中国科学技术大学
University of Science and Technology of China

Face Detection

张举勇
中国科学技术大学

What is Face Detection?

Locate human faces in images

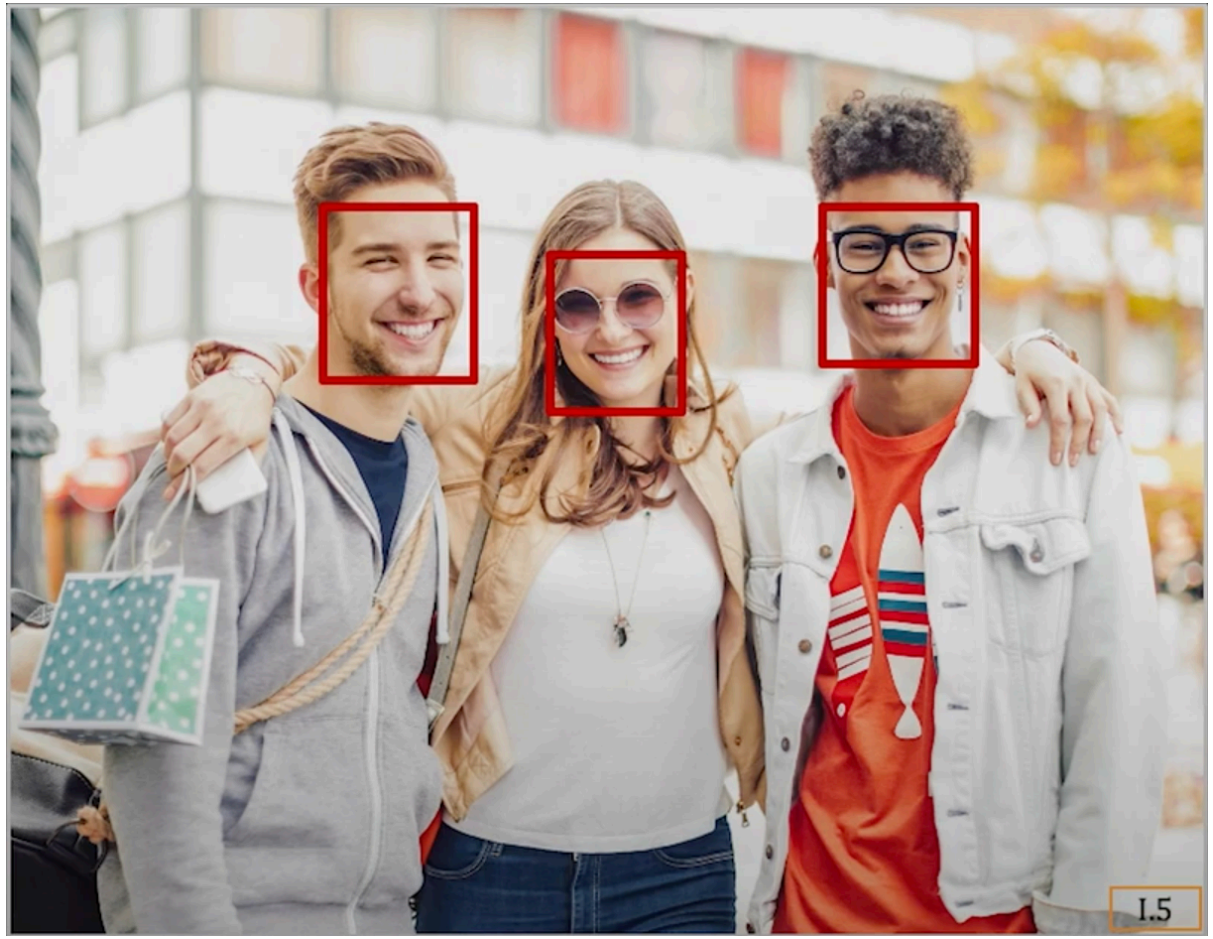


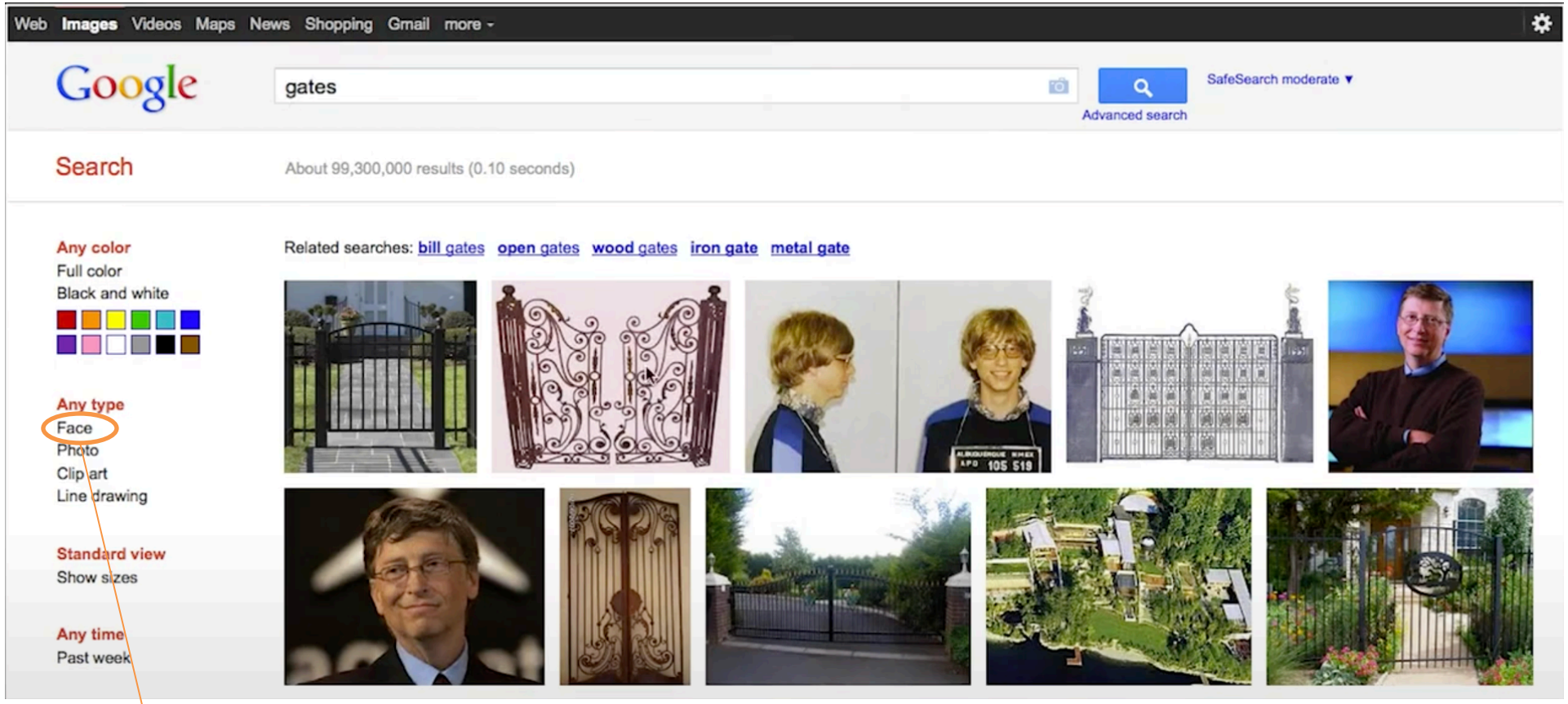
Image Stitching

Locate human faces in images.

Topics:

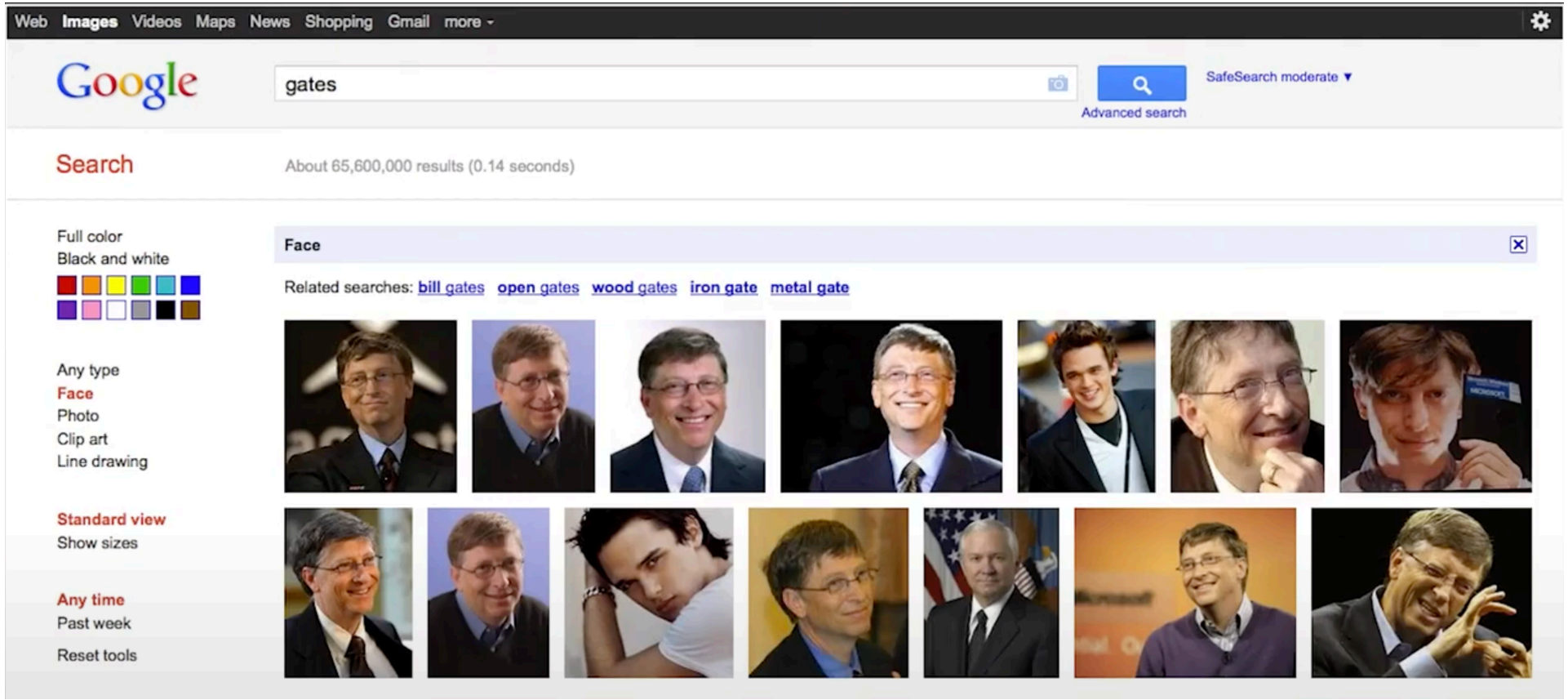
- Uses of Face Detection
- Haar Features for Face Detection
- Integral Image
- Nearest Neighbor Classifier
- Support Vector Machine

Where is Face Detection Used?



Finding People using Search Engines

Where is Face Detection Used?



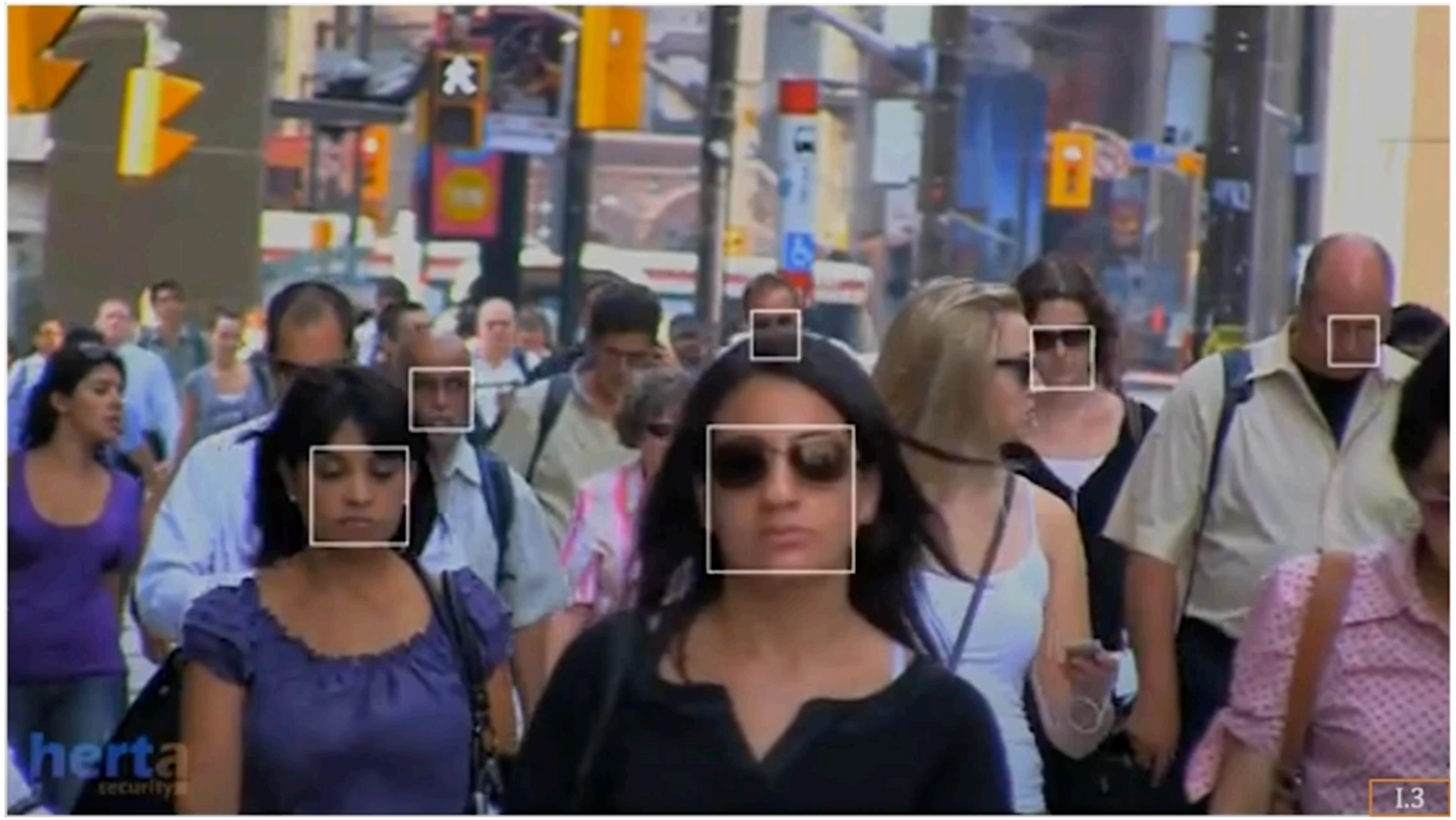
Only faces of people named “Gates”

Finding People using Search Engines

Where is Face Detection Used?

Intelligent Marketing

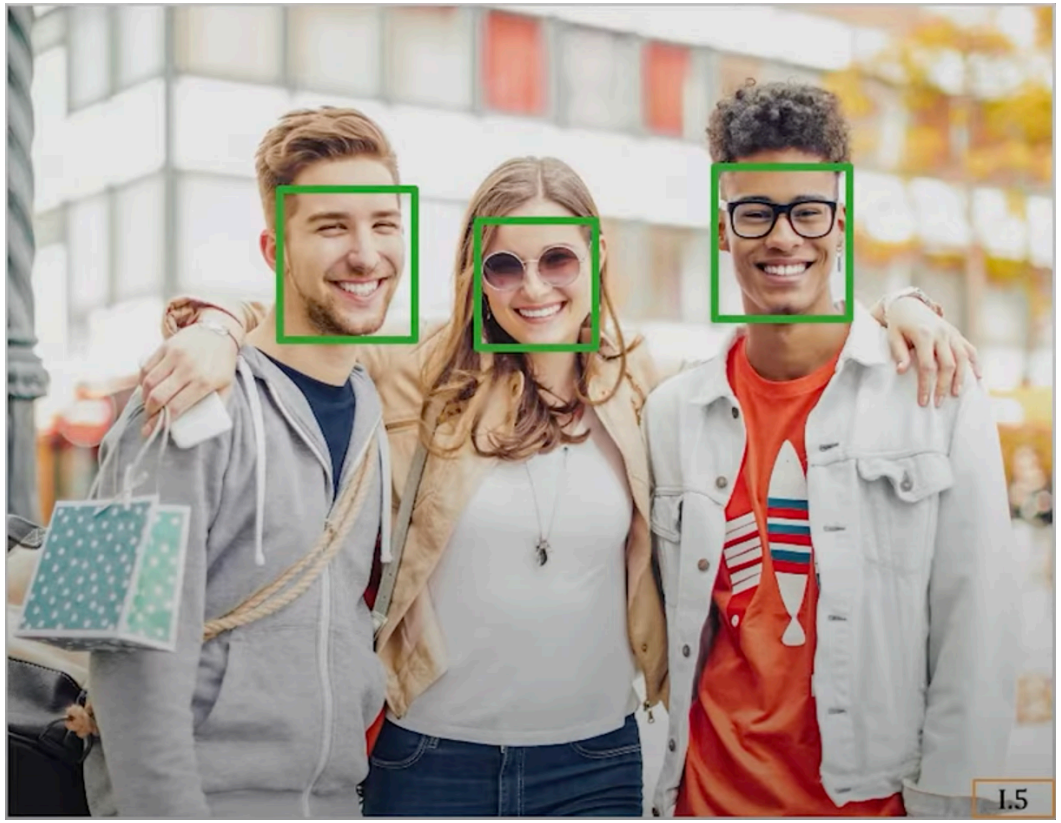
Where is Face Detection Used?



Biometrics, Surveillance, Monitoring

Face Detection in Computers

Slide windows of different sizes across image.
At each location match window to face model.



Face Detection Framework

For each window:

Extract Features → [f] → Match Face Model → Yes / No

Features: Which features represent faces well?

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

What are Good Features?

Interest Points (Edges, Corners, SIFT)?

Facial Components (Templates)?

Charateristics of Good Features

Discriminate Face/Non-Face

Extremely Fast to Compute

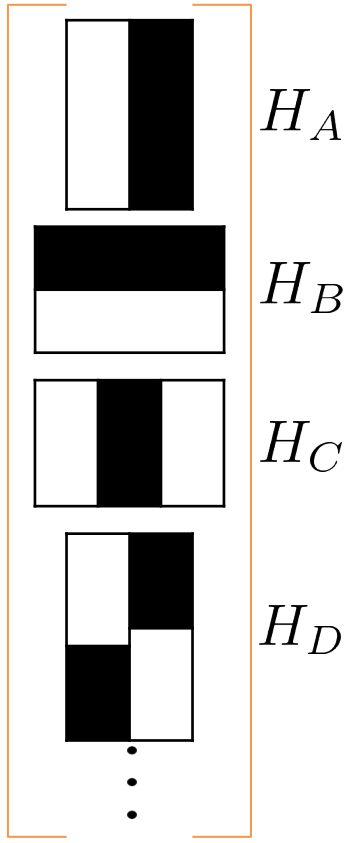
Need to evaluate millions of windows in an image

Haar Features

Set of Correlation Responses to Haar Filters

Input Image

$\otimes$



Haar Filters

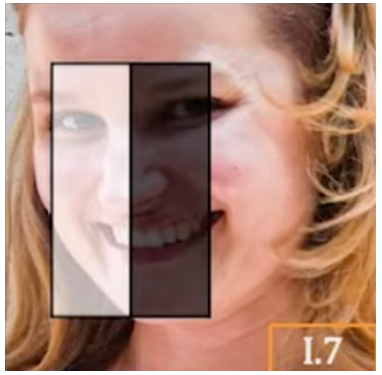
$=$

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

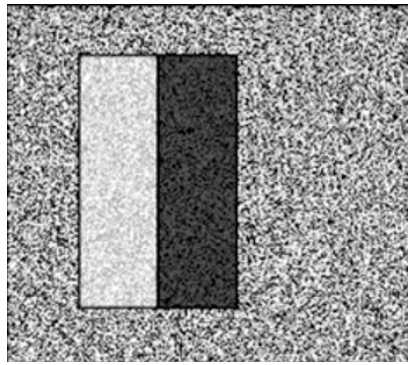
Haar Features

$\mathbf{f}[i,j]$

Discriminative Ability of Haar Feature

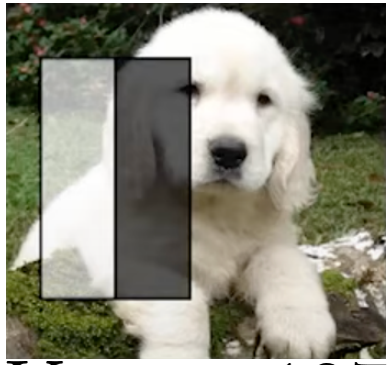


$V_A = 64$



$V_A \approx 0$

$V_A = 16$

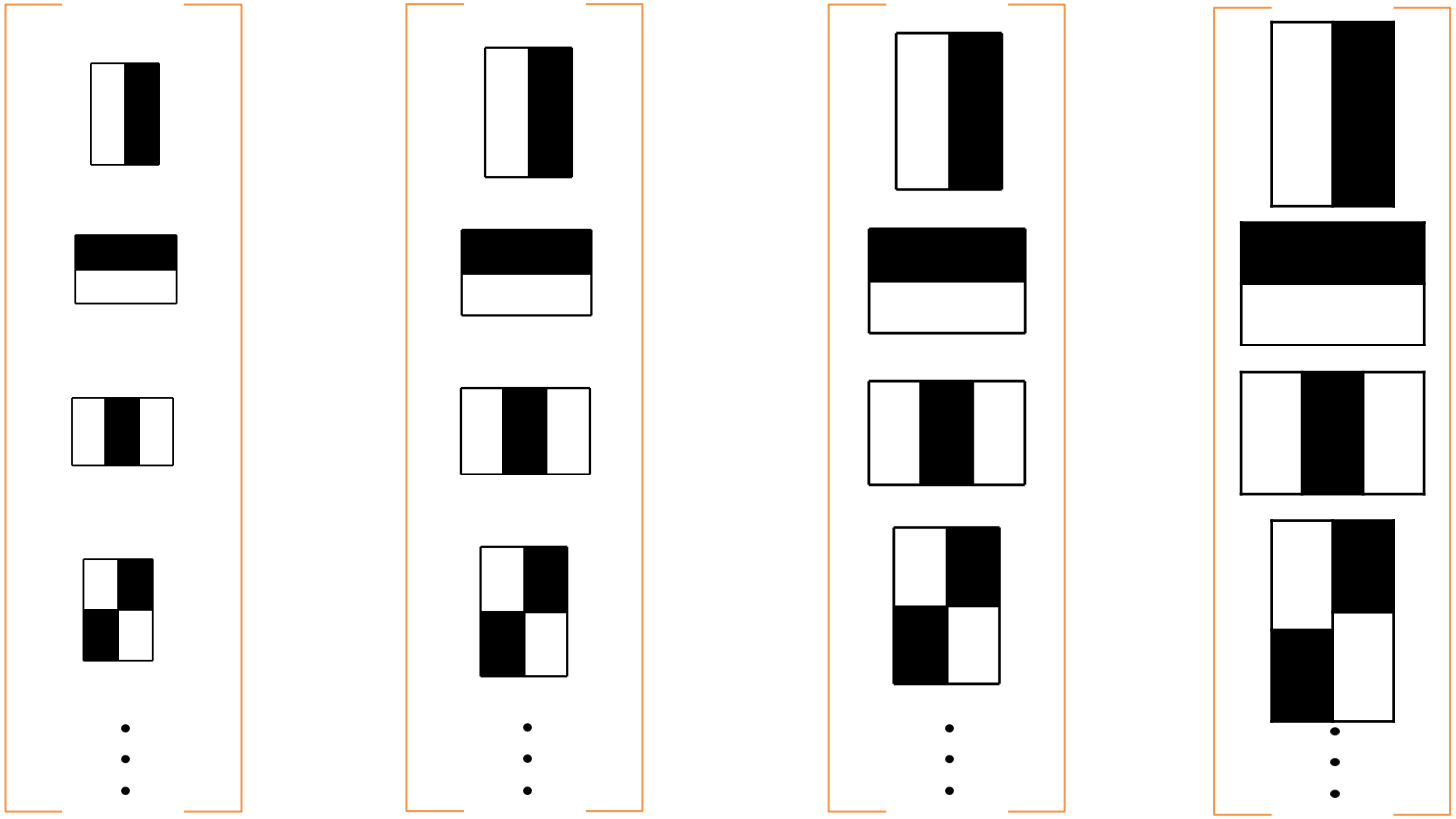


$V_A = -127$

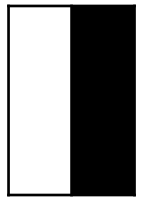
Haar Features are Sensitive to Directionality of Patterns

Haar Features

Compute Haar Features at different scales to detect faces of different sizes.



Haar Features

$\otimes$  H_A

White $= 1$, Black $= -1$

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

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

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

Haar Features: Computation Cost

$$Value = \sum(pixel\ intensities\ in\ white\ area) - \sum(pixels\ intensities\ in\ black\ area)$$

Computation cost = $(N \times M - 1)$ additions per pixel per filter per scale.

Can We Do Better?

Integral Image

A table that holds the sum of all pixel values to the left and top of a given pixel, inclusive.

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

Integral Image II

Integral Image

A table that holds the sum of all pixel values to the left and top of a given pixel, inclusive.

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

Integral Image II

Integral Image

A table that holds the sum of all pixel values to the left and top of a given pixel, inclusive.

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

Integral Image II

Summation Within a Rectangle

Fast summations of arbitrary rectangles using integral images

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

Integral Image II

Summation Within a Rectangle

Fast summations of arbitrary rectangles using integral images

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

P

Integral Image II

$$Sum = II_P + \ldots$$
$$= 3490 + \ldots$$

Summation Within a Rectangle

Fast summations of arbitrary rectangles using integral images

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

Q

P

Integral Image II

$$Sum = II_P - II_Q + \ldots$$
$$= 3490 - 1137 + \ldots$$

Summation Within a Rectangle

Fast summations of arbitrary rectangles using integral images

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

Q

S

P

Integral Image II

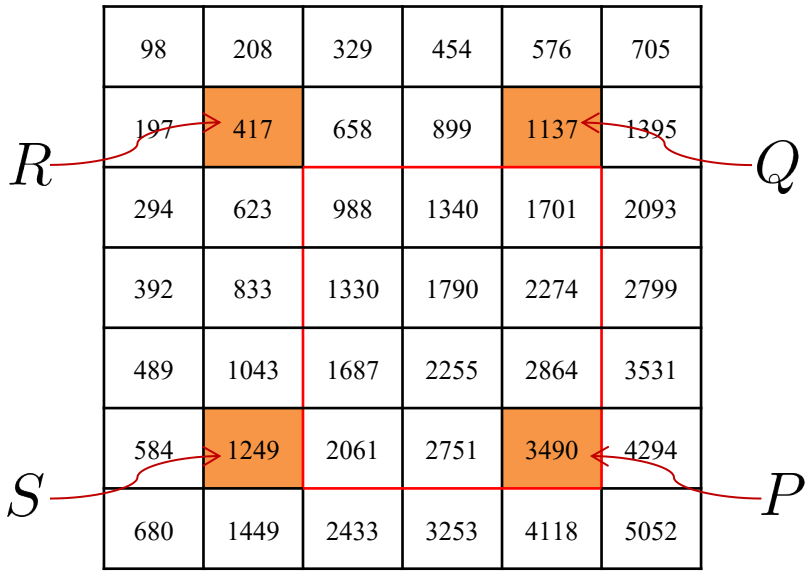
$$Sum = II_P - II_Q - II_S + \ldots$$
$$= 3490 - 1137 - 1249 + \ldots$$

Summation Within a Rectangle

Fast summations of arbitrary rectangles using integral images

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I



Integral Image II

$$Sum = II_P - II_Q - II_S + II_R$$
$$= 3490 - 1137 - 1249 + 417 = 1521$$

Computation Cost: Only 3 additions

Haar Response using Integral Image

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

98	208	329	454	576	705
197	417	658	899	1137	1395
294	623	988	1340	1701	2093
392	833	1330	1790	2274	2799
489	1043	1687	2255	2864	3531
584	1249	2061	2751	3490	4294
680	1449	2433	3253	4118	5052

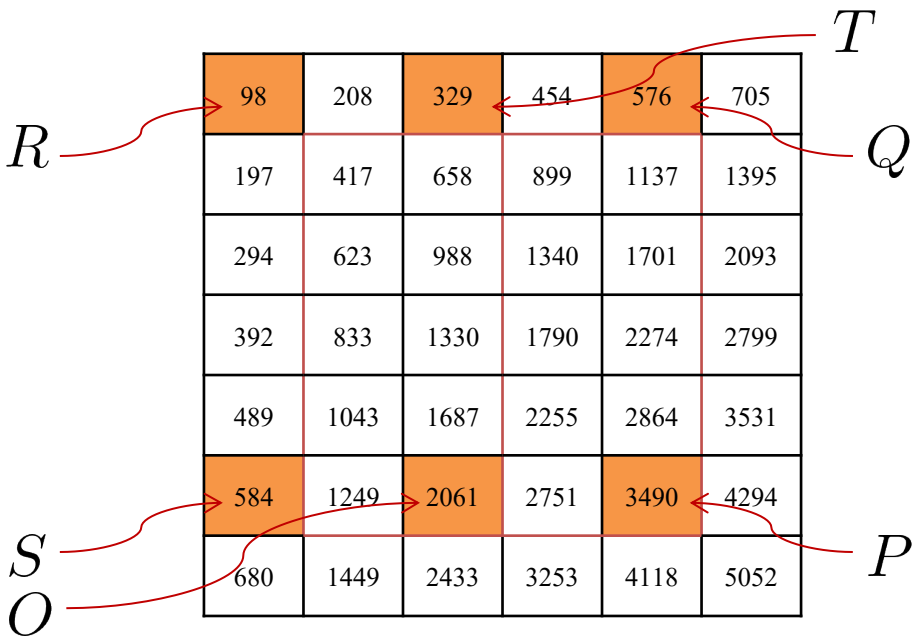
Integral Image II

$$V_A = \sum(pixels\ in\ white) - \sum(pixels\ in\ black)$$

Haar Response using Integral Image

98	110	121	125	122	129
99	110	120	116	116	129
97	109	124	111	123	134
98	112	132	108	123	133
97	113	147	108	125	142
95	111	168	122	130	137
96	104	172	130	126	130

Image I

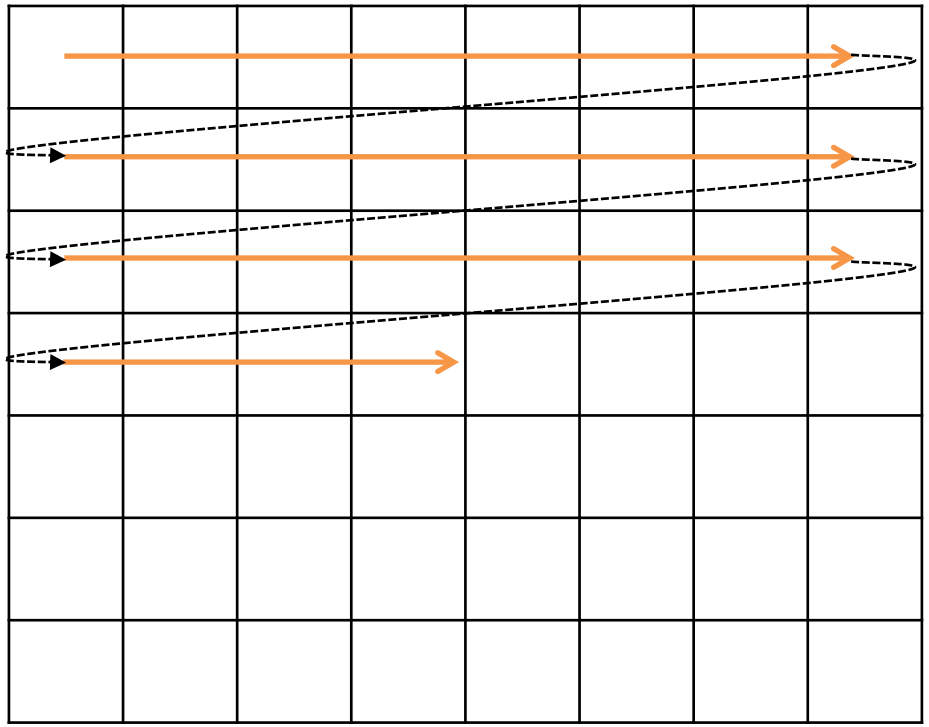


Integral Image II

$$V_A = \sum(pixel\ intensities\ in\ white) - \sum(pixel\ intensities\ in\ black)$$
$$= (II_O - II_T + II_R - II_S) - (II_P - II_Q + II_T - II_O)$$
$$= (2061 - 329 + 98 - 584) - (3490 - 576 + 329 - 2061) = 64$$

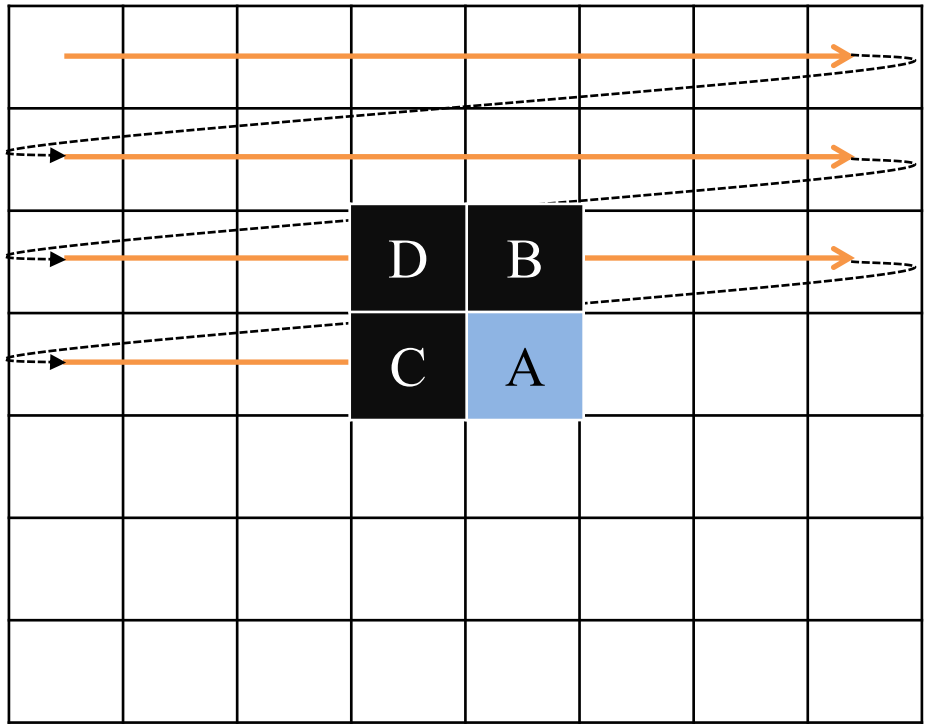
Computation Cost: Only 7 additions

Computing Integral Image



Raster Scanning

Computing Integral Image



Raster Scanning

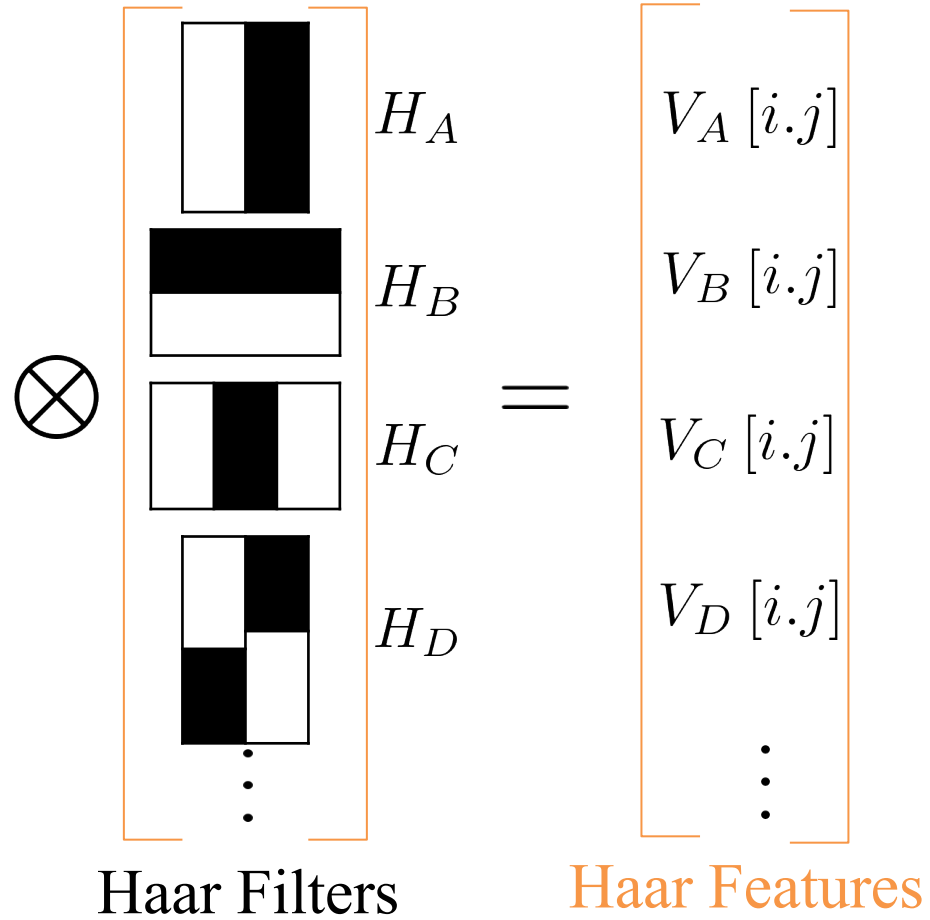
Let I_A and II_A be the values of Image and Integral Image, respectively, at pixel A.

$$II_A = II_B + II_C - II_D + I_A$$

Haar Features using Integral Images

Integral image needs to be computed once per test image.
Allows fast computations of Haar features.

Input Image

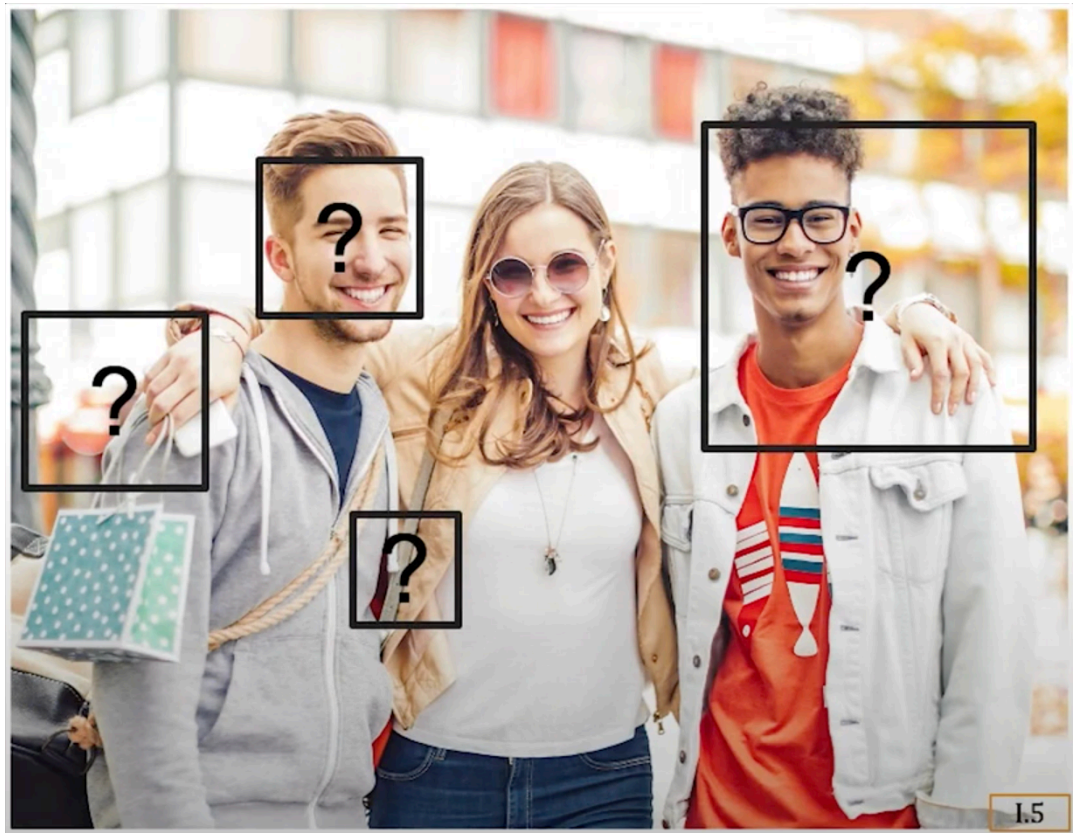


Haar Filters

Haar Features

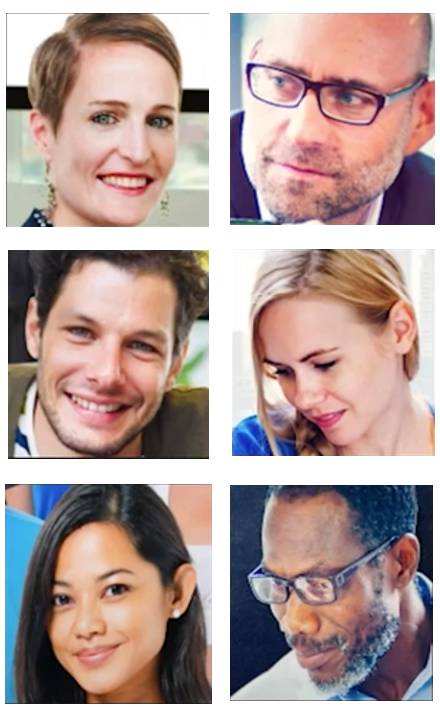
Classifier for Face Detection?

Given the features for a window, how to decide whether it contains a face or not?

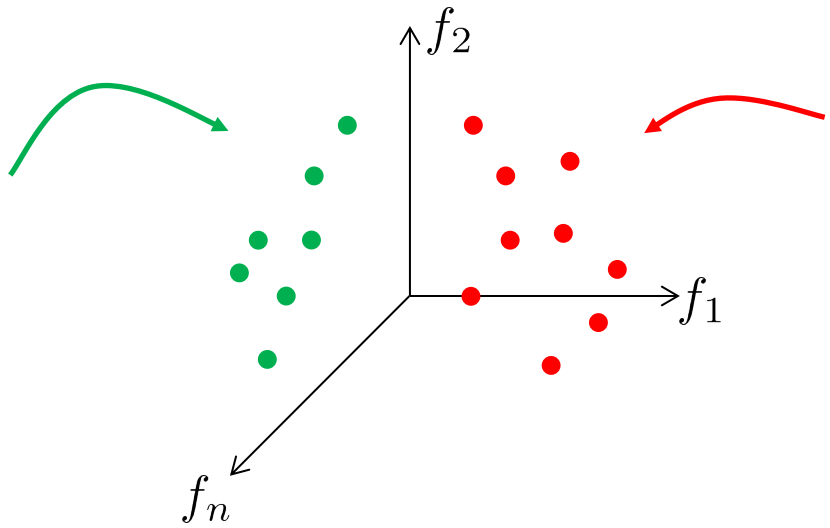


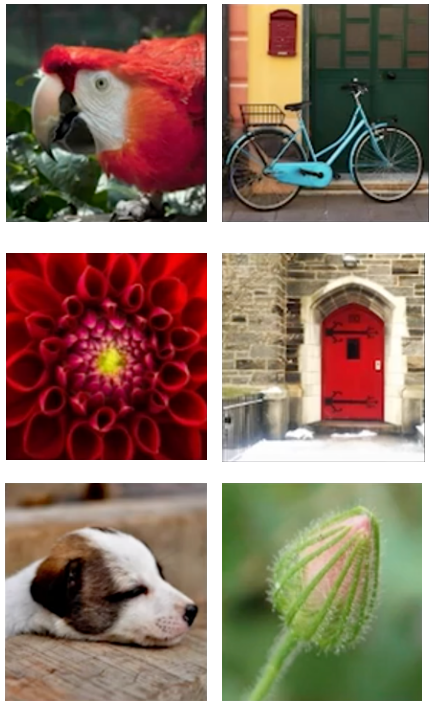
Feature Space

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



Training Data of Face

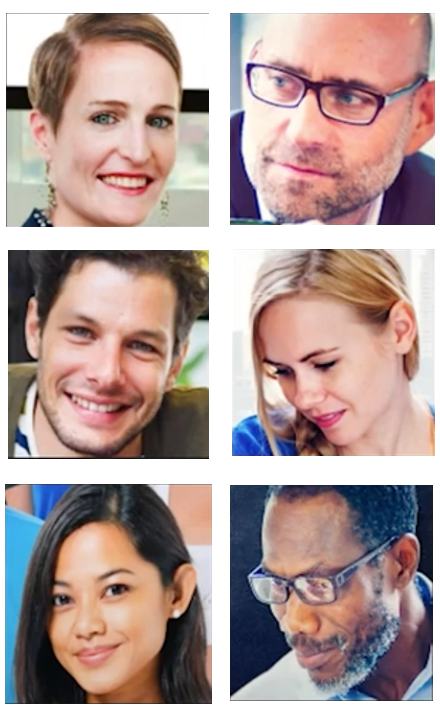




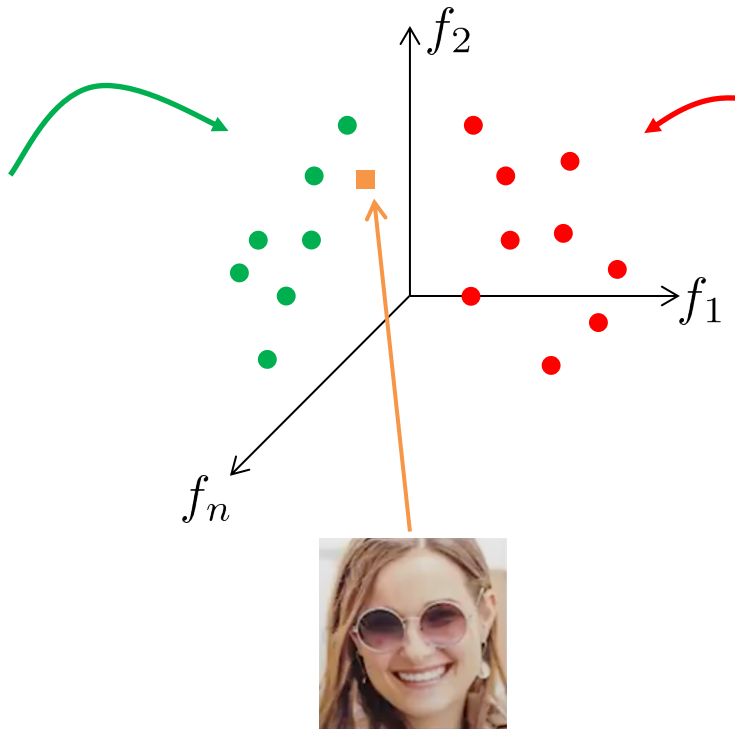
Training Data of Non-Face

Feature Space

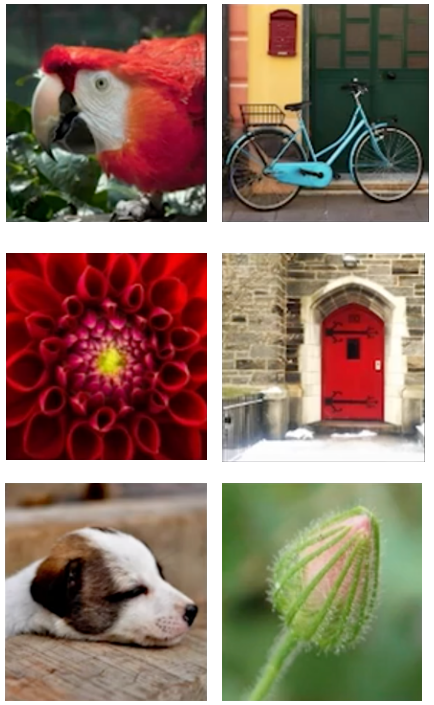
Haar Features **f** (a vector) at a pixel is a point in an n-D space, $\mathbf{f} \in \mathbf{R}^n$



Training Data of Face



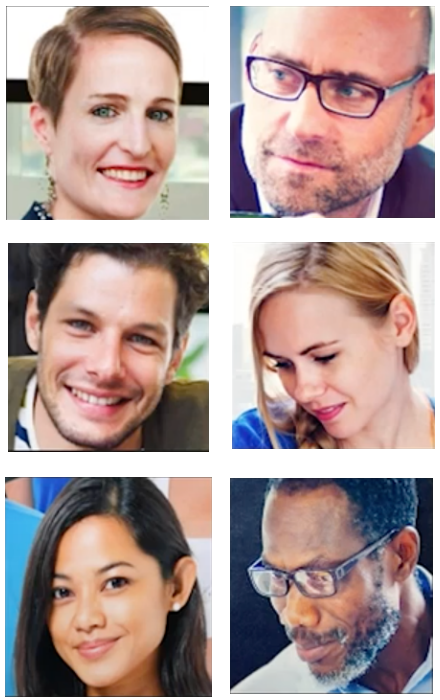
Test Image



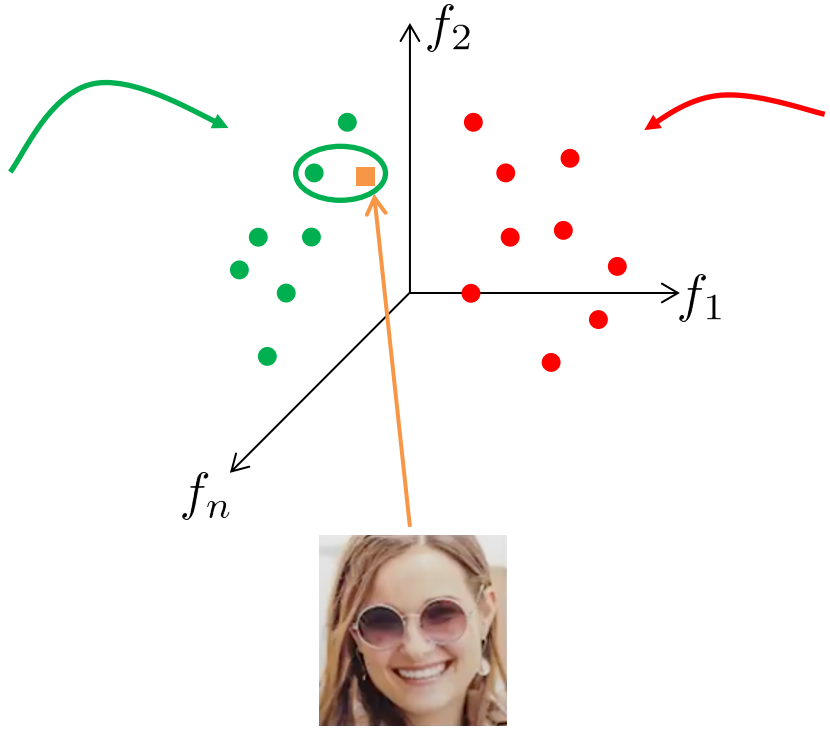
Training Data of Non-Face

Feature Space

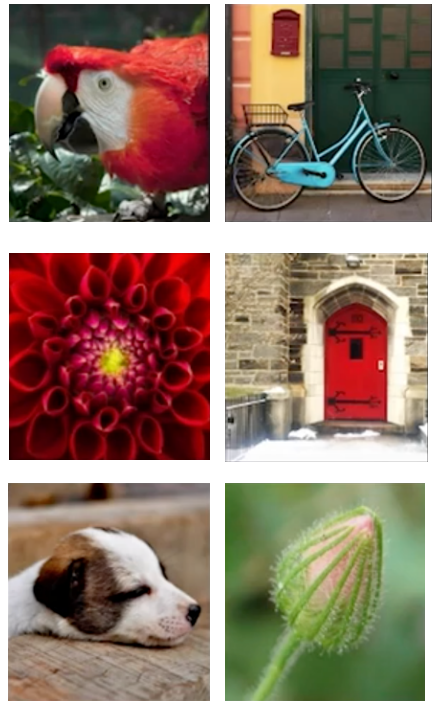
Find the Nearest training sample using L^2 distance and assign its label.



Training Data of Face



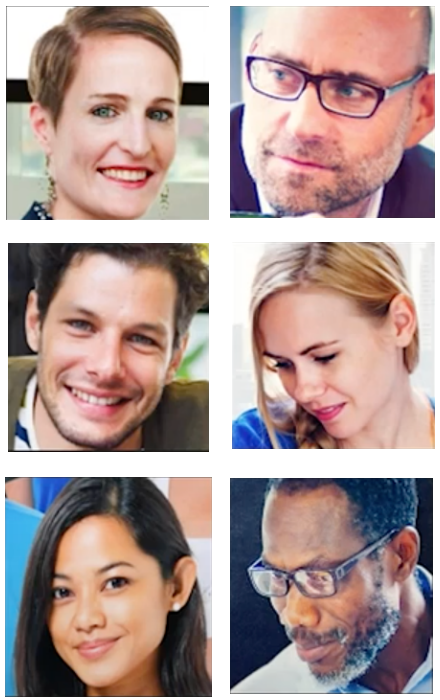
Face



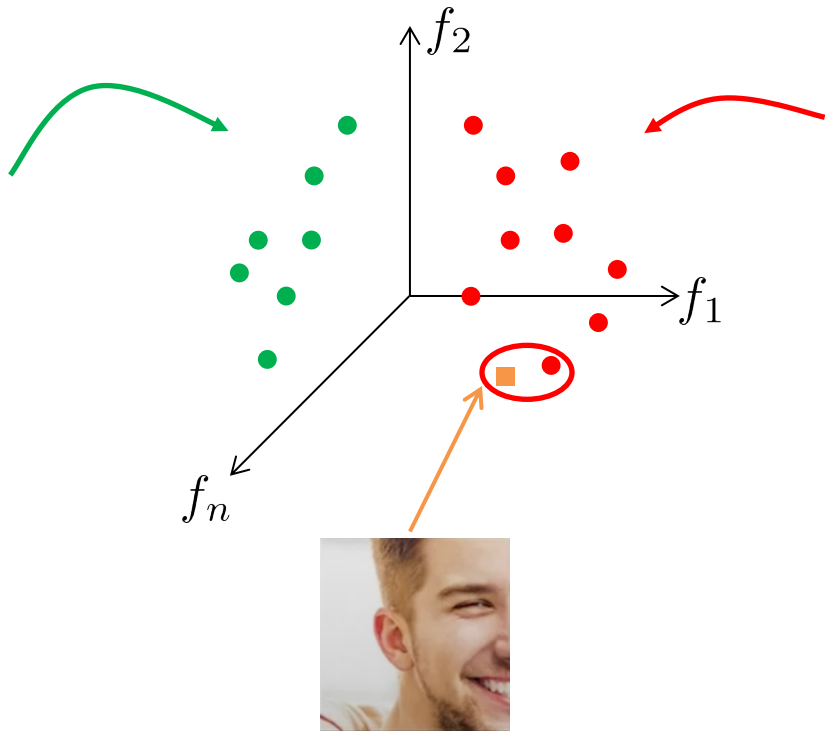
Training Data of Non-Face

Feature Space

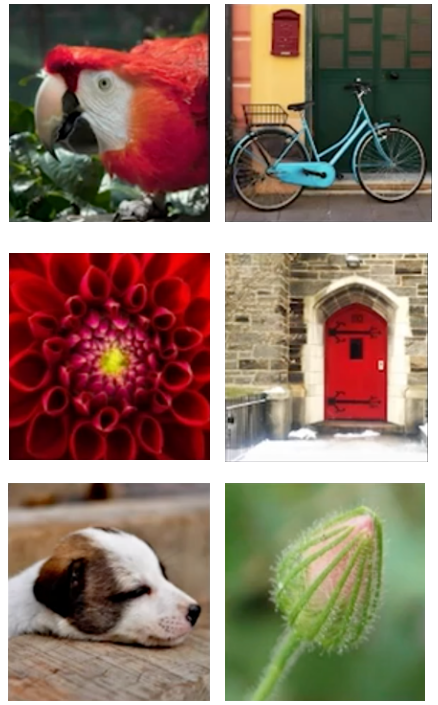
Find the Nearest training sample using L^2 distance and assign its label.



Training Data of Face



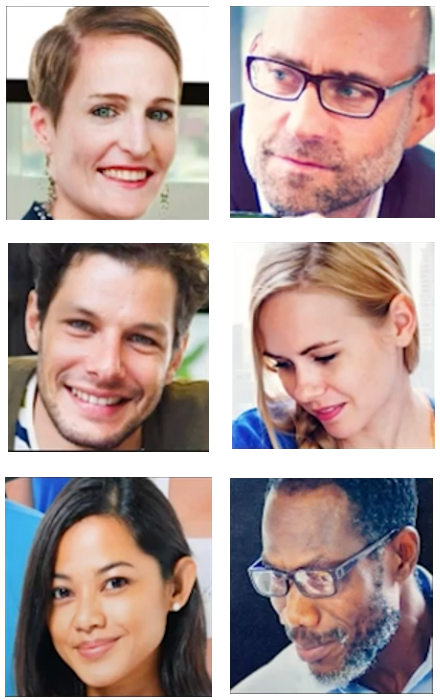
Not Face



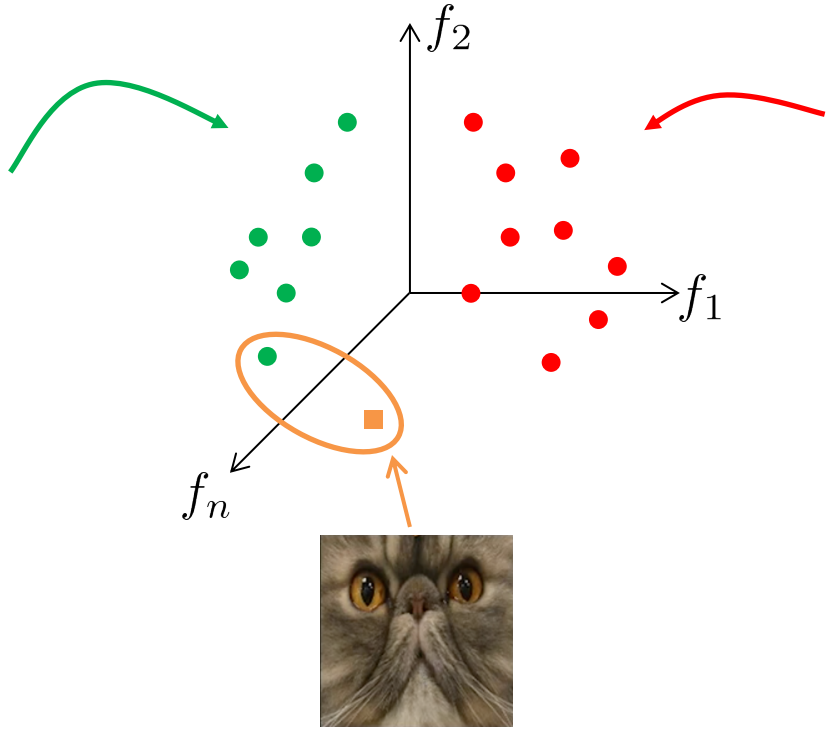
Training Data of Non-Face

Feature Space

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

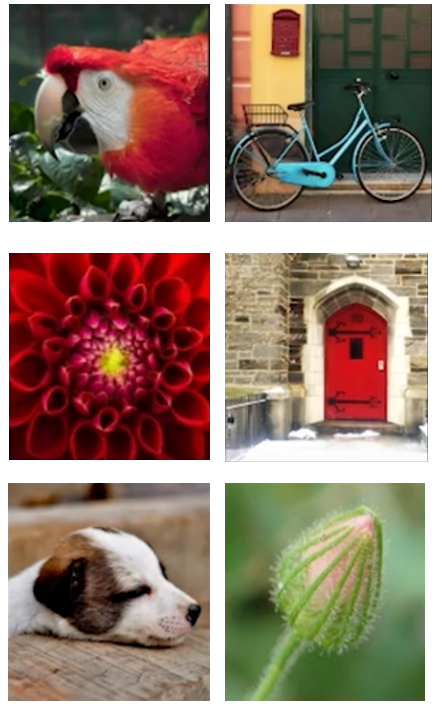


Training Data of Face



Face

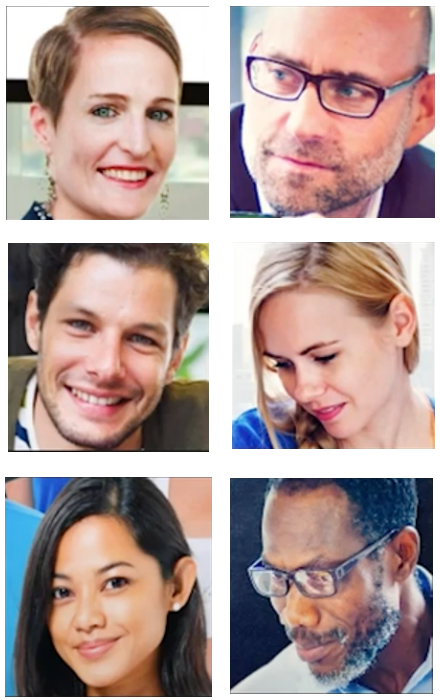
False Positive



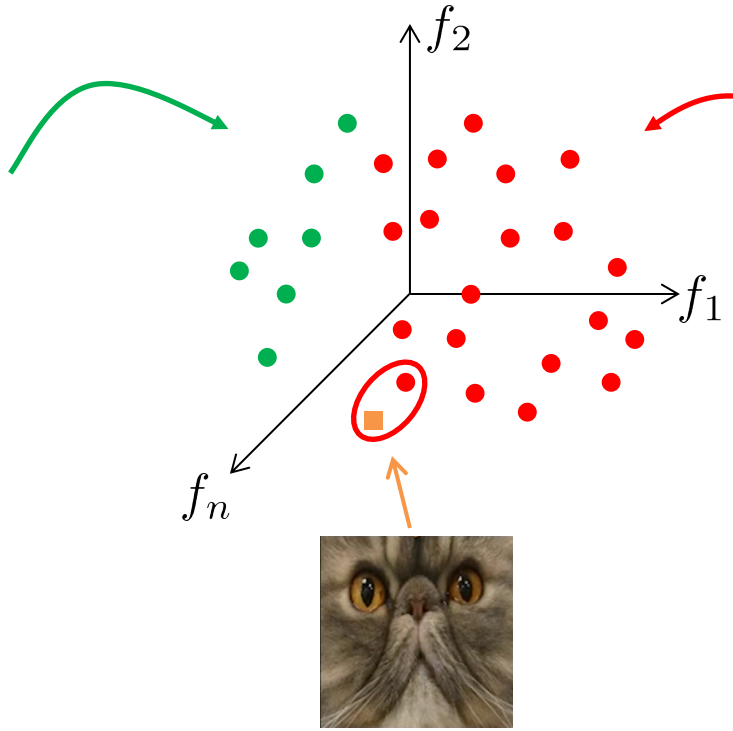
Training Data of Non-Face

Feature Space

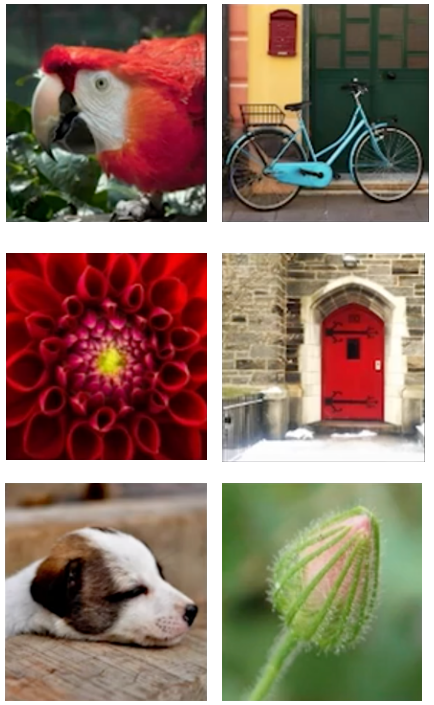
Larger the training set, more robust the NN classifier



Training Data of Face



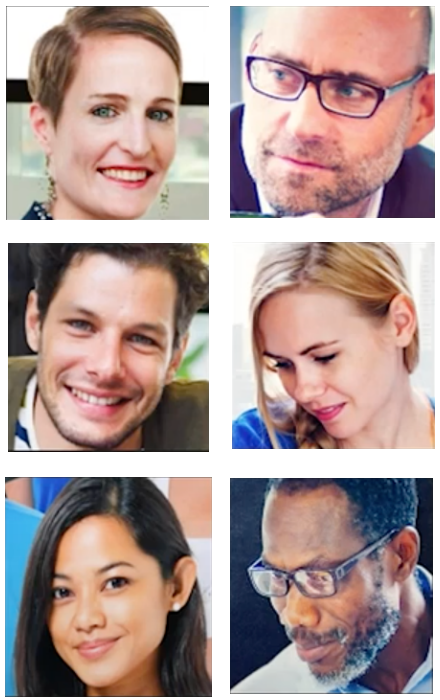
Non-Face



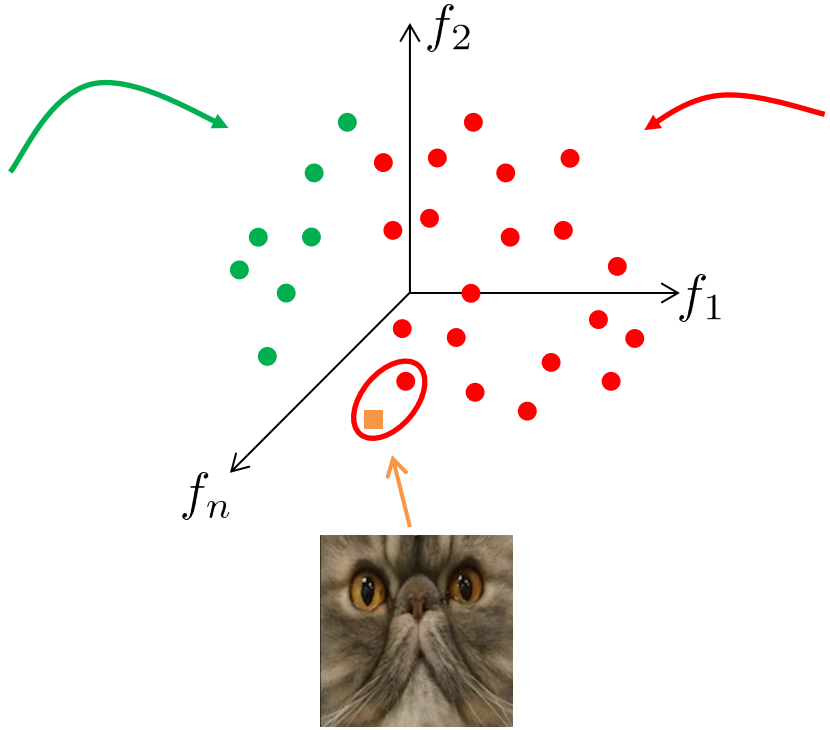
Training Data of Non-Face

Feature Space

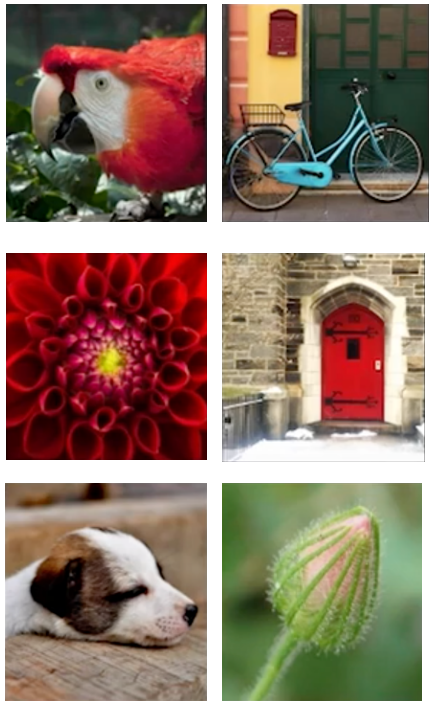
Larger the training set, slower the NN classifier



Training Data of Face



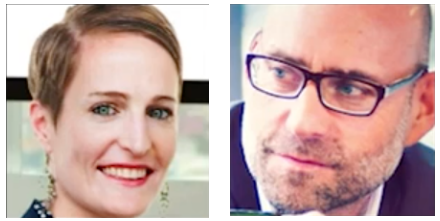
Non-Face

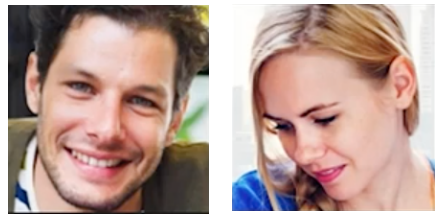


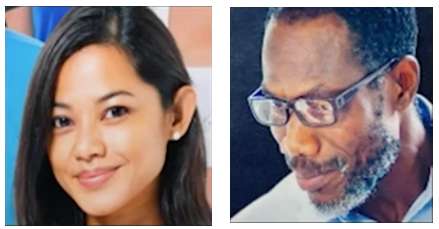
Training Data of Non-Face

Feature Space

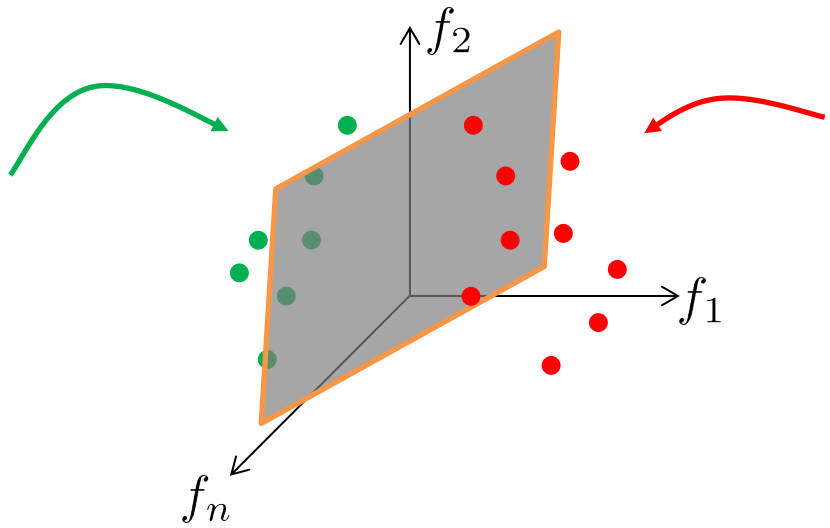
A simple dicision boundary separating face and non-face classes will suffice

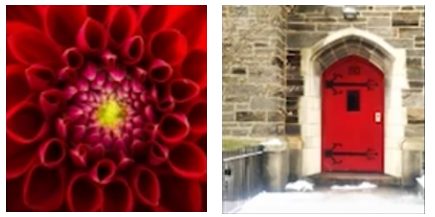






Training Data of Face

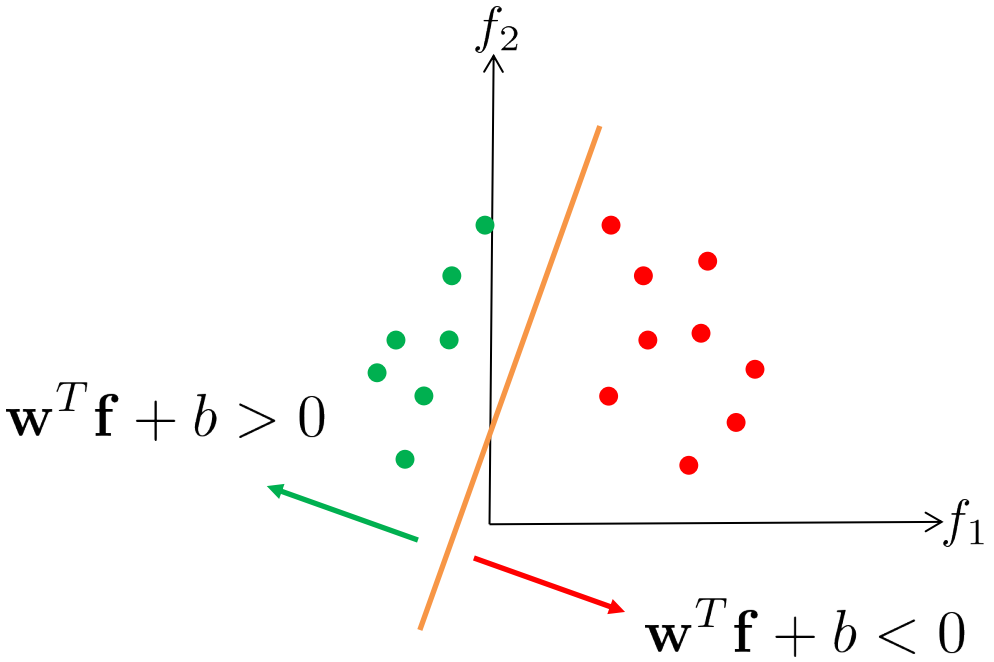




Training Data of Non-Face

Linear Dicision Boundaries

A Linear Decision Boundary in 2-D space is a 1-D Line



Equation of Line:

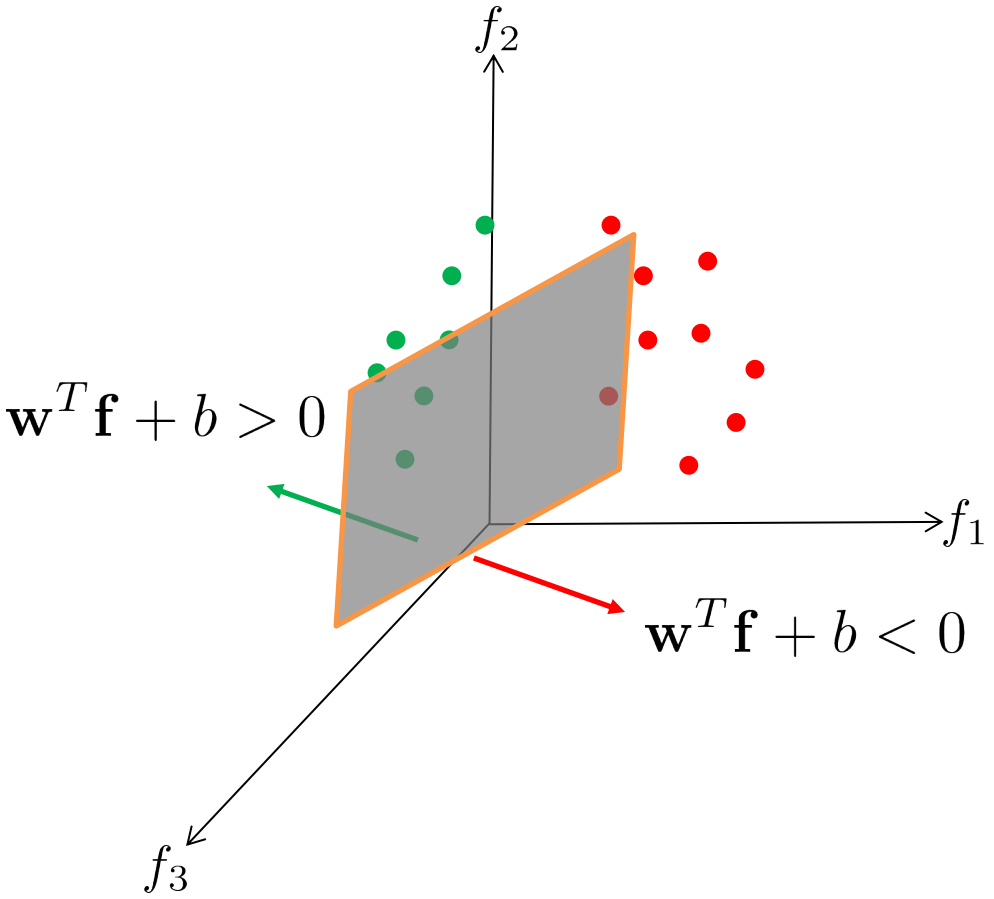
$$w_1 f_1 + w_2 f_2 + b = 0$$

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

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

Linear Dicision Boundaries

A Linear Decision Boundary in 3-D space is a 2-D Plane



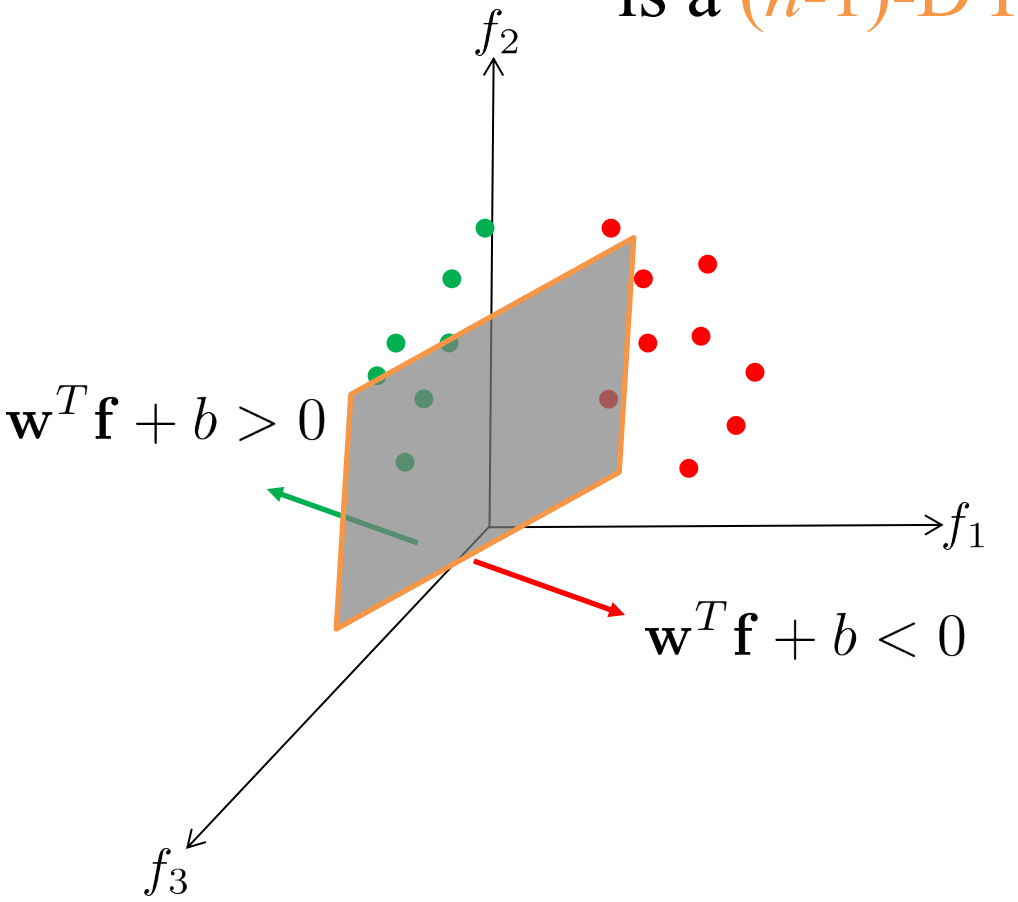
Equation of Plane:

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

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

Linear Dicision Boundaries

A Linear Decision Boundary in *n*-D space
is a (*n*-1)-D Hyperplane



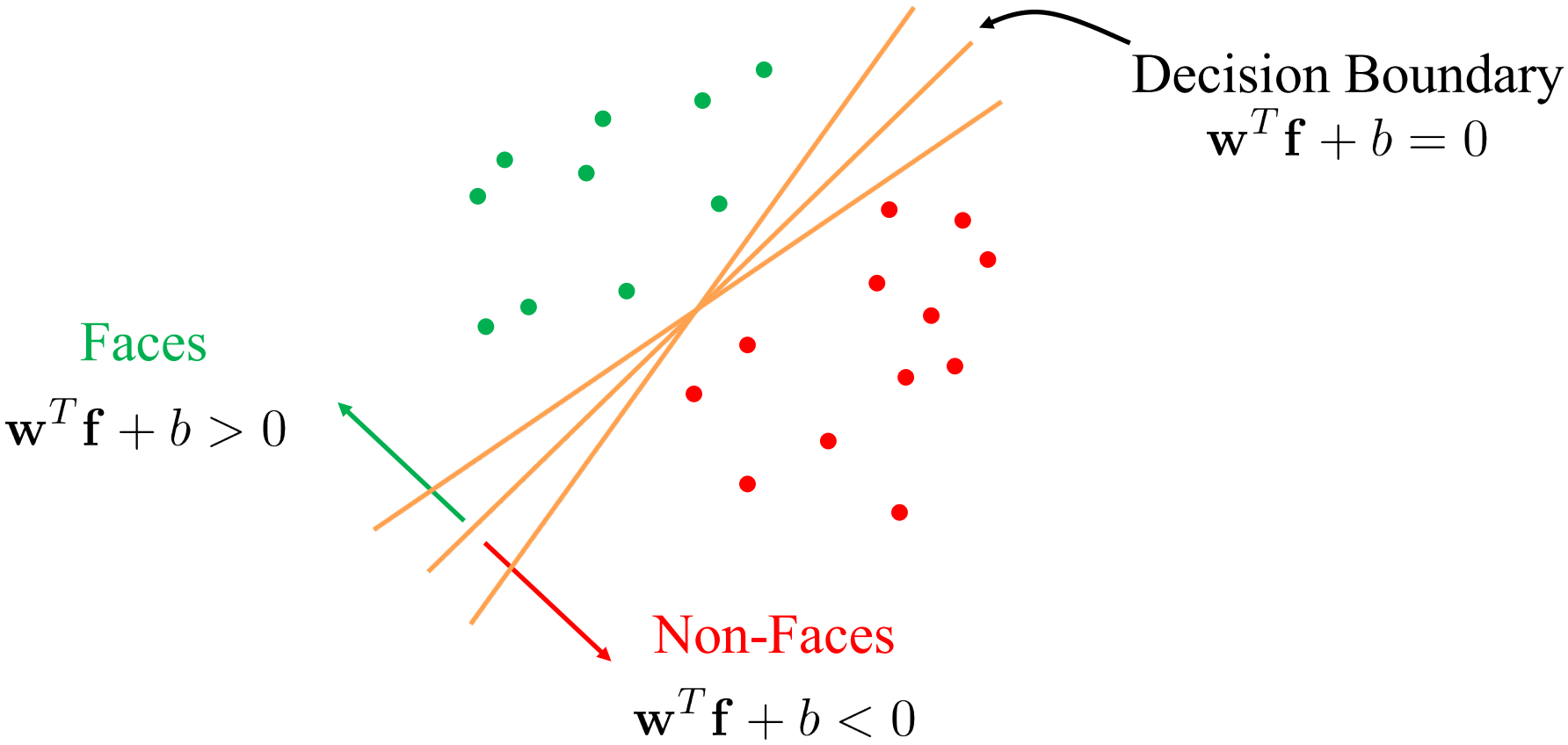
Equation of Hyperplane:

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

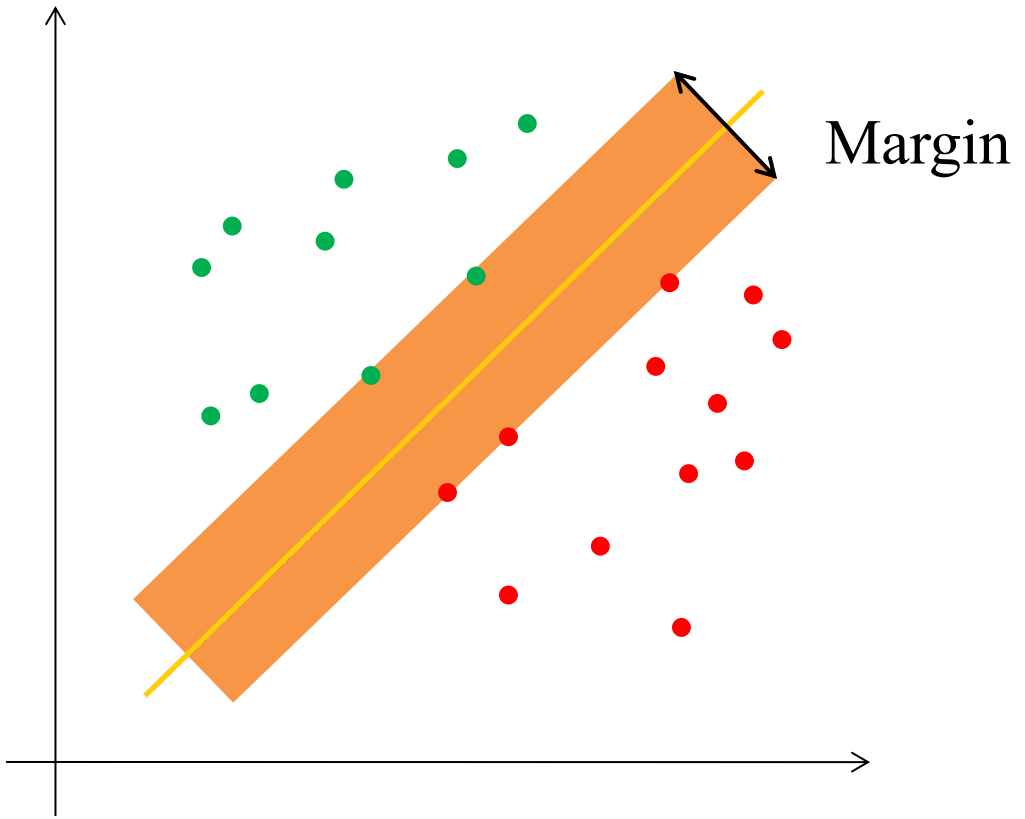
$$\mathbf{w}^T\mathbf{f} + b = 0$$

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

What is the optimal decision boundary?

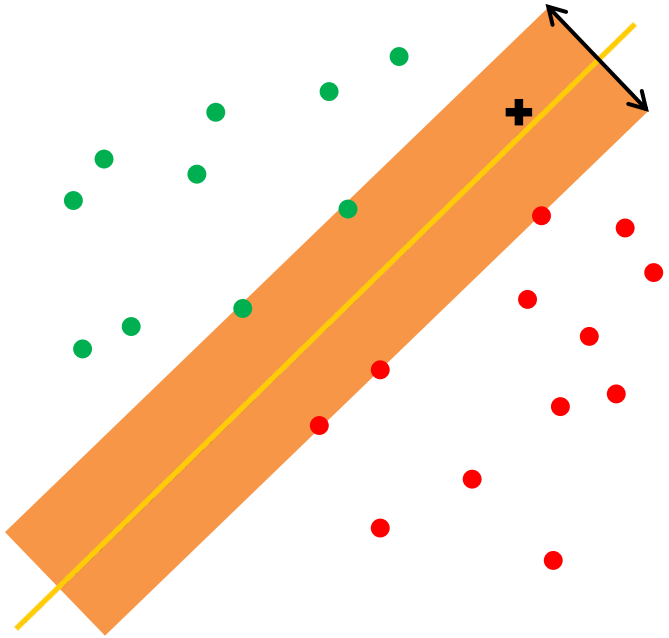


Evaluating a Decision Boundary

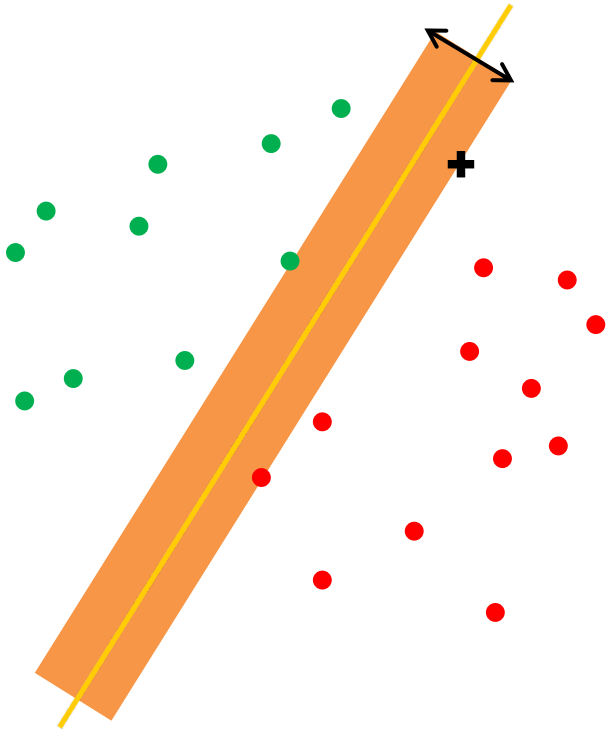


Margin or Safe Zone: The width that the boundary could be increased by, before hitting a feature point.

Evaluating a Decision Boundary



Margin I



Margin II

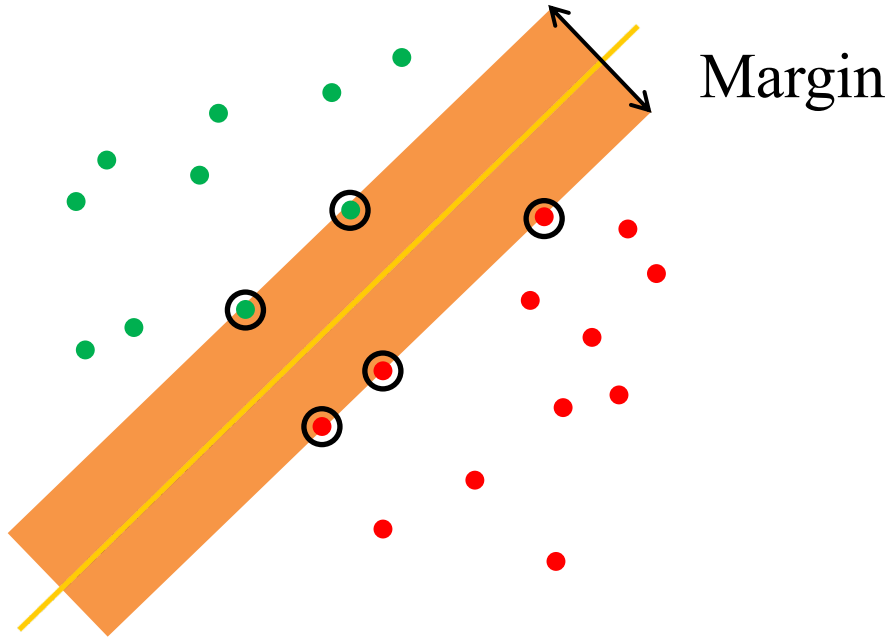
Decision I: Face

Decision II: Non-Face

Choose Decision Boundary with Maximum Margin!

Support Vector Machine (SVM)

Classifier optimized to Maximum Margin



Support Vectors: Closest data samples to the boundary

Decision Boundary & Margin depend only on Support Vectors

Support Vector Machine (SVM)

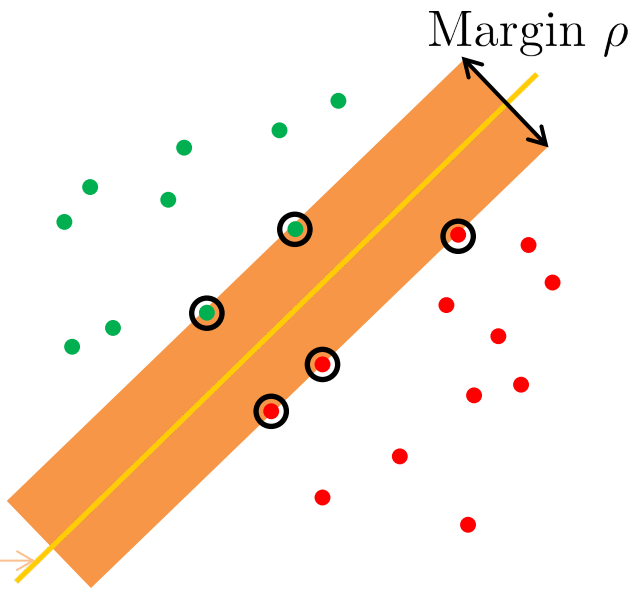
Given:

- k training images $\{I_1, I_2, \ldots, I_k\}$ and their Haar features $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k\}$.
- k corresponding labels $\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$, where $\lambda_j = +1$ if I_j is a face and $\lambda_j = -1$ if I_j is not a face.

Find:

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

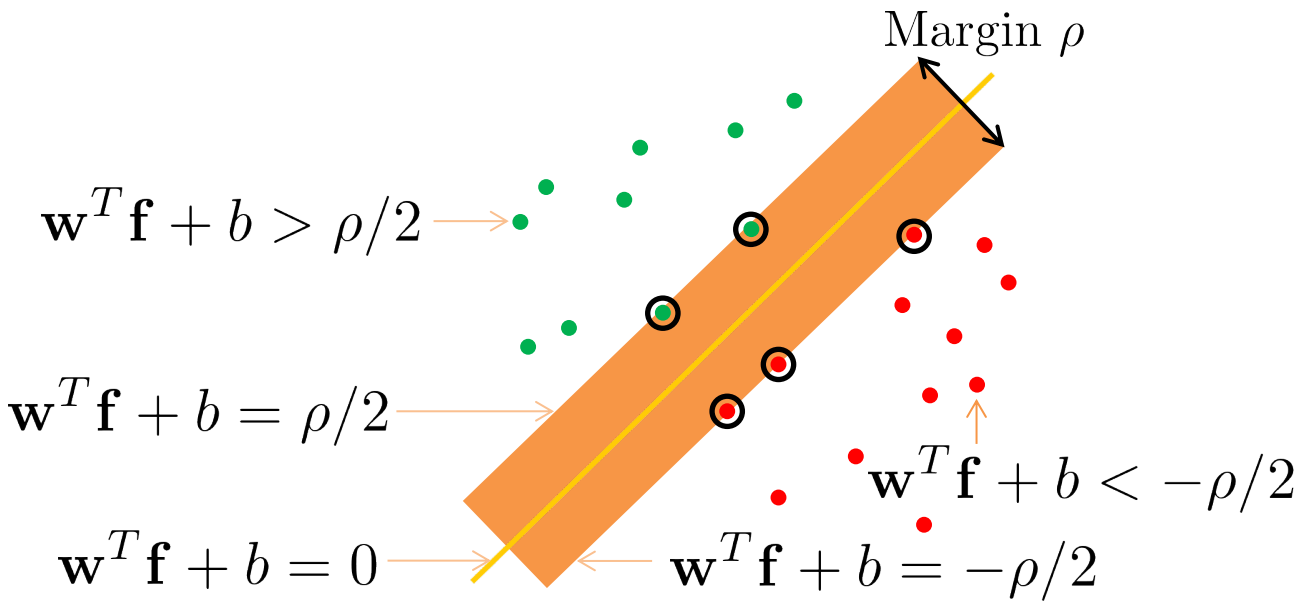
with Maximum Margin ρ



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

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

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

MATLAB: svmtrain

Support Vector Machine (SVM)

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

Classification:

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

If:

$d \geq \rho/2$	Face
$d > 0$ and $d < \rho/2$	Probably Face
$d < 0$ and $d > -\rho/2$	Probably Not-Face
$d \leq -\rho/2$	Not-Face

Face Detection Results

Remarks

- Current face detection systems are mature but not perfect.
- Frontal and side poses usually require different face models.
- Successful vision technology used in cameras, surveillance, biometrics, search.
- Performance continues to improve.