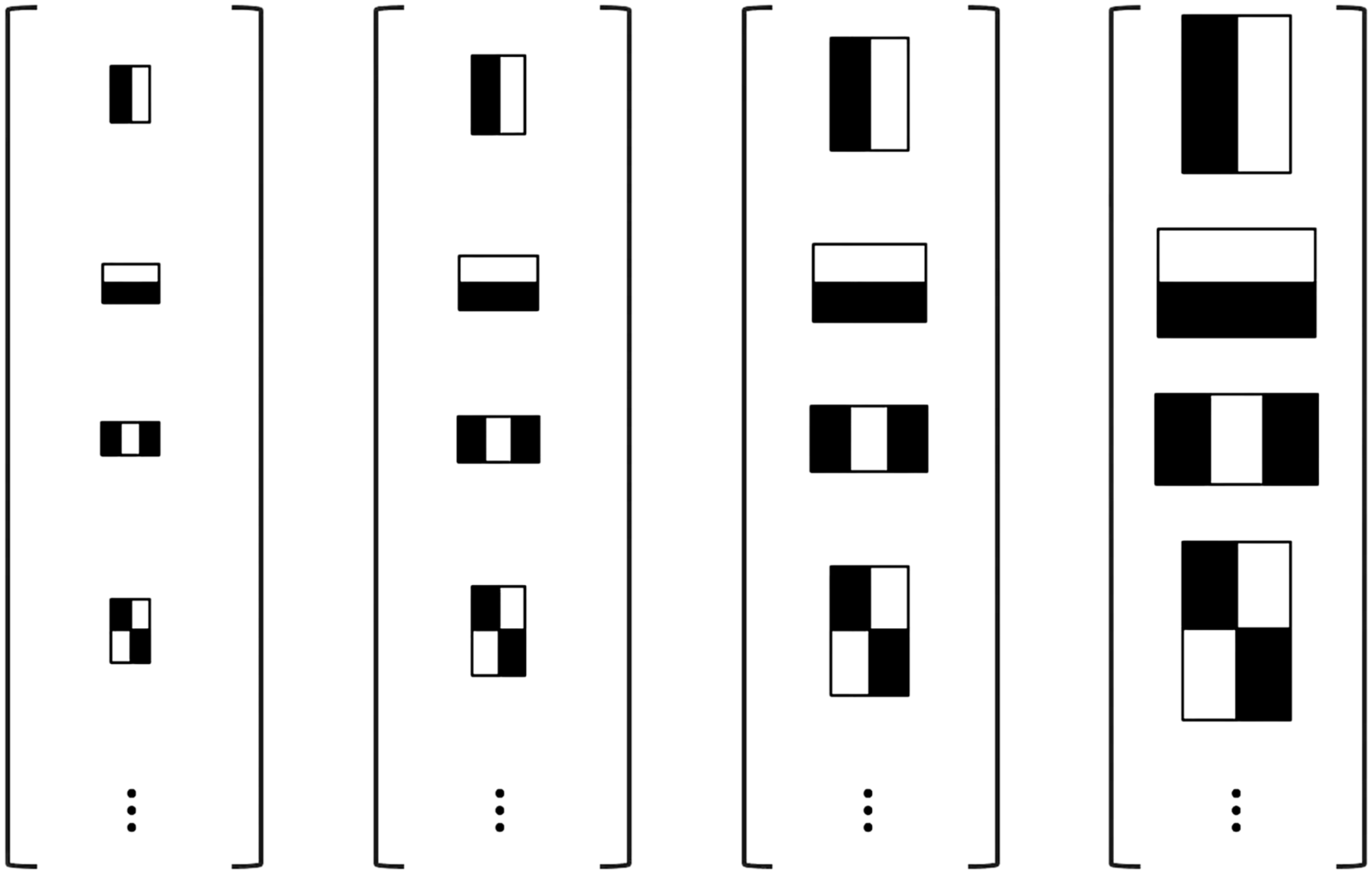
$V_A = 0$



$V_A = -127$

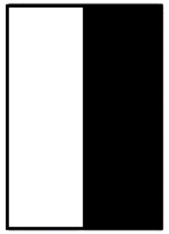
Detecting Faces of Different Size

Compute Haar features at **different scales** to detect faces of different sizes!



Computing a Haar Feature

$\otimes$



H_A

White = 1, Black = -1

Response to Filter H_A at location (i, j):

$$V_A[i,j] = \sum_m \sum_n I[m-i, n-j] H_A[m,n]$$

$$V_A[i,j] = \sum (\text{pixel intensities in white area}) - \sum (\text{pixels intensities in black area})$$

Computing Integral Image

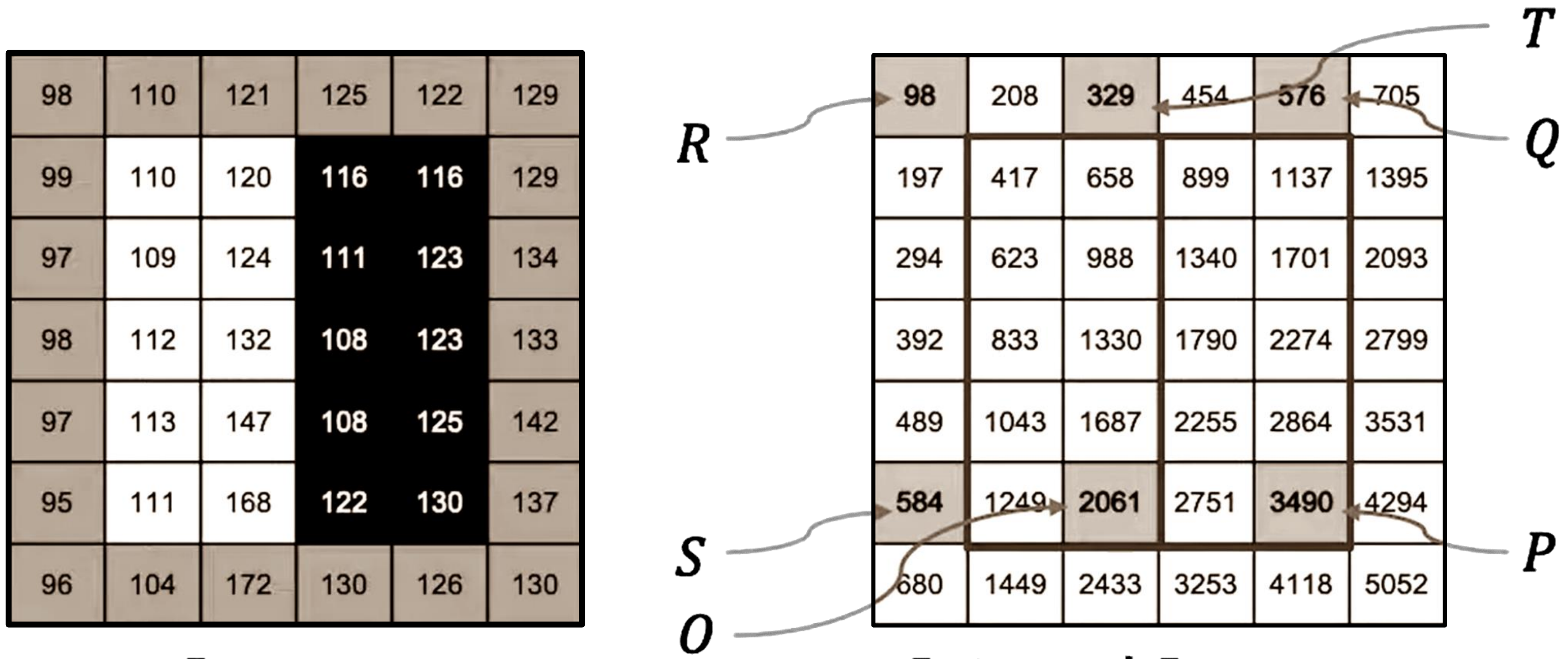


Image I Integral Image II

$$V_A = \sum(\textit{pixel intensities in white}) - \sum(\textit{pixel intensities in black})$$

$$= (II_O - II_T + II_R - II_S) - (II_P - II_Q + II_T - II_O)$$

$$= (2061 - 329 + 98 - 584) - (3490 - 576 + 329 - 2061) = 64$$

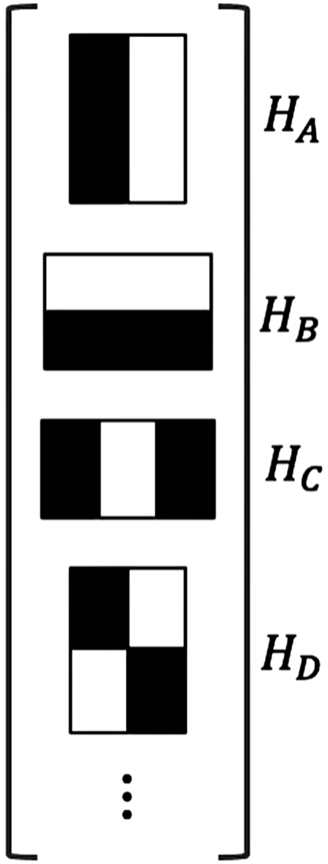
Computational Cost: Only 7 additions

Haar Features Using Integral Images

Integral image needs to be computed **once** per test image.

Allows **fast** computations of Haar features.

$\otimes$



$$=$$

$$\begin{bmatrix} V_A[i,j] \\ V_B[i,j] \\ V_C[i,j] \\ V_D[i,j] \\ \vdots \end{bmatrix}$$

Haar filters

Haar features

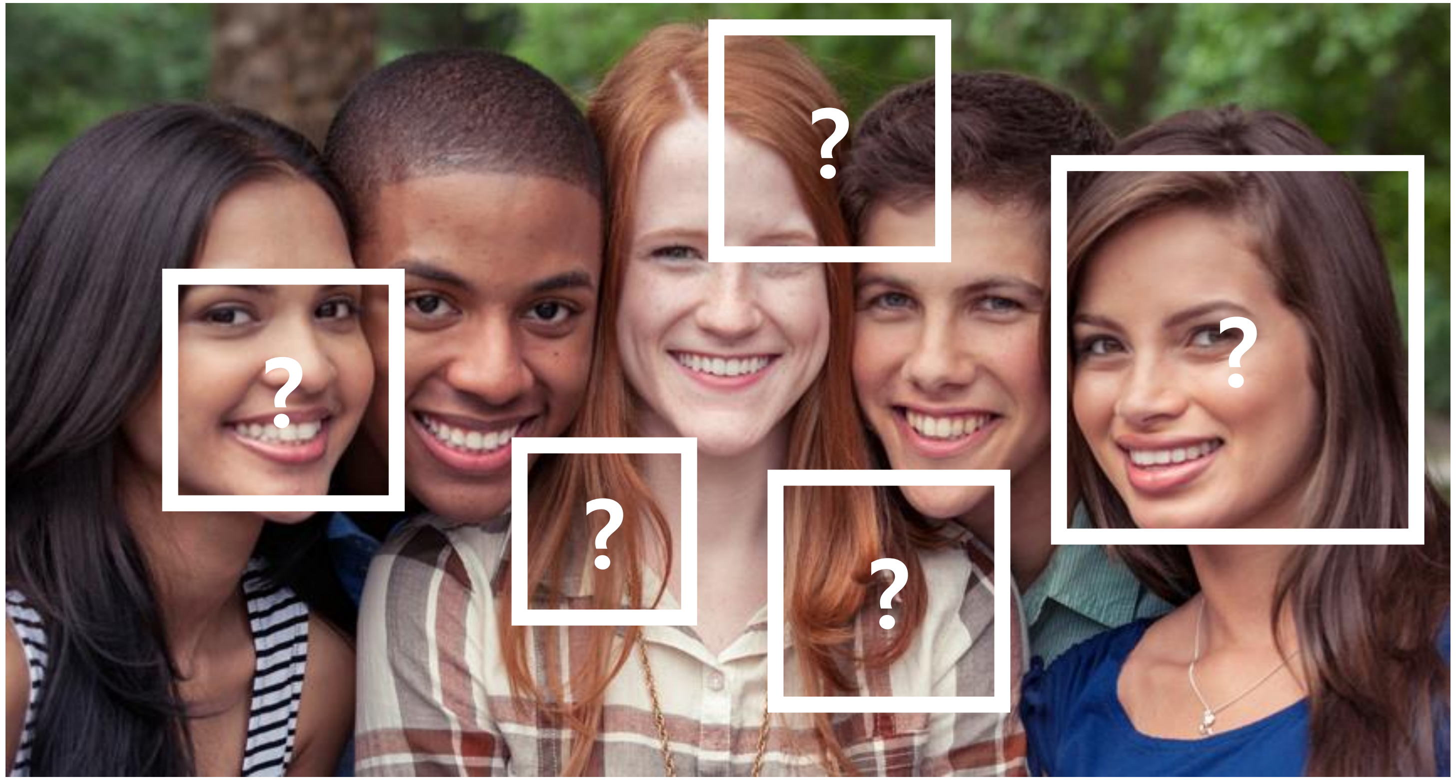
KENTECH
Korea Institute of Energy Technology

Classifier for Face Detection

Classifier for Face Detection

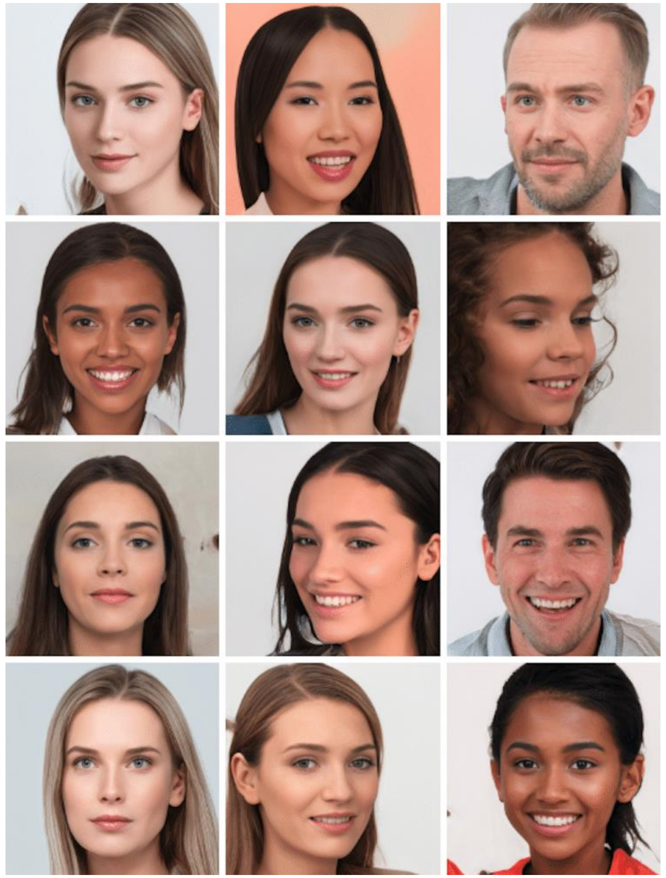
Given the **features** for a window,

How to **decide** whether it contains a face or not?

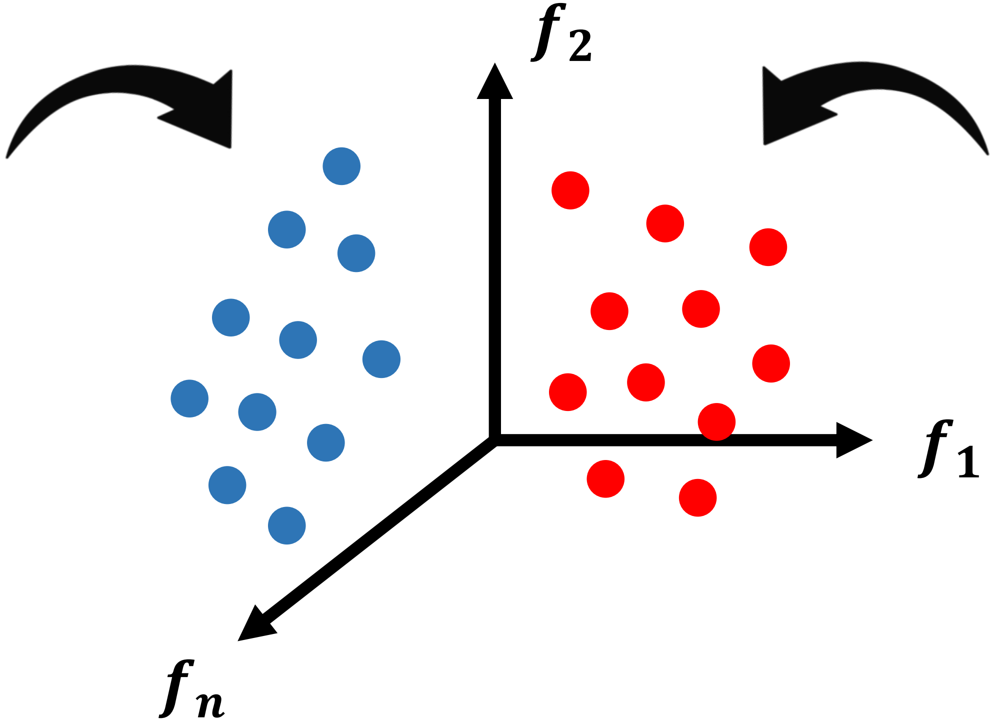


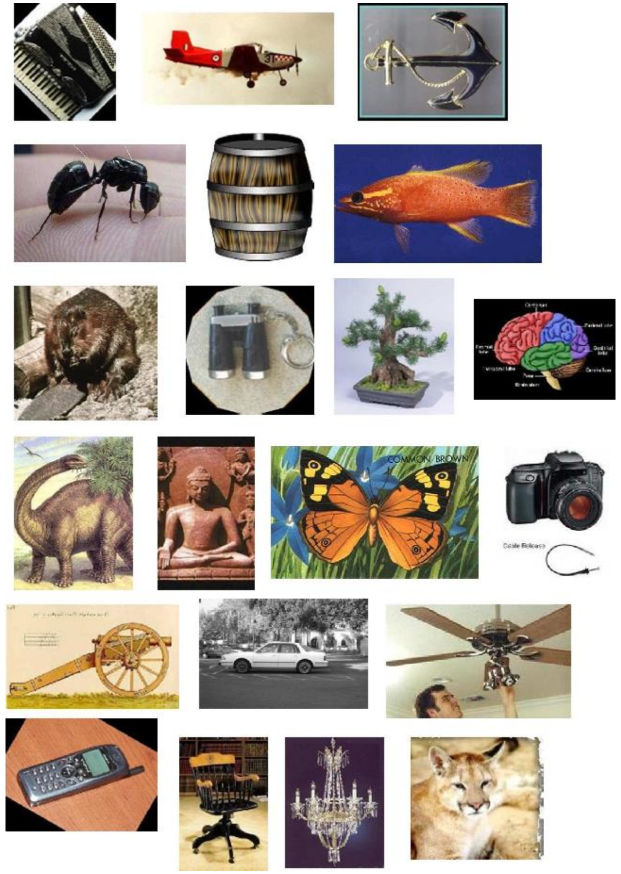
Feature Space

Haar features $\mathbf{f}$ (a vector) at a pixel is a point in an n-D space, $\mathbf{f} \in \mathbb{R}^n$



Training data of **face**

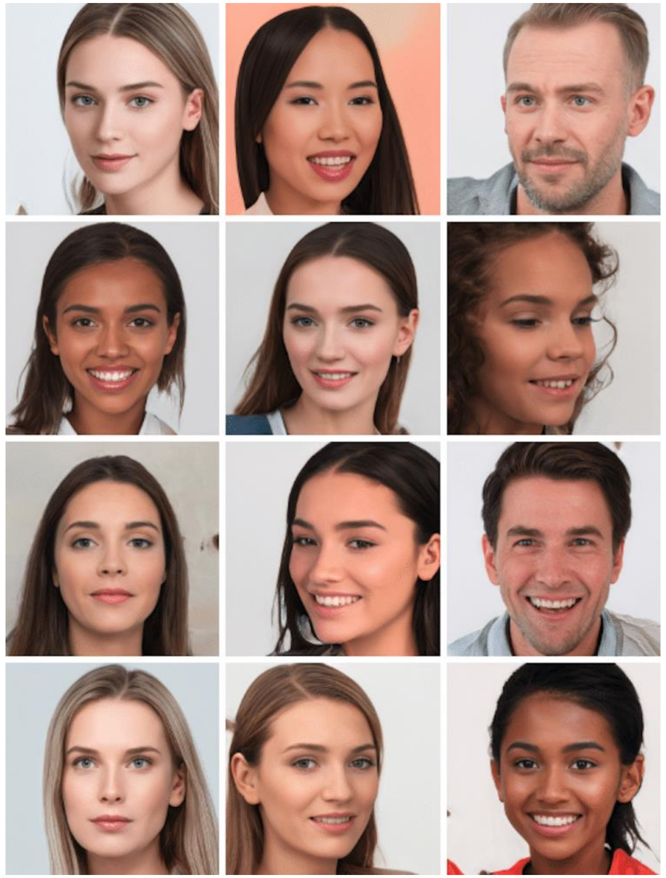




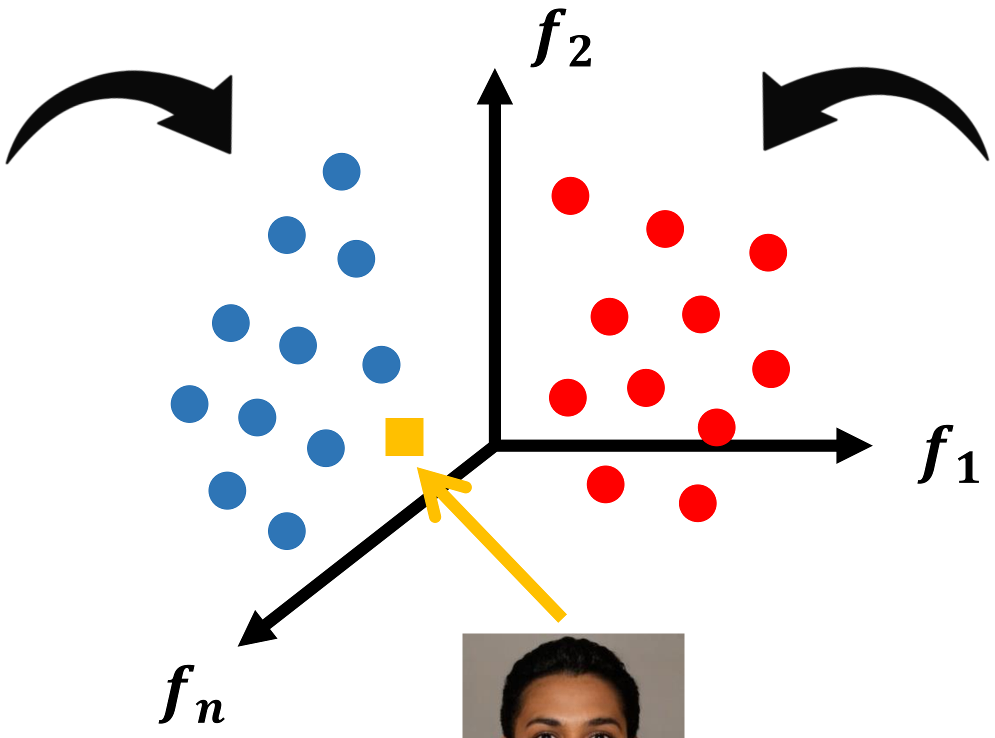
Training data of **non-face**
(Caltech 101)

Nearest Neighbor Classifier

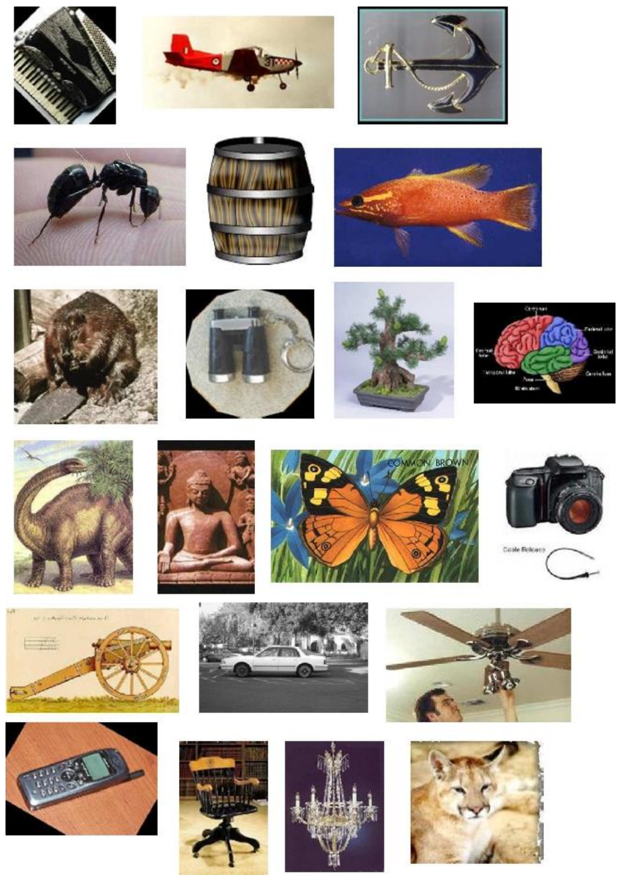
Find the **nearest** training sample using $\boldsymbol{L^2}$ distance and assign its label.



Training data of **face**



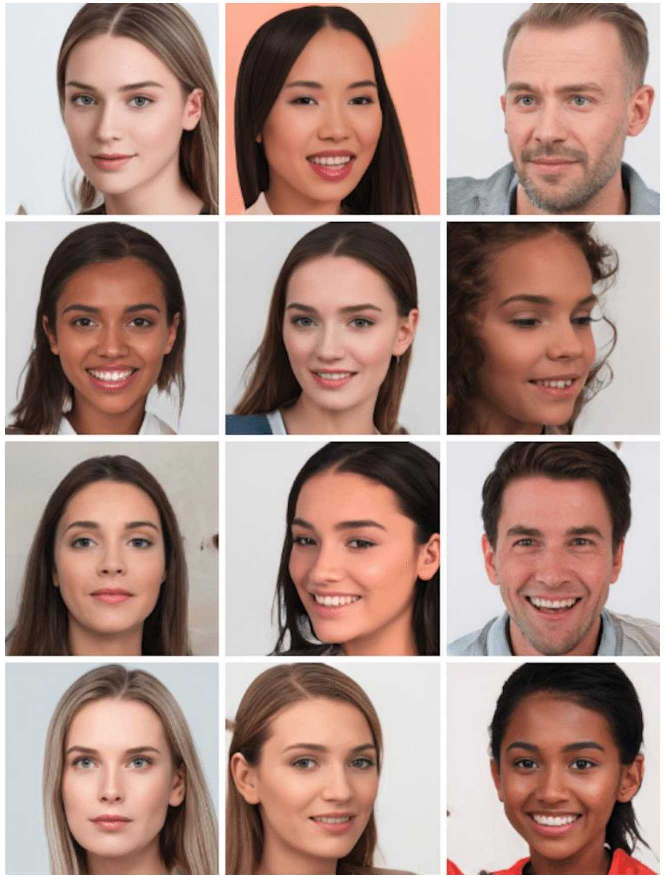
Test image
(true positive)



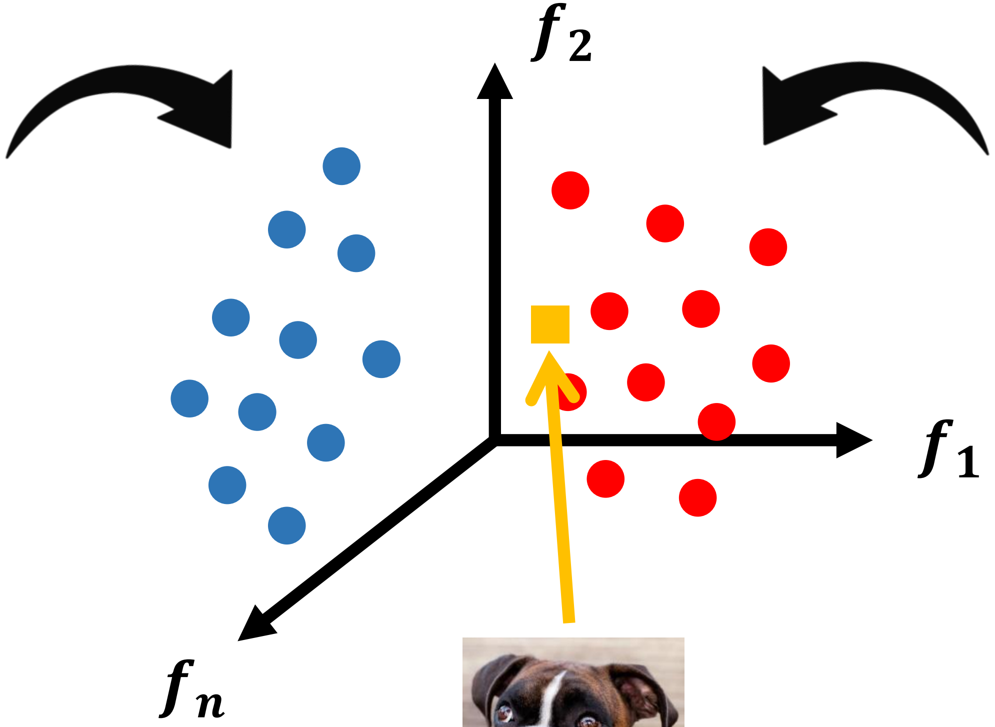
Training data of **non-face**
(Caltech 101)

Nearest Neighbor Classifier

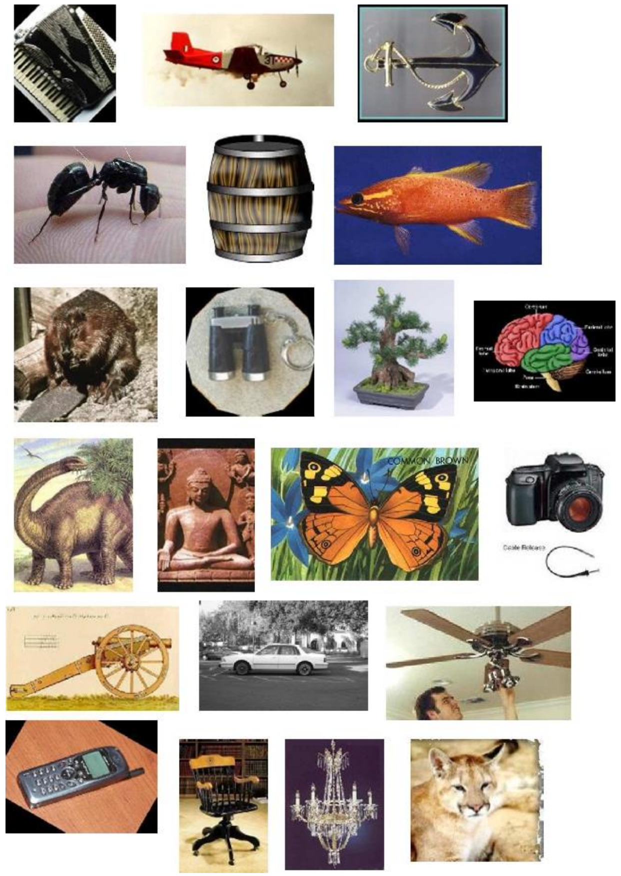
Find the **nearest** training sample using L^2 distance and assign its label.



Training data of **face**



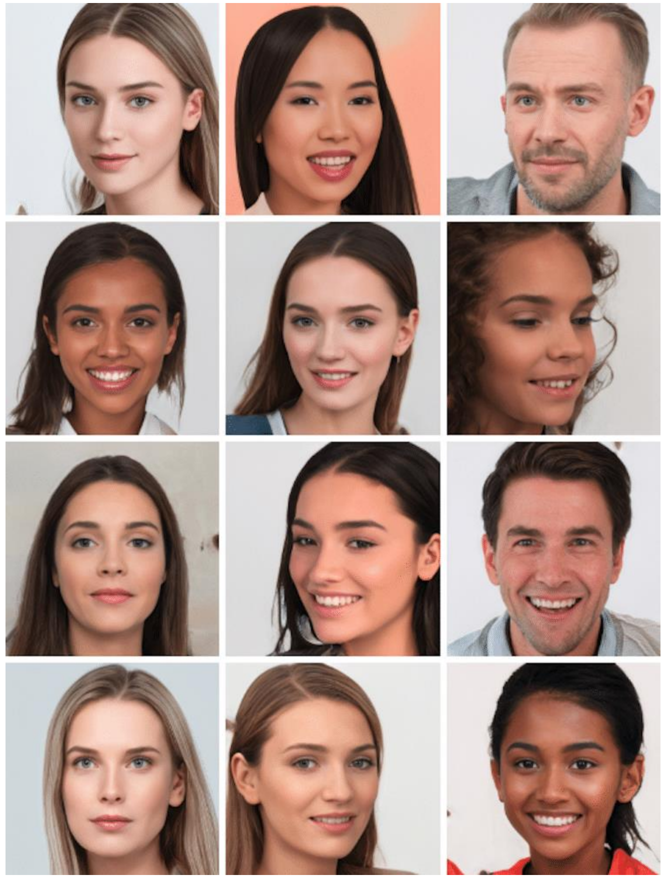
Test image
(true negative)



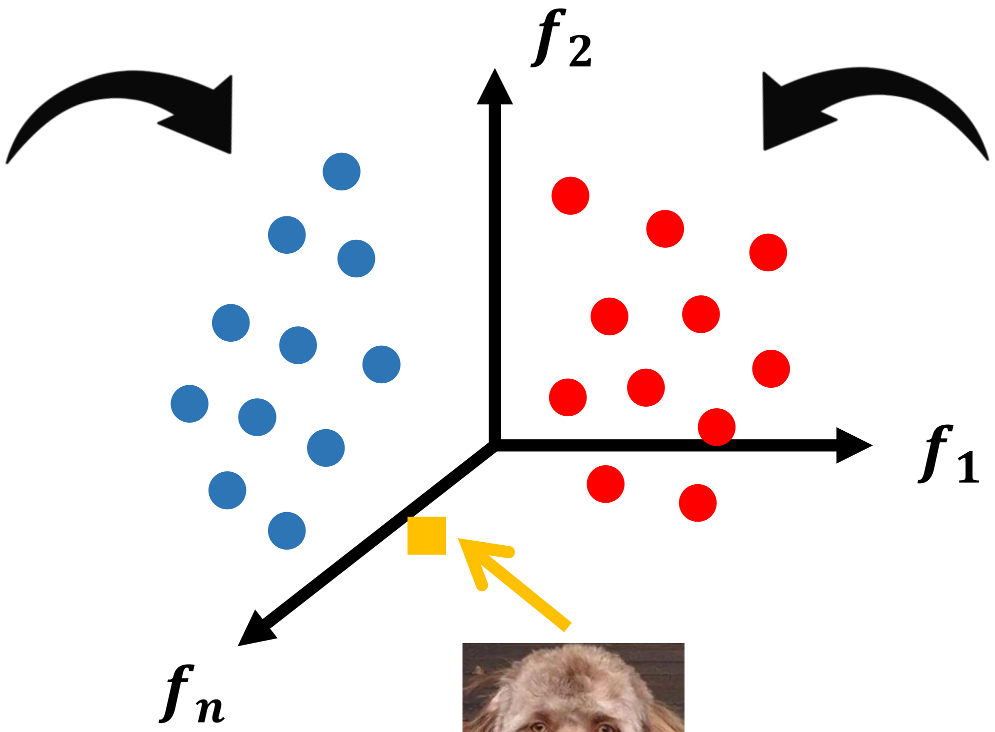
Training data of **non-face**
(Caltech 101)

Nearest Neighbor Classifier

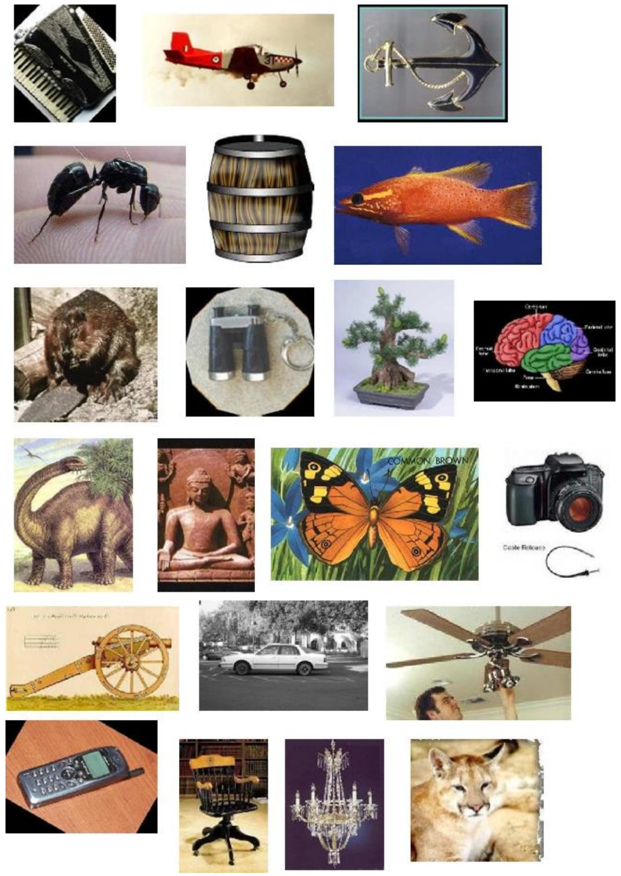
Find the **nearest** training sample using L^2 distance and assign its label.



Training data of **face**



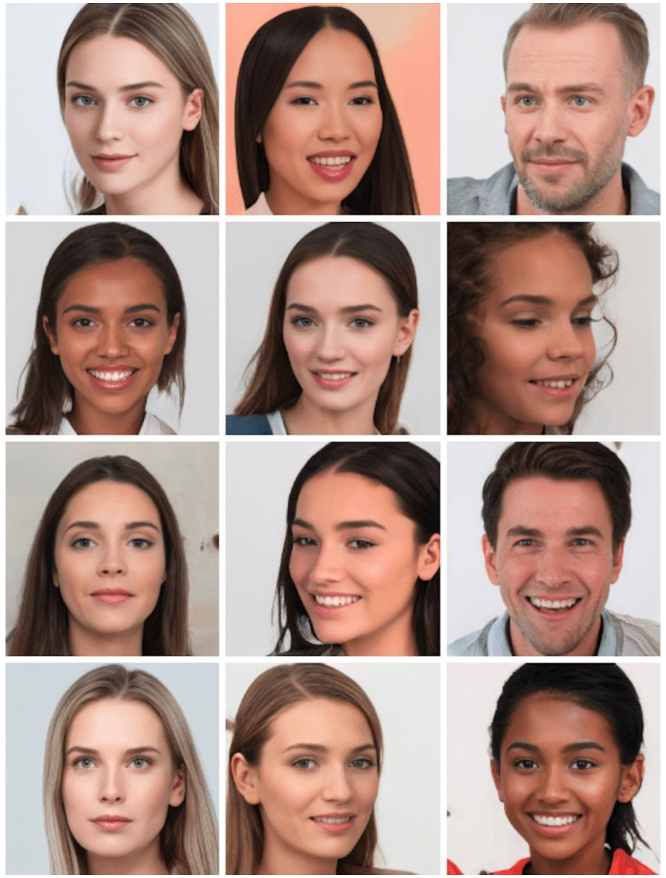
Test image
(false positive)



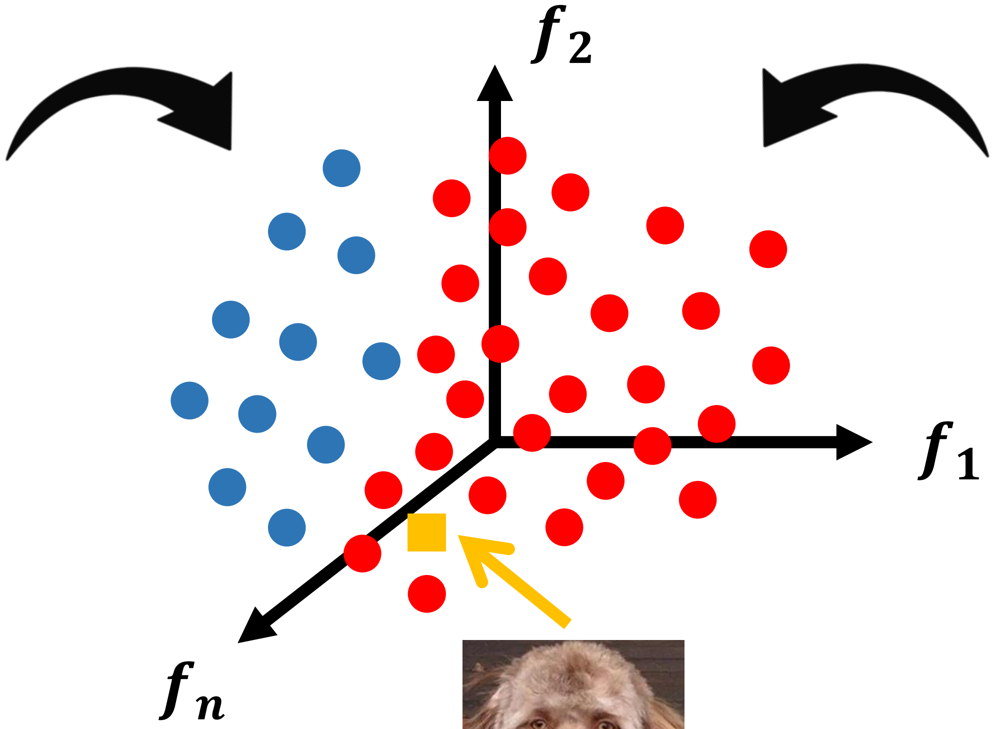
Training data of **non-face**
(Caltech 101)

Nearest Neighbor Classifier

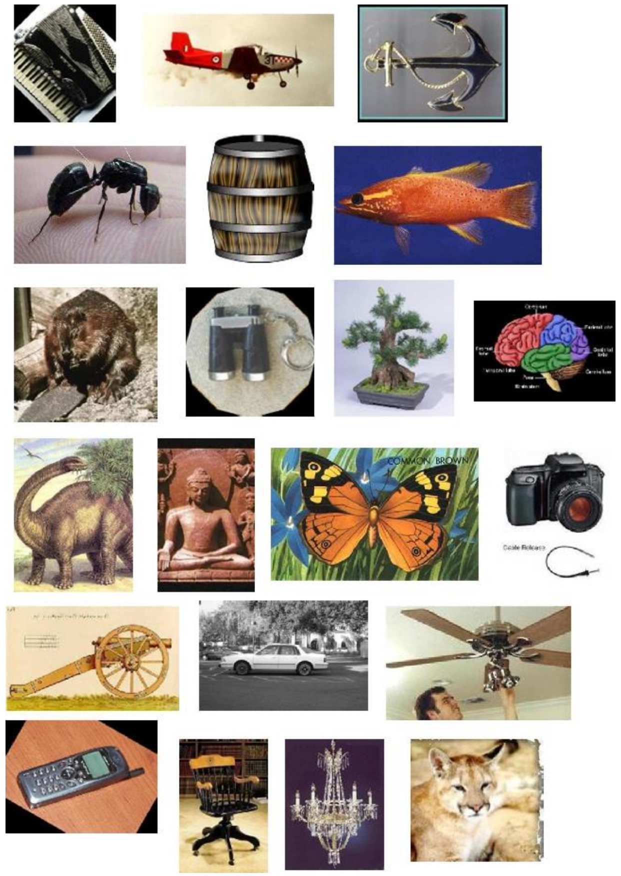
Larger the training set, **more robust** the NN classifier!



Training data of **face**



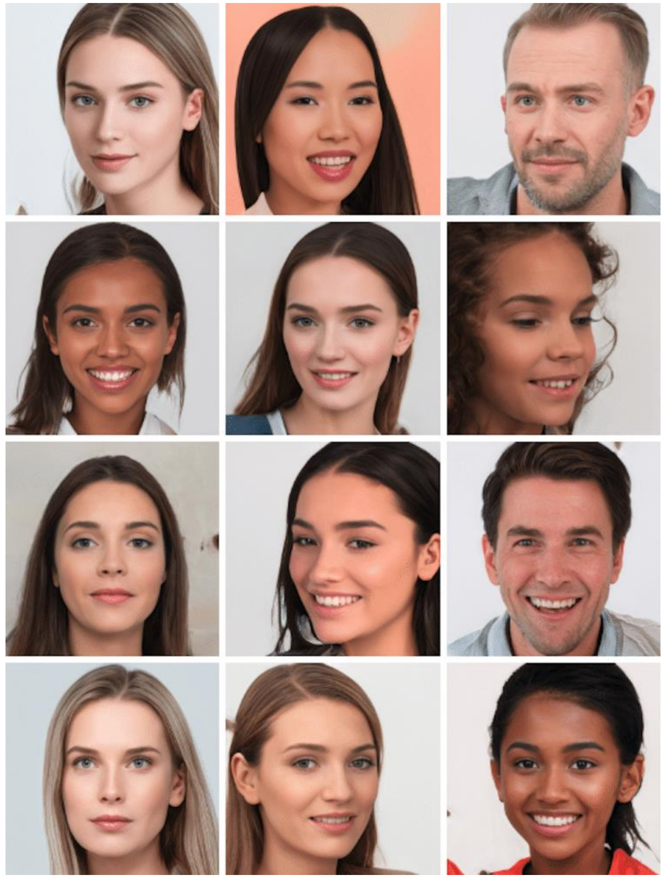
Test image
(false positive)



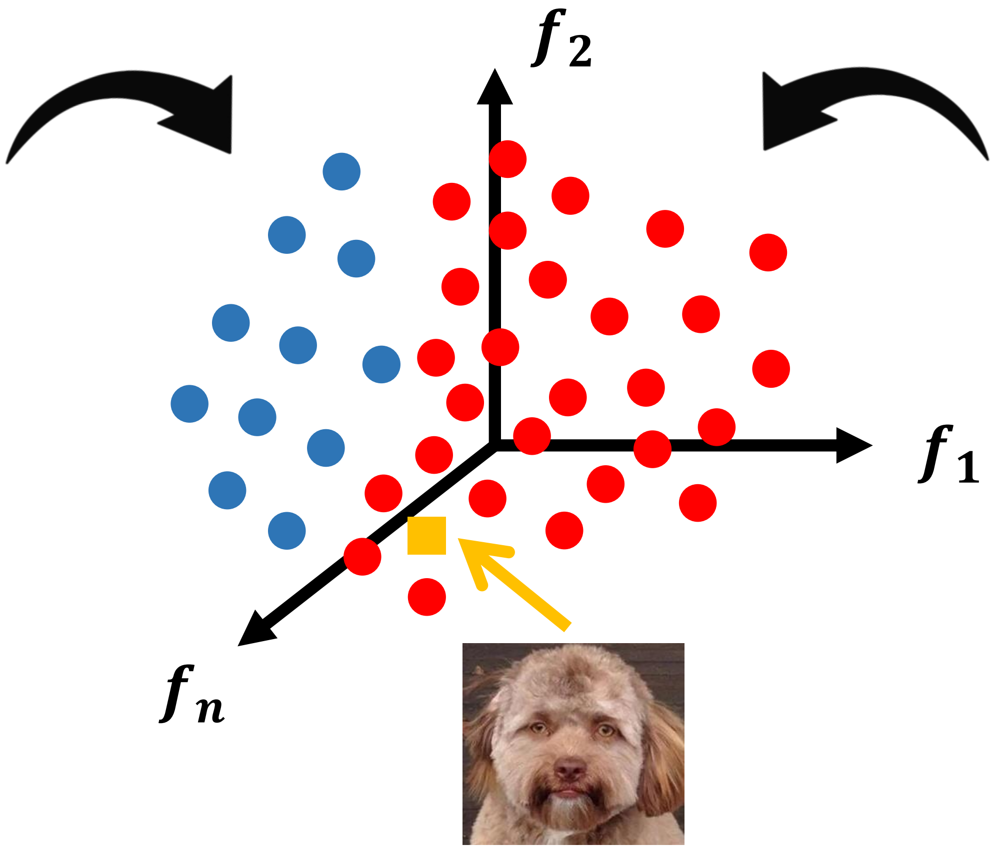
Training data of **non-face**
(Caltech 101)

Nearest Neighbor Classifier

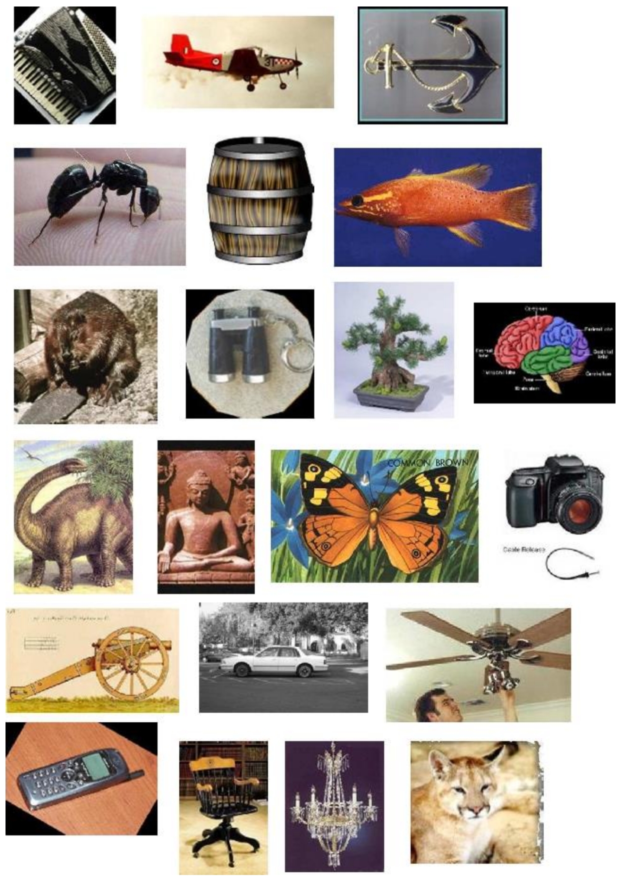
Larger the training set, **slower** the NN classifier!



Training data of **face**



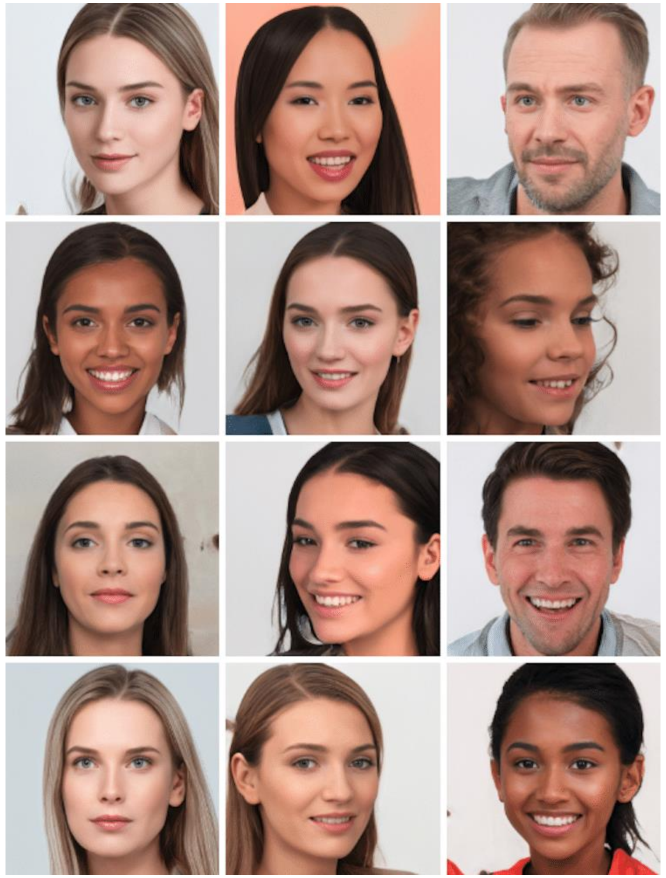
Test image
(false positive)



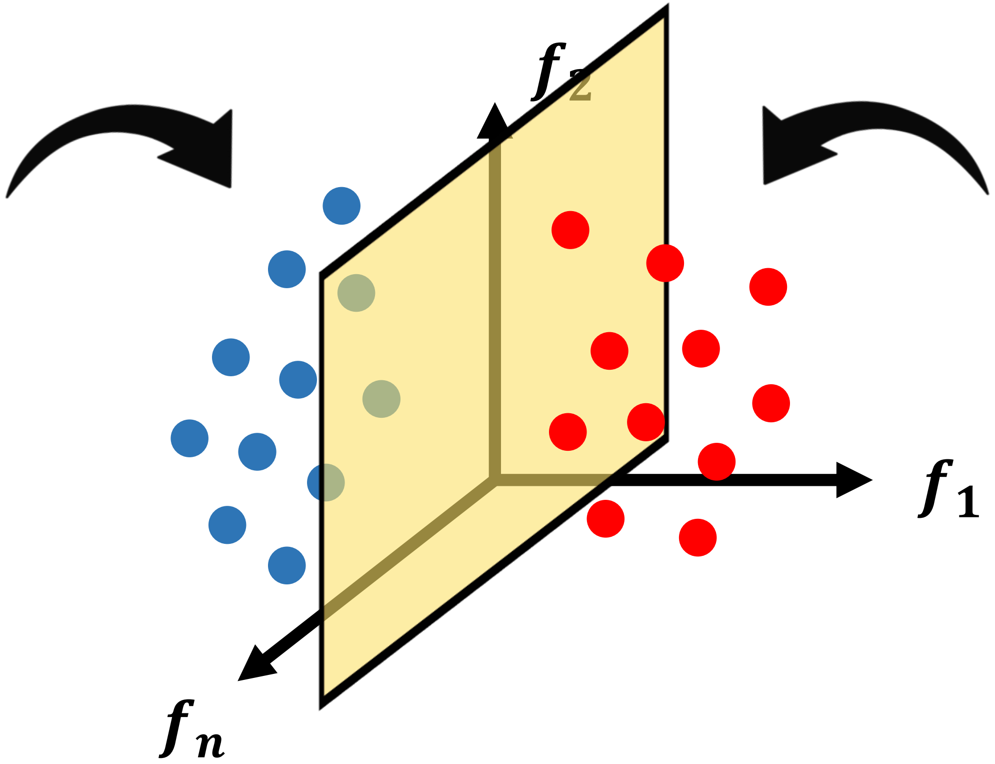
Training data of **non-face**
(Caltech 101)

Decision Boundary

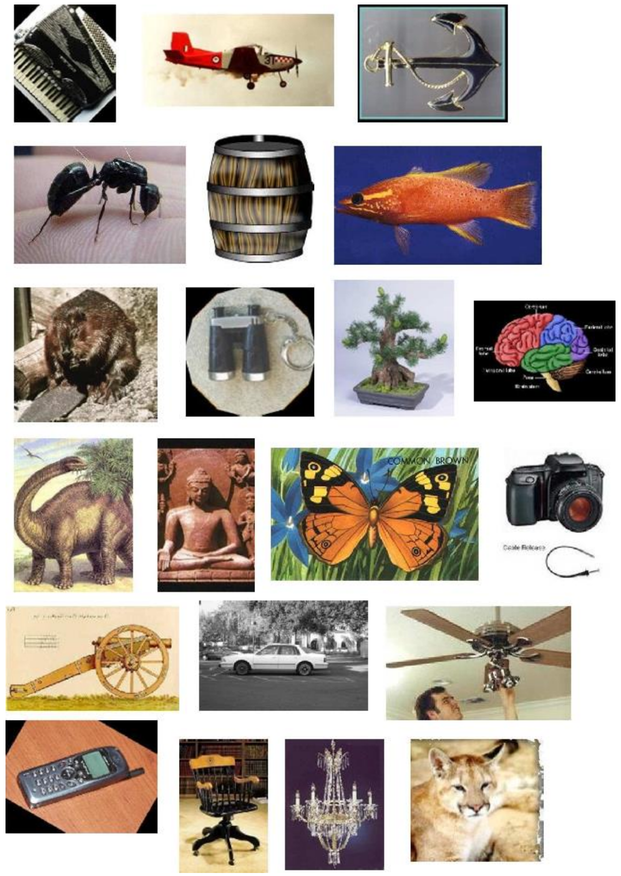
A simple decision boundary **separates** face and non-face.



Training data of **face**



→ **How to design it?**



Training data of **non-face**
(Caltech 101)

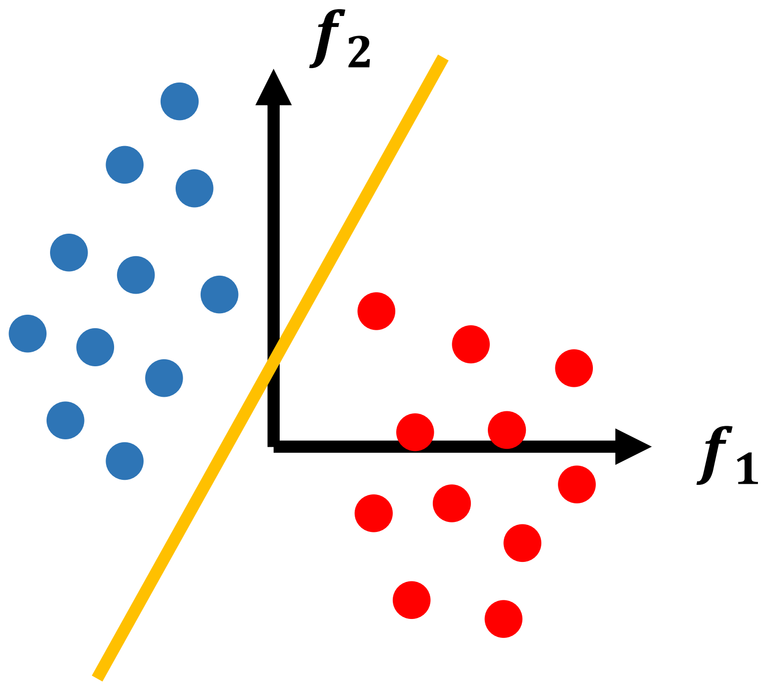
Parts of slides are by Prof. In So Kweon and Prof. Shree Nayar

KENTECH
Korea Institute of Energy Technology

Support Vector Machine (SVM)

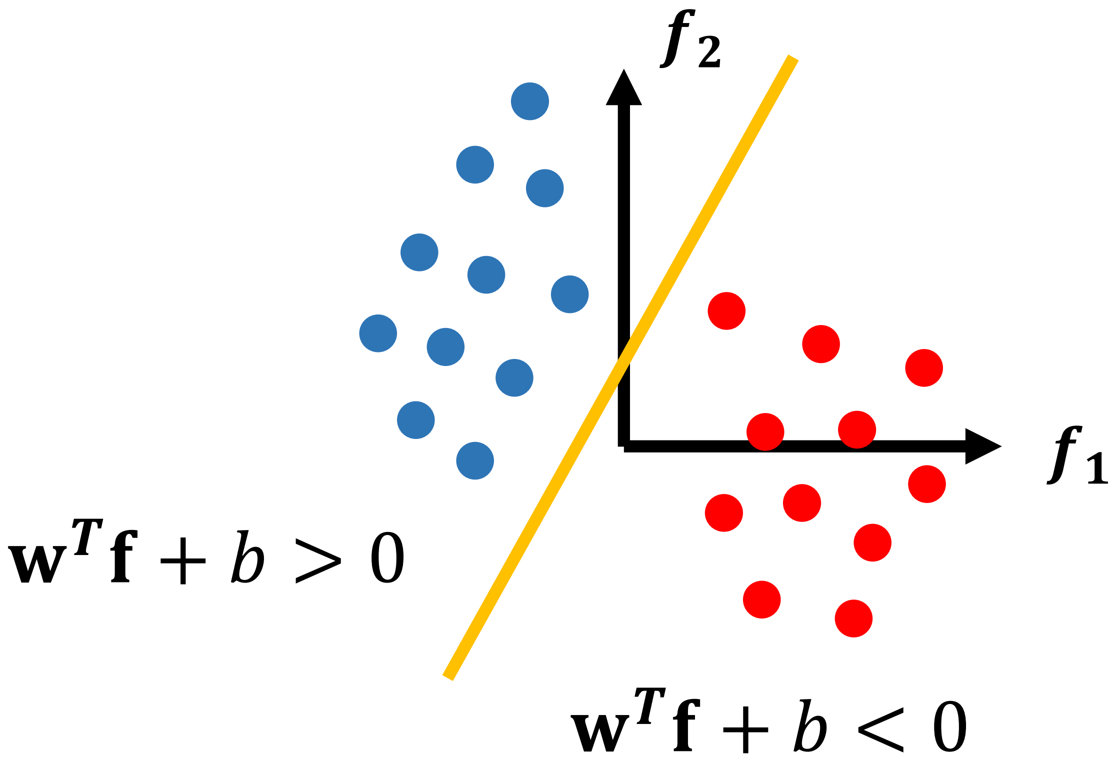
Linear Decision Boundaries

A linear decision boundary in 2D space is a 1D line.



Linear Decision Boundaries

A linear decision boundary in 2D space is a 1D line.



Equation of Line:

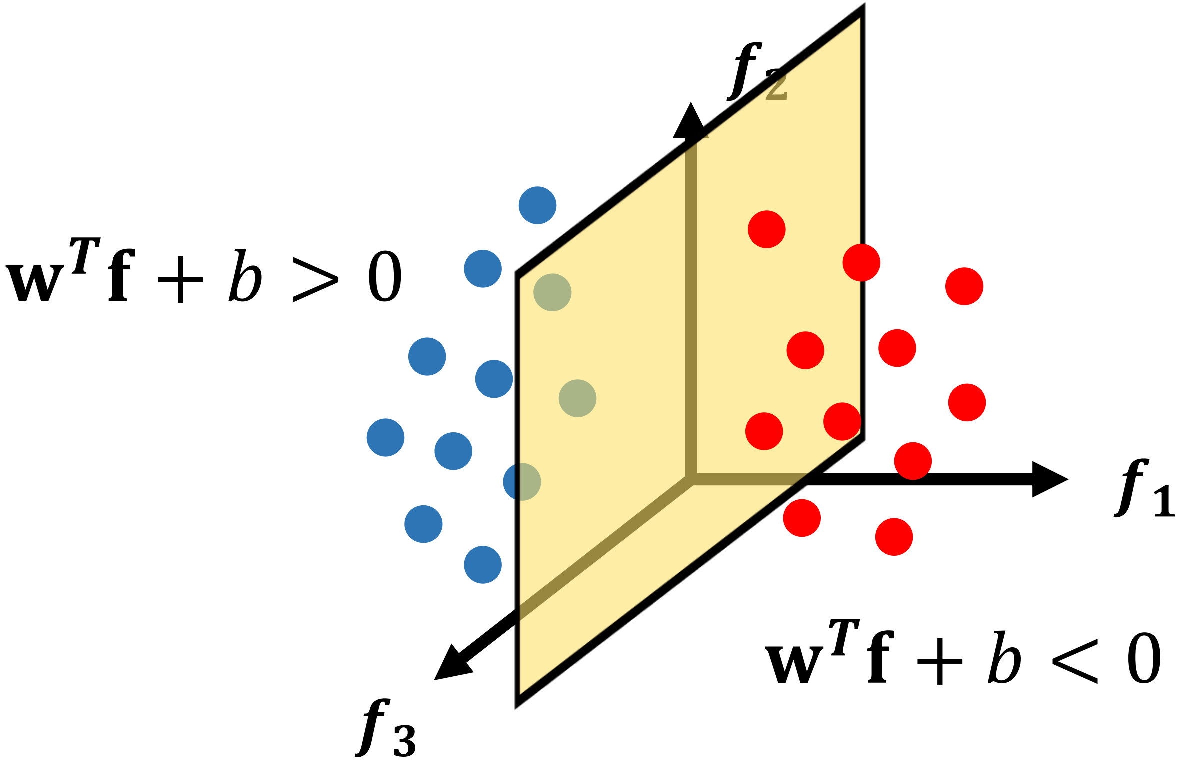
$$w_1 f_1 + w_2 f_2 + b = 0$$

$$\begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + b = 0$$

$$\boxed{\mathbf{w}^T\mathbf{f} + b = 0}$$

Linear Decision Boundaries

A linear decision boundary in 3D space is a 2D plane.



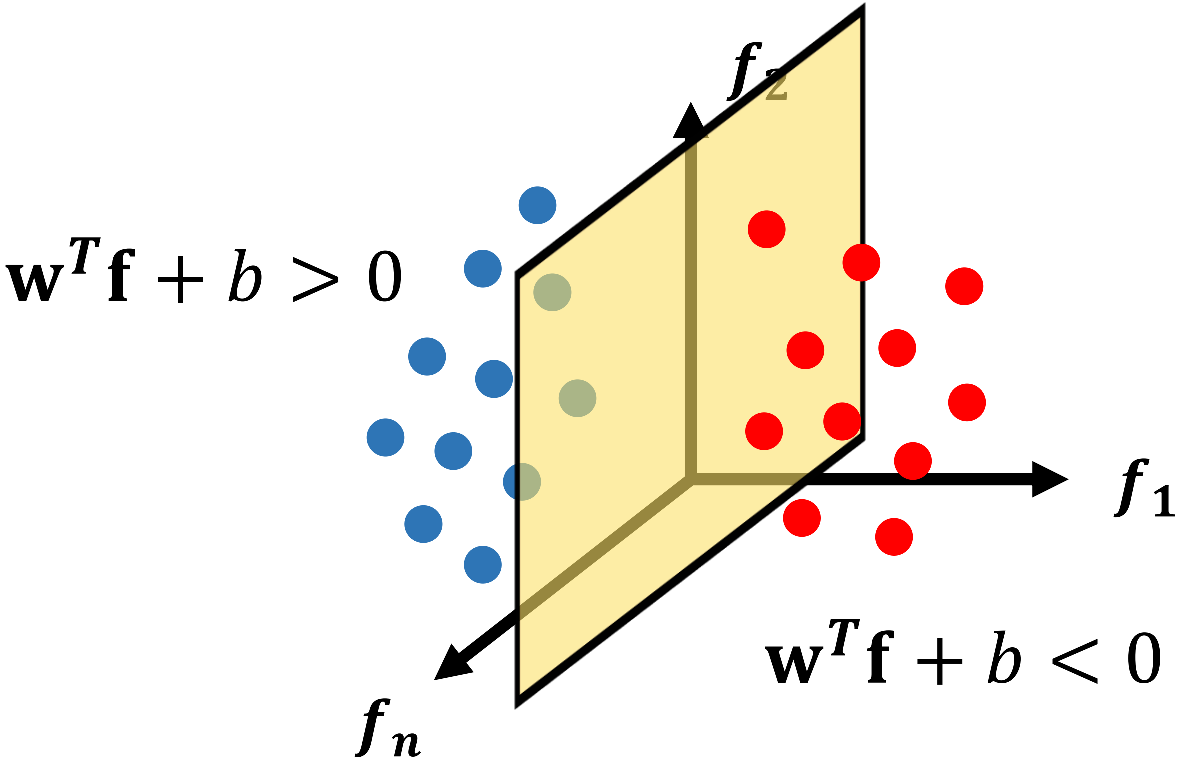
Equation of Plane:

$$w_1 f_1 + w_2 f_2 + w_3 f_3 + b = 0$$

$$\mathbf{w}^T\mathbf{f} + b = 0$$

Linear Decision Boundaries

A linear decision boundary in n-D space is a $(n-1)$-D hyperplane.



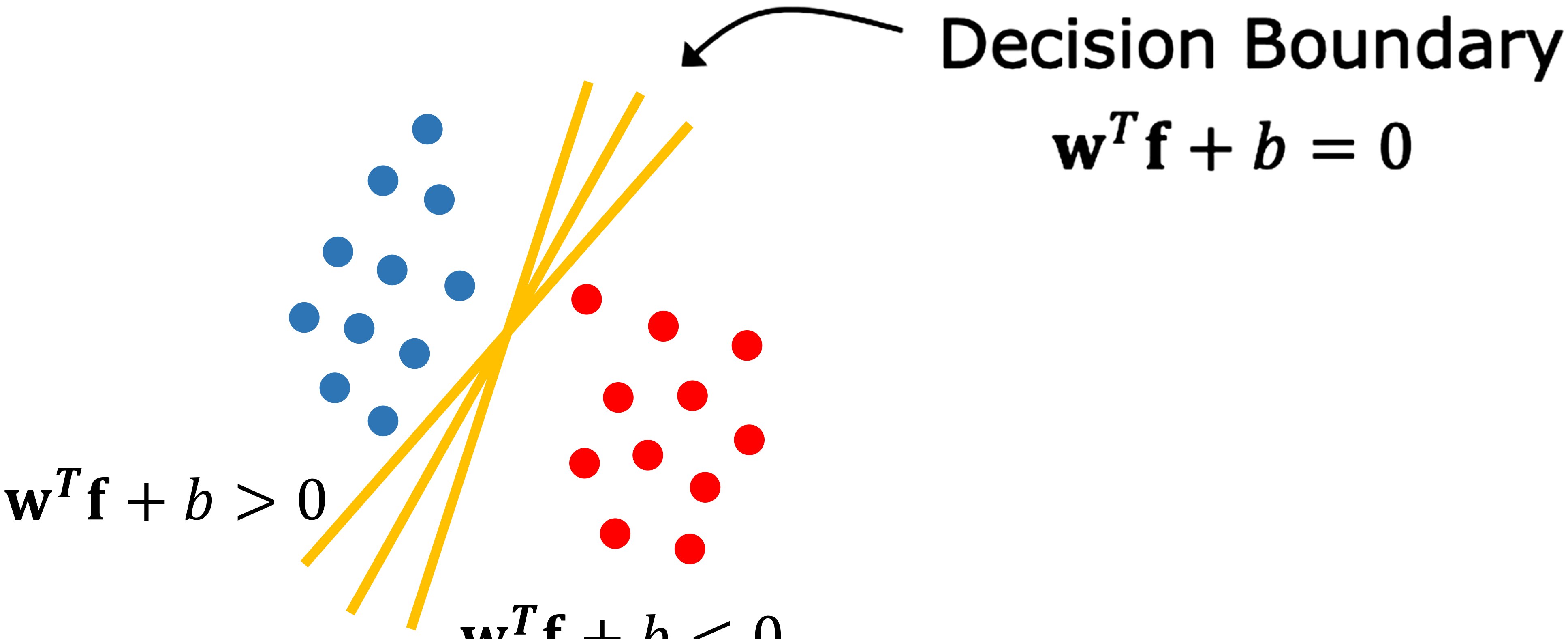
Equation of Hyperplane:

$$w_1 f_1 + w_2 f_2 + \cdots + w_n f_n + b = 0$$

$$\mathbf{w}^T\mathbf{f} + b = 0$$

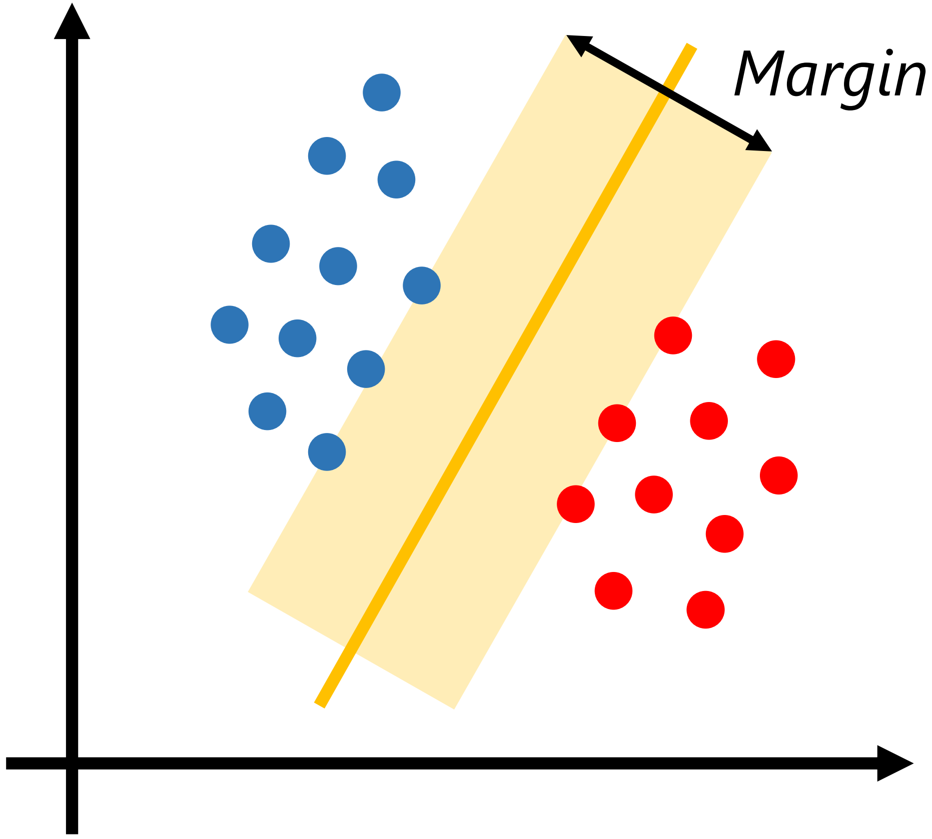
Decision Boundary ($\mathbf{w}$, b)

What is the **optimal** decision boundary?



Evaluating a Decision Boundary

Margin or **safe zone**: The width that the boundary could be increased by, before hitting a feature point.

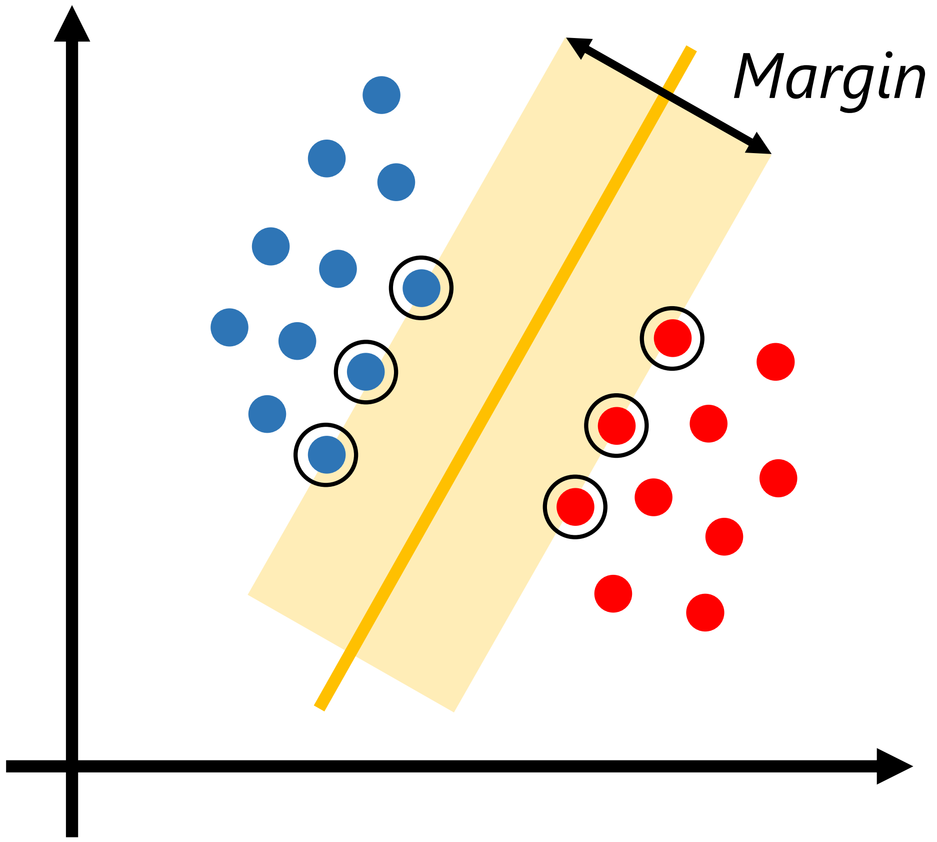


→ Choose decision boundary with **maximum** margin!

Support Vector Machine (SVM)

Classifier optimized to **maximize** margin.

Support vectors: **Closest** data samples to the boundary.



→ Decision boundary & margin are **only** dependent on support vectors!

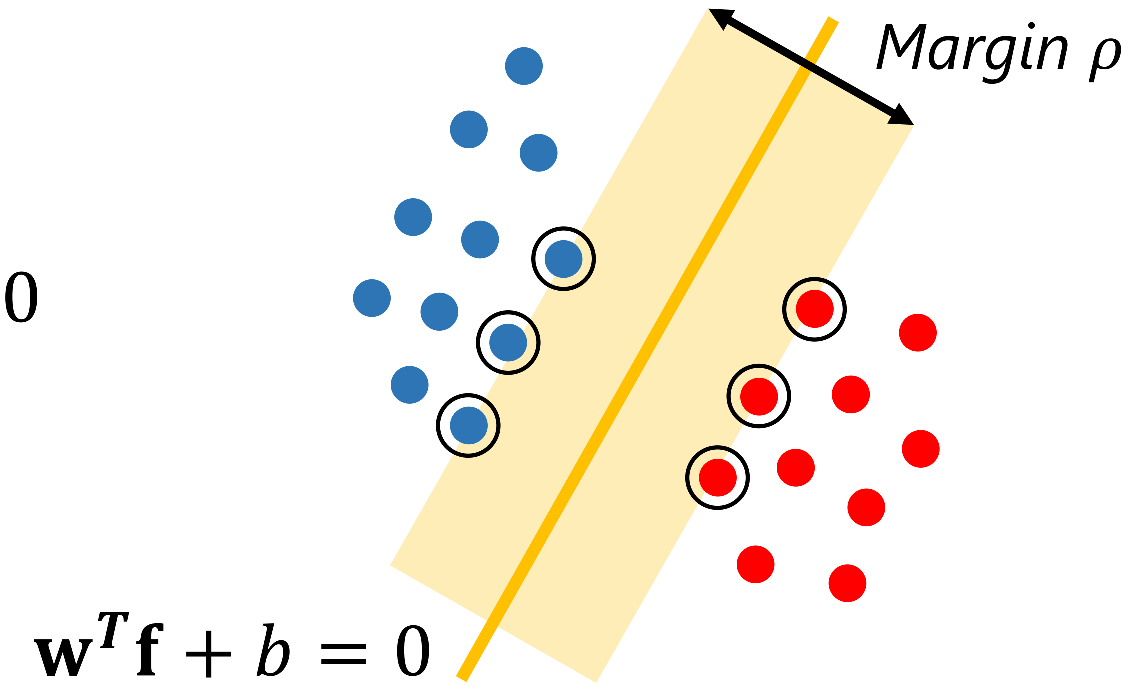
Support Vector Machine (SVM)

Given:

- k training images $\{I_1, I_2, \dots, I_k\}$ and their Haar features $\{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_k\}$.
- k corresponding labels $\{\lambda_1, \lambda_2, \dots, \lambda_k\}$, where $\lambda_j = +1$ if I_j is a face and $\lambda_j = -1$ if I_j is a face.

Find:

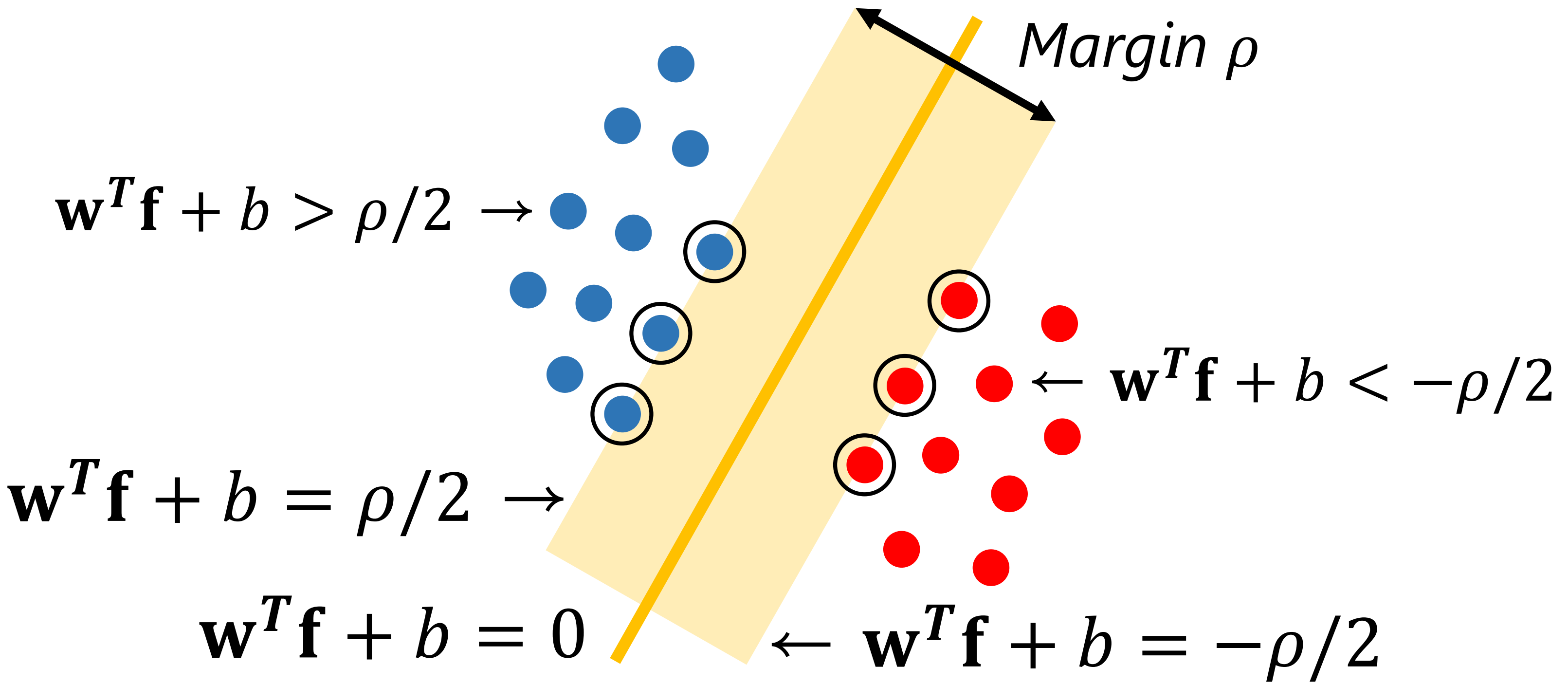
- Decision boundary $\mathbf{w}^T\mathbf{f} + b = 0$
- with maximum margin ρ



Finding Decision Boundary $(\mathbf{w}, b)$

For each training sample $(\mathbf{f}_i, \lambda_i)$:

$$\left.\begin{array}{ll} \text{If } \lambda_i = +1: & \mathbf{w}^T\mathbf{f}_i + b \geq \rho/2 \\ \text{If } \lambda_i = -1: & \mathbf{w}^T\mathbf{f}_i + b \leq -\rho/2 \end{array}\right\} \quad \boxed{\lambda_i(\mathbf{w}^T\mathbf{f}_i + b) \geq \rho/2}$$



Finding Decision Boundary ($\mathbf{w}$, b)

For each training sample $(\mathbf{f}_i, \lambda_i)$:

$$\left.\begin{array}{ll} \text{If } \lambda_i = +1: & \mathbf{w}^T\mathbf{f}_i + b \geq \rho/2 \\ \text{If } \lambda_i = -1: & \mathbf{w}^T\mathbf{f}_i + b \leq -\rho/2 \end{array}\right\} \quad \boxed{\lambda_i(\mathbf{w}^T\mathbf{f}_i + b) \geq \rho/2}$$

If $\mathcal{S}$ is the set of support vectors,

Then for every support vector $s \in \mathcal{S}$: $\boxed{\lambda_s(\mathbf{w}^T\mathbf{f}_s + b) = \rho/2}$

Numerical methods exist to find
$\mathbf{w}, b$ and $\mathcal{S}$ that maximize ρ

Classification Using SVM

Given: Haar features $\mathbf{f}$ for an image window and SVM parameters $\mathbf{w}, b, \rho, \mathcal{S}$

Classification:

Compute $d = \mathbf{w}^T\mathbf{f} + b$

If:
$$\begin{cases} d \geq \rho/2 & \text{Face} \\ d > 0 \; and \; d < \rho/2 & \text{Probably face} \\ d < 0 \; and \; d > -\rho/2 & \text{Probably non-face} \\ d \leq -\rho/2 & \text{Non-face} \end{cases}$$

Experiments

Face detection & SVM

Updated codes are available in https://view.kentech.ac.kr/f088fa7f-874e-44bc-bd6d-6084b42dfdf7

```
$ python face.py
```

```
$ python svm.py
```