

# Algorithme de détection de visage de Viola & Jones

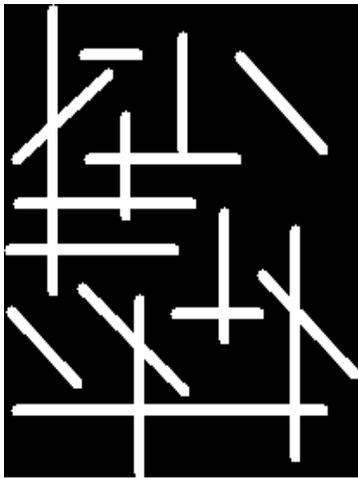
D'après Ramsri Goutham Golla, ASU

# Algorithme de Viola & Jones

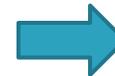
- Caractéristiques pseudo-Haar
- Image intégrale
- Adaboost
- Cascade de classifieurs

# Caractéristiques pseudo-Haar

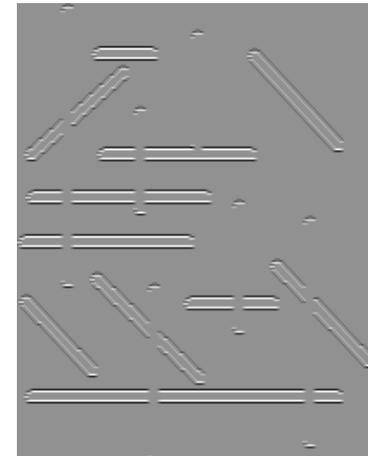
Rappel : convolution

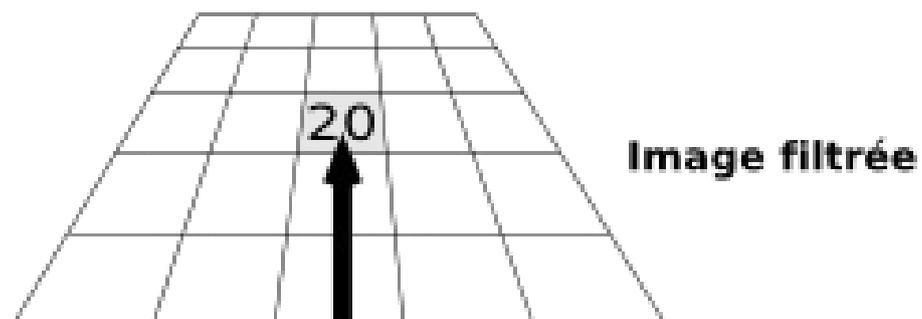


-1	-1	-1
2	2	2
-1	-1	-1

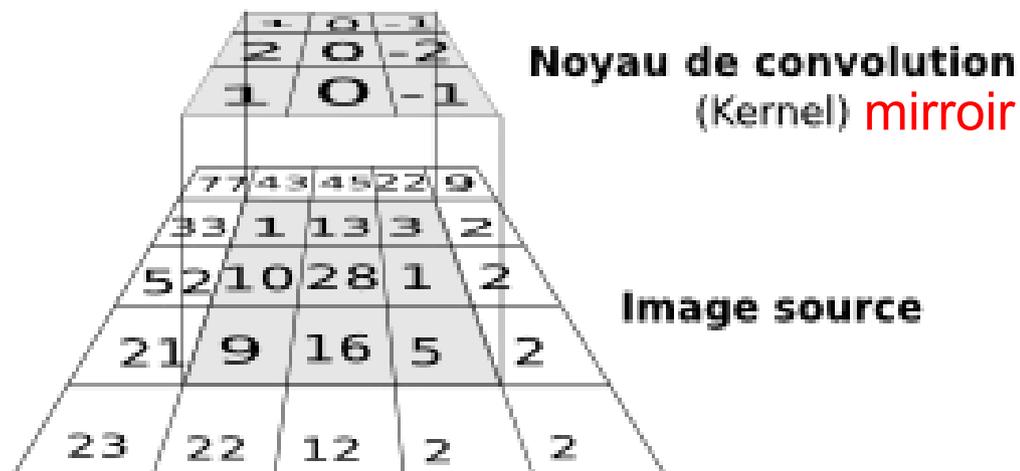


Noyau de convolution





$$= 1 \cdot 1 + 0 \cdot 13 + -1 \cdot 3 + 2 \cdot 10 + 0 \cdot 28 + -2 \cdot 1 + 1 \cdot 9 + 0 \cdot 16 + -1 \cdot 5$$



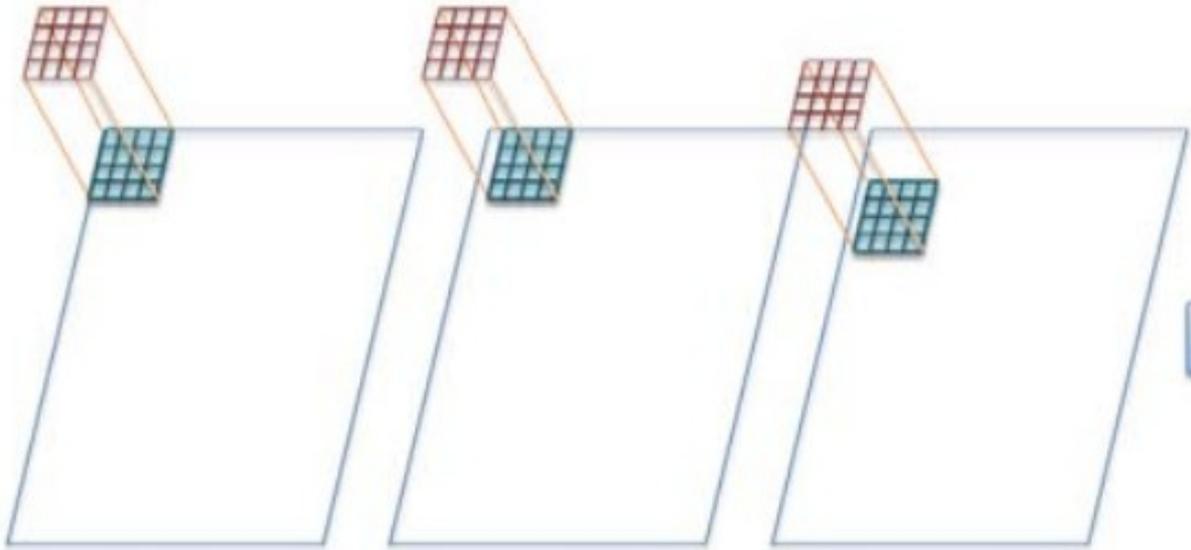


Image traitée

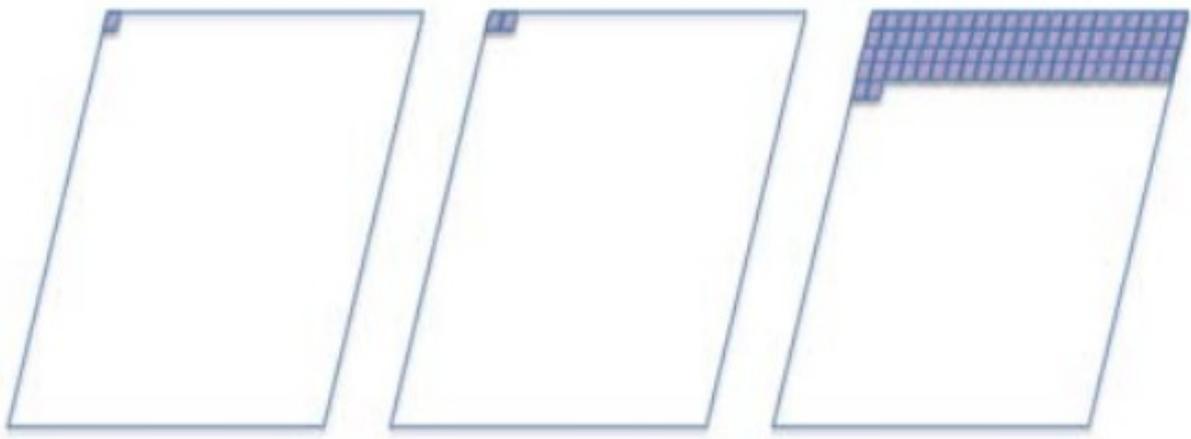
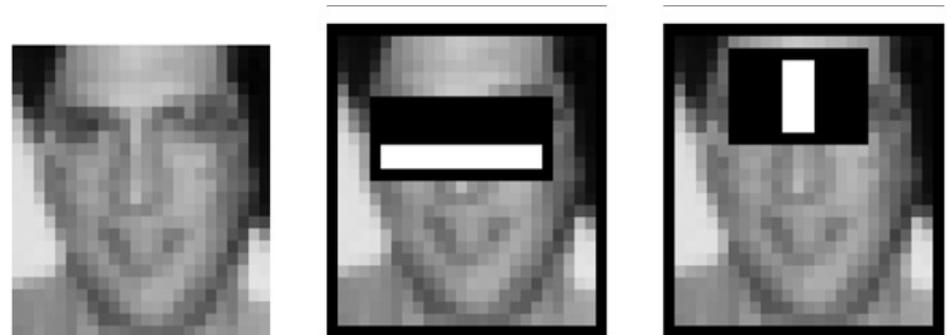
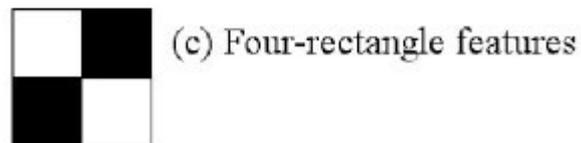
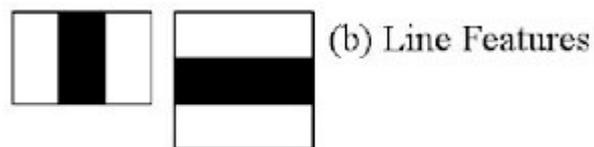
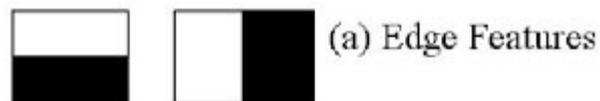


Image filtrée

# Caractéristiques pseudo-Haar

- Les caractéristiques pseudo-Haar sont « similaires » aux noyaux de convolution et sont utilisés pour détecter la présence de caractéristiques dans des images.
- Pourquoi « pseudo » : inspirées des ondelettes de Haar (mais pas identiques)
- **Attention**, le calcul n'est pas celui de la convolution :
  - ils sont utilisés comme dans la **corrélacion sans miroir** du noyau
  - Chaque caractéristique a comme résultat une seule valeur calculée en soustrayant la somme des pixels sous les pixels blancs (du masque) à la somme des pixels sous les pixels noirs (du masque)



0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

1 masque de Haar :

- Blanc : 0
- Noir : 1

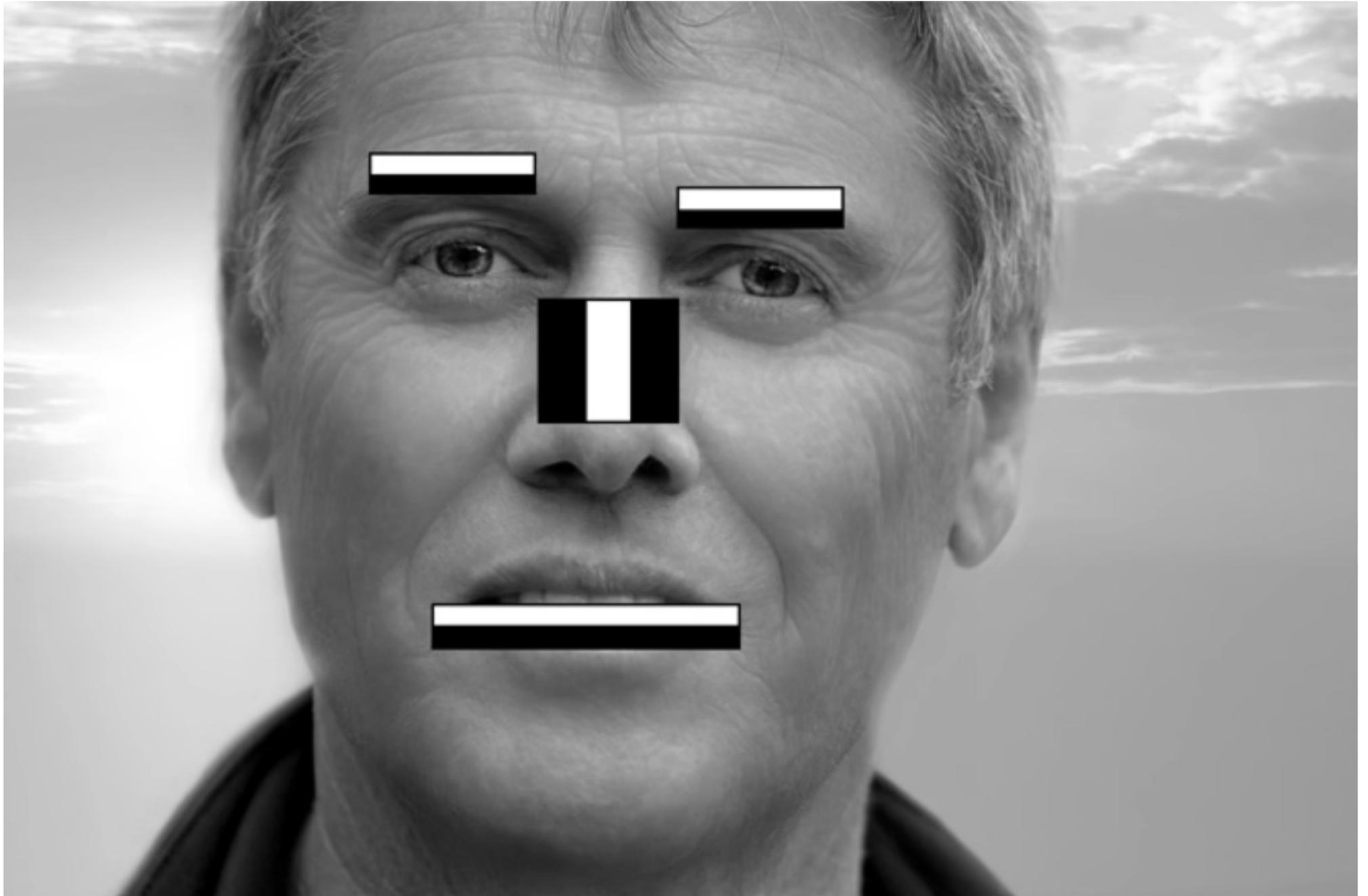
0.1	0.2	0.6	0.8
0.2	0.3	0.8	0.6
0.2	0.1	0.6	0.8
0.2	0.1	0.8	0.9

Valeurs des pixels

$$\text{Résultat} = \text{noirs} - \text{blancs} = \frac{1}{n} \sum_{x \in \text{noirs}} I(x) - \frac{1}{n} \sum_{x \in \text{blancs}} I(x)$$

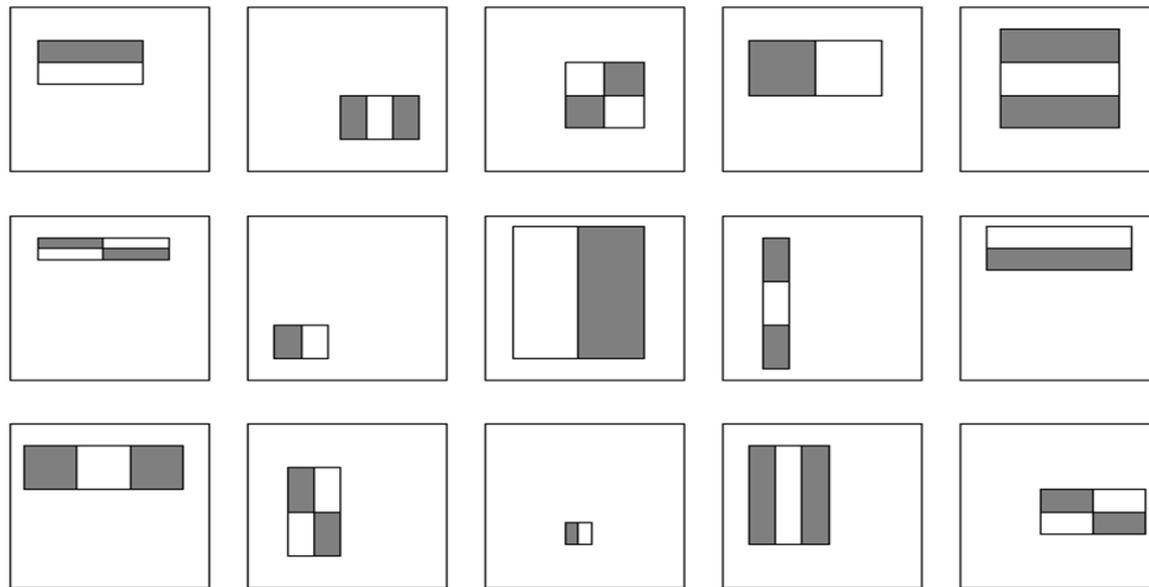
Dans le cas idéal, le résultat est 1 :

**Plus la valeur calculée est proche de 1 plus la détection est forte.**



Balazs Holczer

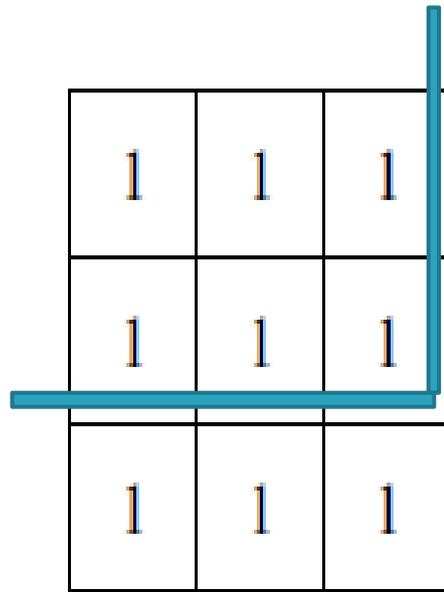
- L'algorithme de Viola & Jones utilise des fenêtres 24x24  
<https://www.youtube.com/watch?v=k3bJUP0ct08>
- En prenant en compte tous les paramètres comme la position, l'échelle et le type, il existe plus de **160,000 masques** !  
<https://stackoverflow.com/questions/1707620/viola-jones-face-detection-claims-180k-features>



- Il faut donc une méthode rapide de calcul : les **images intégrales**.

# Images intégrales

L'image intégrale au pixel  $(x,y)$  est la somme des pixels au-dessus et à gauche de  $(x,y)$  (lui compris)



1	1	1
1	1	1
1	1	1

Input image

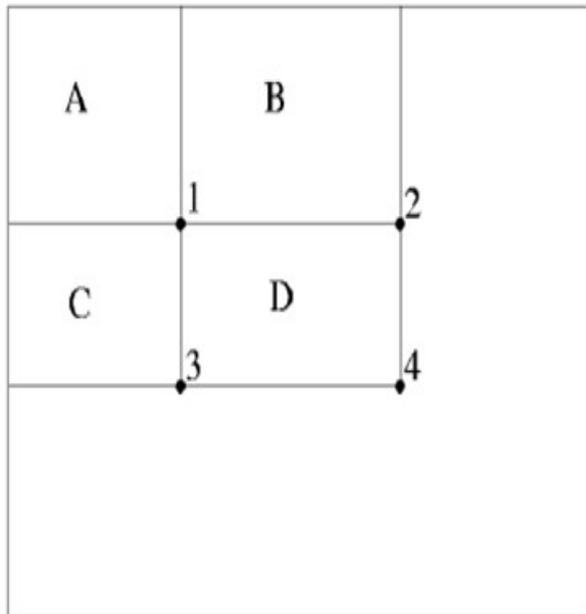
1	2	3
2	4	6
3	6	9

Integral image

# Images intégrales

Permet le calcul de la somme de tous les pixels à l'intérieur d'un rectangle en utilisant les valeurs des quatre coins de l'image intégrale.

Image intégrale



Somme de tous les pixels dans D

$$\begin{aligned}
 D &= 4 - 2 - 3 + 1 \\
 &= (A+B+C+D) - (A+B) - (A+C) + A \\
 &= D
 \end{aligned}$$

Original

5	2	3	4	1
1	5	4	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

$$5 + 2 + 3 + 1 + 5 + 4 = 20$$

Integral

5	7	10	14	15
6	13	20	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

Original

5	2	3	4	1
1	5	4	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

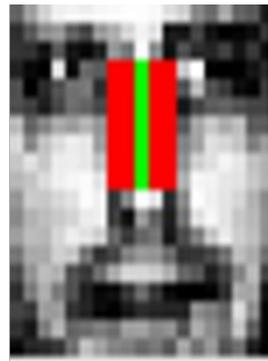
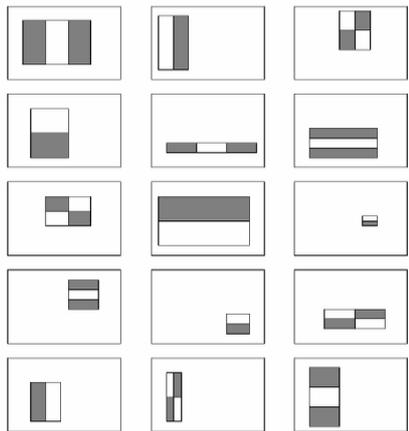
$$\begin{aligned}
 &5 + 4 + 2 + \\
 &2 + 1 + 3 = 17
 \end{aligned}$$

Integral

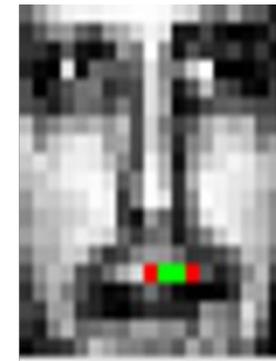
5	7	10	14	15
6	13	20	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

$$34 - 14 - 8 + 5 = 17$$

Seule une petite partie des 160000 caractéristiques sont utiles pour reconnaître un visage, il faut apprendre lesquels !



Relevant feature



Irrelevant feature

# Adaboost

- Adaboost est une méthode d'apprentissage automatique (machine learning algorithm) qui permet de trouver les « meilleures » caractéristiques parmi les 160,000 possibles.
- On parle de classifieur faible : une caractéristique (pertinente) et un seuil associé.
- Sont conservés tous les classsifeurs faibles qui font mieux que le hasard.
- Une combinaison linaire de tous les classifieurs faibles donne un classifieur plus efficace appelé classifieur fort.

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \dots$$

Classifieur fort

Classifieurs  
faibles

# Base de données

- Base de données
  - 5000 visages (de face)
  - $10^8$  non visages
  - Visages normalisés
    - échelle, translation
- Beaucoup de variations
  - Personnes différentes
  - Illumination
  - Pose (rotations)



**Faces**



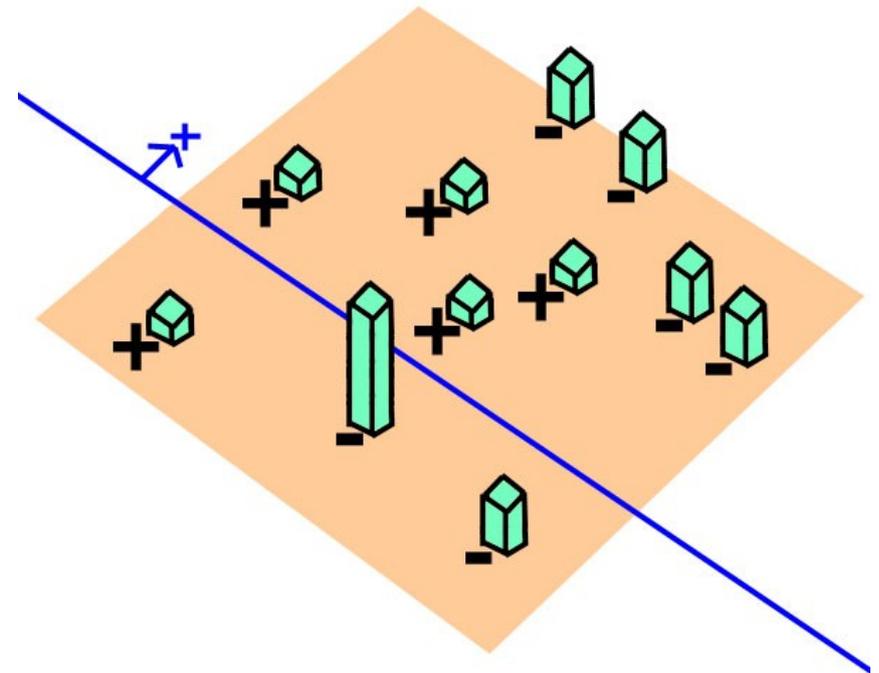
**Non-faces**

# Adaboost

$$\epsilon_j = \sum_{i=1}^n w_i |h_j(x_i) - y_i|$$
$$\epsilon_t = \min_j(\epsilon_j)$$

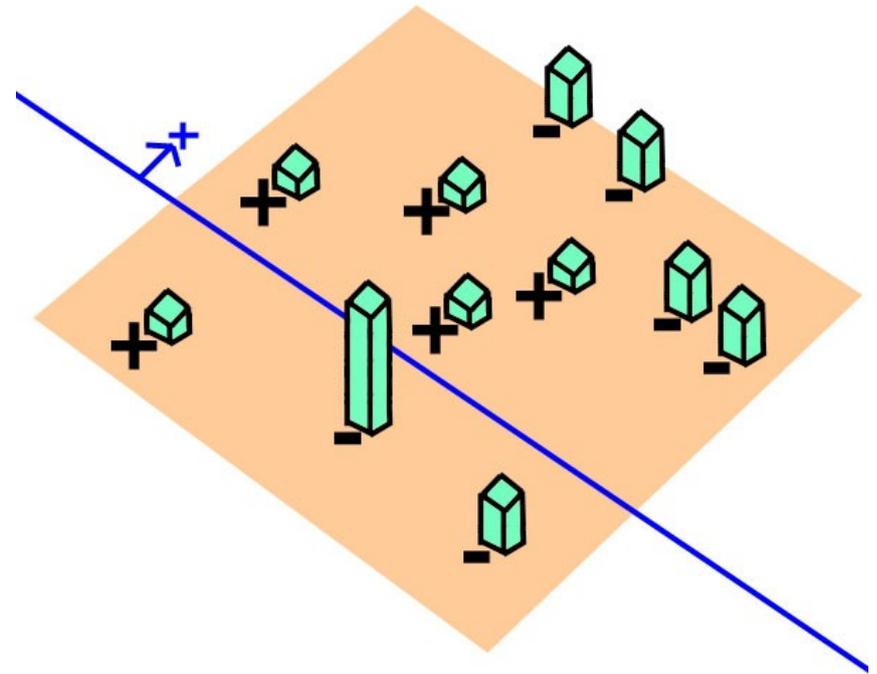
- AdaBoost débute avec des poids uniformes sur les exemples d'apprentissage (visages/non visages)
- Selection du classifieur faible avec la plus petite erreur pondéré :  $\epsilon_t$
- Le poids  $\alpha_t$  associé à ce classifieur est inversement proportionnel à  $\epsilon_t$  :  
**les classifieurs avec la plus faible erreur sont favorisés.**
- Augmentation des poids sur les exemples mal classés
- (Répétition)
- Finalement, combinaison linéaire des classifieurs faibles → **classifieur fort**

Idéalement  $y_i = h(x_i)$   
classe réelle = classe prédite



$$h_{\text{strong}}(\mathbf{x}) = \begin{cases} 1 & \alpha_1 h_1(\mathbf{x}) + \dots + \alpha_n h_n(\mathbf{x}) \geq \frac{1}{2}(\alpha_1 + \dots + \alpha_n) \\ 0 & \text{otherwise} \end{cases} \quad \alpha_t = \ln \left( \frac{1}{\beta_t} \right)$$

# Adaboost

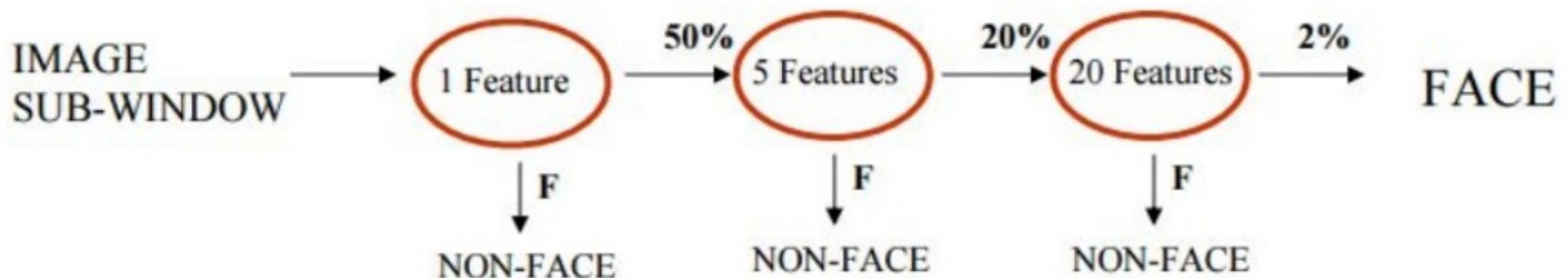


$$h_{\text{strong}}(\mathbf{x}) = \begin{cases} 1 & \alpha_1 h_1(\mathbf{x}) + \dots + \alpha_n h_n(\mathbf{x}) \geq \frac{1}{2}(\alpha_1 + \dots + \alpha_n) \\ 0 & \text{otherwise} \end{cases}$$

- Le principe de l'algorithme de Viola & Jones est de faire des parcours successifs avec des masques de tailles différentes sur des sous-fenêtres.
- Même s'il y a plusieurs visages sur l'image, il est évident que la grande majorité des sous-fenêtres ne contient pas de visages (non-faces).
- L'idée est d'éliminer rapidement les sous-fenêtres qui ne contiennent pas de visages et de se concentrer sur les régions qui contiennent probablement des visages.
- Un seul classifieur fort n'est pas suffisant car trop coûteux en temps de calcul → **cascade de classifieurs.**

# Cascade de classifieurs

- Cette cascade est constituée d'une suite de classifieurs forts dont le nombre caractéristiques (classifieurs faibles) va en **augmentant**.
- Chaque classifieur fort détermine de façon de plus en plus précise si la sous-fenêtre contient, ou non, un visage.
- Les sous-fenêtres qui ne contiennent manifestement pas de visages sont rapidement écartées.



# Cascade de classifieurs

## Apprentissage sur une base de données (visages)

- Pour définir une cascade de filtres on doit définir :
  - Nombre de classifieurs forts successifs.
  - Nombre de caractéristiques dans chaque classifieur fort.
  - Seuil sur chaque classifieur fort.
- Problème d'optimisation:
  - Recherche de la combinaison optimale ?

# Cascade de classifieurs

Trouver cette combinaison est extrêmement difficile et Viola & Jones ont proposés l'heuristique suivante :

## Heuristique :

- Choix de  $f_i$  = Taux maximal acceptable de Faux Positifs pour le  $i^{\text{ème}}$  classifieur fort
- Choix de  $d_i$  = Taux minimal acceptable de Vrais Positifs pour le  $i^{\text{ème}}$  classifieur fort
- Choix de  $F_{\text{target}}$  = Taux de Faux Positifs Global à l'issue de la cascade.

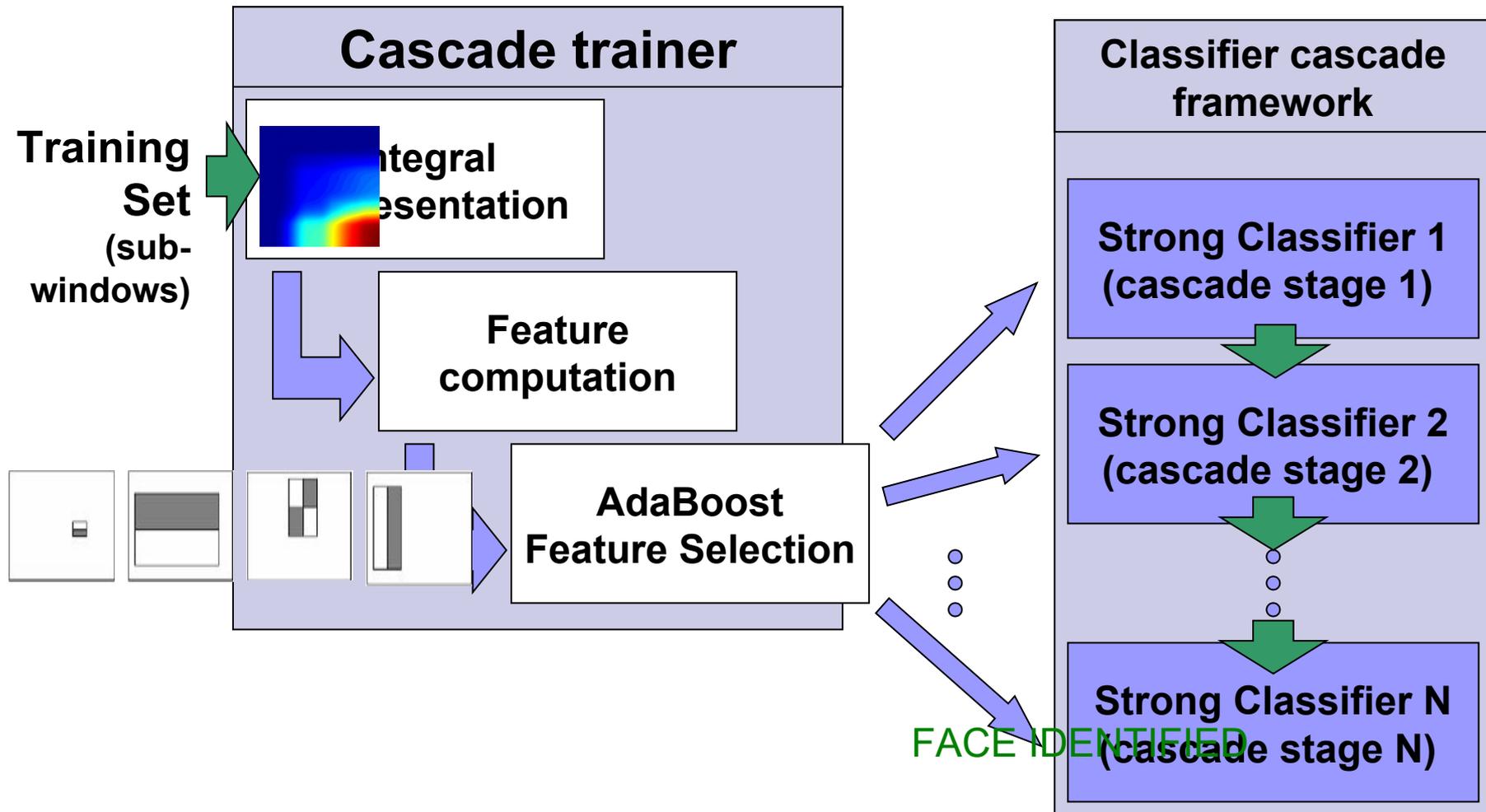
Tant que  $F_{\text{target}}$  n'est pas atteint :

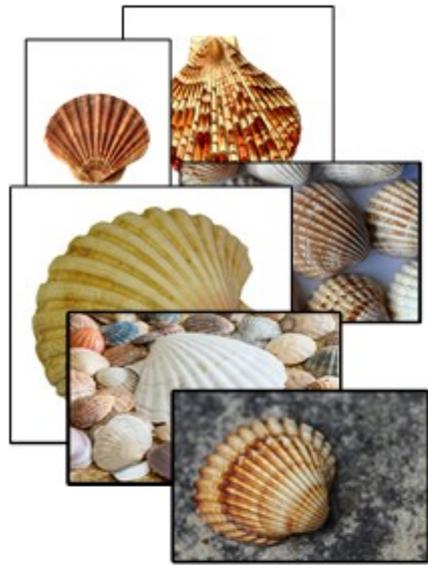
- Ajouter un nouveau classifieur fort  $i$  :
  - Tant que les Taux  $f_i, d_i$  ne sont pas atteints :
  - Ajout de caractéristiques (classifieur faible) dans ce classifieur par AdaBoost.

		Prédiction (classifieur)	
		Négatifs	Positifs
Réalité	Négatifs	VN (Vrais Négatifs)	FP (Faux Positifs)
	Positifs	FN (Faux Négatifs)	VP (Vrais Positifs)

		Prédiction (classifieur)		$\Sigma$
		non visages	visages	
Réalité	non visages	89.5 %	10.5 %	100.0 %
	visages	37.5 %	62.5 %	100.0 %

# Testing phase





Sample Images



OpenCV Cascade  
Training Tool

Trained Cascade  
XML



<https://kongakura.fr/article/haar-cascade>



USB Camera



OpenCV libraries



OpenCV Object  
Detection  
Module



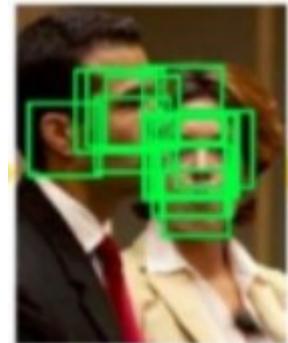
Identified Objects

# Avantages

- Calcul rapide des caractéristiques
- Sélection efficace des caractéristiques
- Détection invariante en translation (fenêtres glissantes) et en échelle :
  - A la place d'une pyramide on fait varier la taille des caractéristiques.
- Peut être adapté à d'autres types d'objets que les visages.

# Inconvénients

- Efficace uniquement en vue de face :
  - Perte de performance au-delà d'une rotation de 45°
- Sensible aux changements d'illumination
- Il peut y avoir plusieurs détections du même visage à cause des fenêtres glissantes (mais il existe des solutions simples).



# Viola Jones face detection algorithm

Haar features | Integral Image | Adaboost | Cascading

## Résumé

