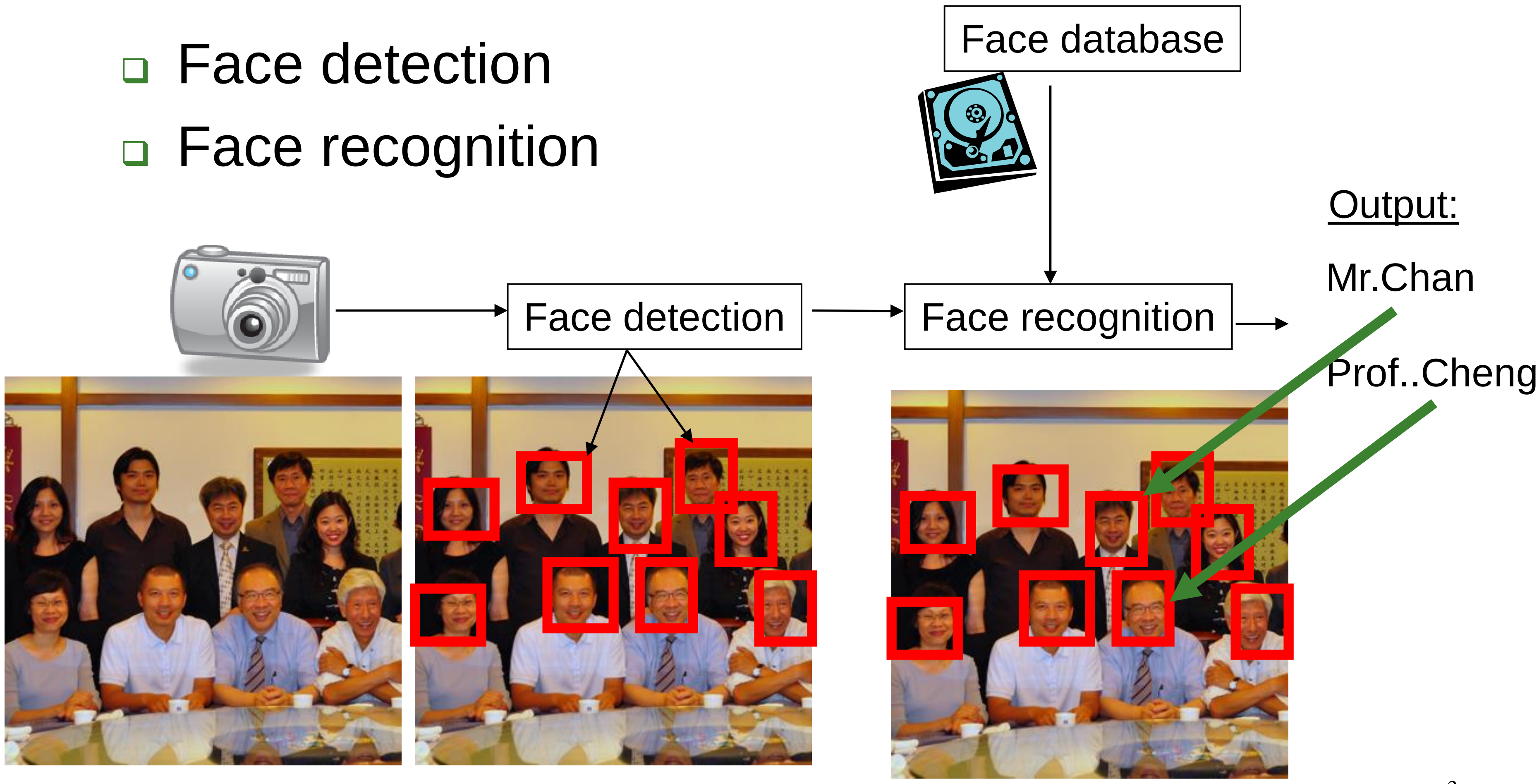
Dr. George Karraz, Ph. D.

Ch. 6: Face detection

Dr. George Karraz, Ph. D.

Introduction

- Face interface
  - Face detection
  - Face recognition

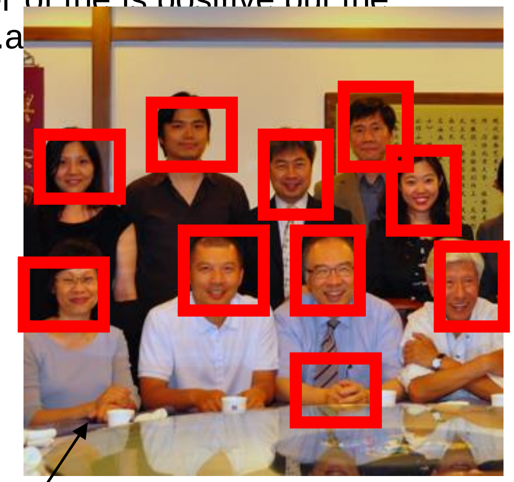


Face detection [1]

- To detect faces in an image (Not recognize it yet)
- Challenges
  - A picture has 0,1 or many faces.
  - Faces are not the same: with spectacles, mustache etc.
  - Sizes of faces vary a lot.
- Available in most digital cameras nowadays
- The simple method
  - Slide a window across the window and detect faces.
    - Too slow, pictures have too many pixels. (1280x1024=1.3M pixels)

Evaluation of face detection

- Detection rate
  - Total number of faces that are correctly detected/total number of faces actually exist in the picture
  - Should be high > 95%.
- False positive rate
  - The detector output is positive but it is false (there is actually no face). Definition of False positive: A result that is erroneously positive when a situation is normal. An example of a false positive: a particular test designed to detect cancer of the is positive but the person does not have cancer. (http://www.medterms.com/script/main/art.a
  - Should be low $<10^{-6}$
- A good system has
  - High detection rate,
  - Low false positive rate.



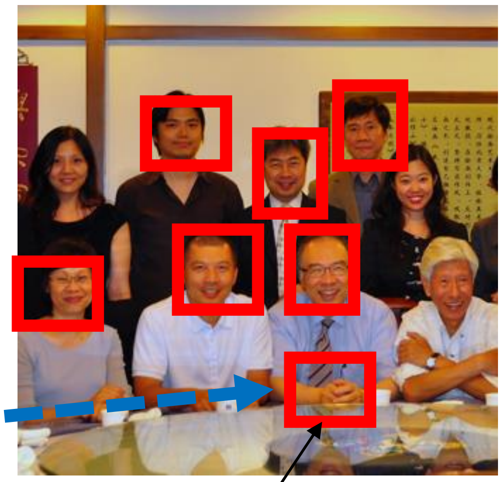
False positive result

Example

- What are the detection rate and false positive rate here?
  - Answer
    - detection rate=(6/9)*100%
    - false positive rate=(1/7)*100%

6 faces correctly detected in the picture, 9 actually faces exit in the image

7 windows reported to have faces , but in 1 window it is not a face



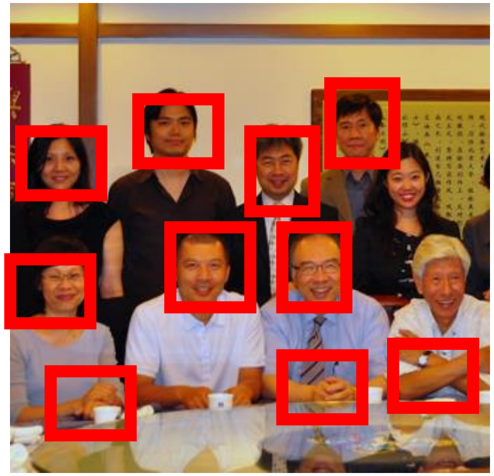
False positive result

The Viola and Jones method [1]

- The most famous method
- Training may need weeks
- Recognition is very fast, e.g. real-time for digital cameras.
- Techniques
  1. Integral image for feature extraction
  2. Ada-Boost for face detection
  3. Attentional cascade for fast rejection of non-face sub-windows

Class exercise 6.1

- Detected results are in red frames
- What are the detection rate and false positive rate here?



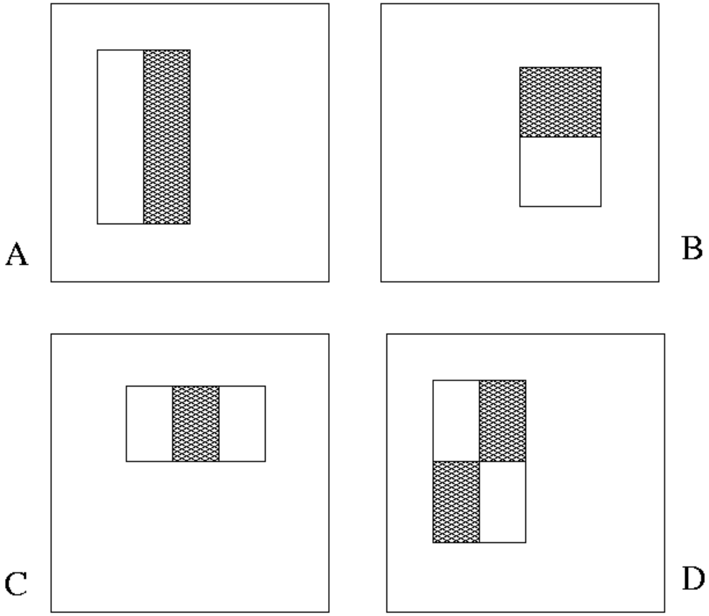
  - Answer
    - detection rate=?
    - false positive rate=?

The Viola and Jones method
Technique 1:

Integral image for feature extraction

Image Features ref[3]

A very simple feature calculation method “Rectangle filters”



Rectangle_Feature_value f=

∑ (pixels values in white area) –
∑ (pixels values in shaded area)

Example

- Find the *Rectangle_Feature_value* (f) of the box enclosed by the dotted line
- *Rectangle_Feature_value f=*
- *∑ (pixels values in white area) – ∑ (pixels values in shaded area)*
- *f=(8+7)-(0+1)*
- *=15-1= 14*

1	2	3	3
3	0	1	3
5	8	7	1
0	2	3	6

Class exercise 6.2

- Find the *Rectangle_Feature_value* (f) of the box enclosed by the dotted line
- *Rectangle_Feature_value f=*
- *∑ (pixels values in white area) –*
  ∑ (pixels values in shaded area)
- *f=*

2	7	5	8
2	6	2	9
5	4	8	2
8	2	5	10

Example: A simple face detection method using one feature

- ❑*Rectangle_Feature_value f*
- ❑*f*= ∑*(pixels in white area) – ∑ (pixels in shaded area)*

- ❑If (f) is large, then it is face ,i.e.
- ❑if (f)>threshold, then
- ❑ face
- ❑Else
- ❑ non-face

This is not a face.
Because f is small

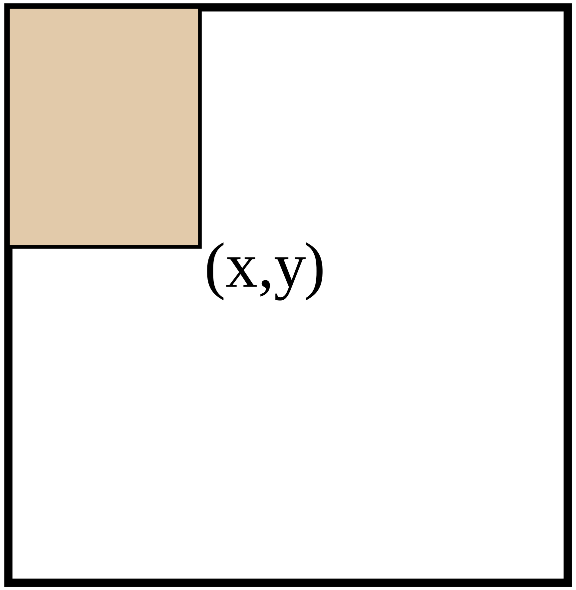
Result

This is a face: The eye-area (shaded area)is dark, the nose-area(white area) is bright.
So f is large, hence it is face

How to find features faster?

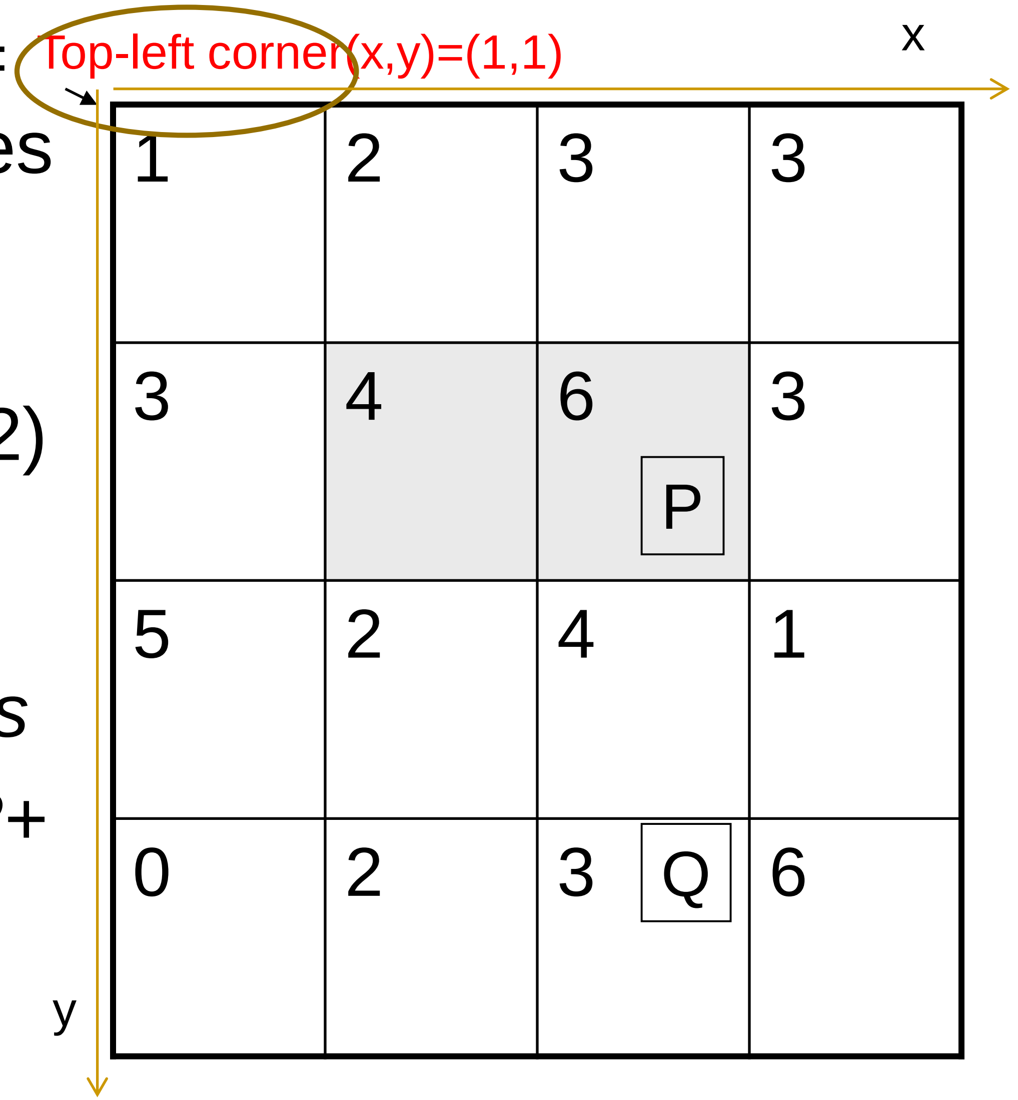
Integral images fast calculation method [Lazebnik09]

- The *integral image = sum of all* pixel values above and to the left of (*x*,*y*)
- Can be found very quickly

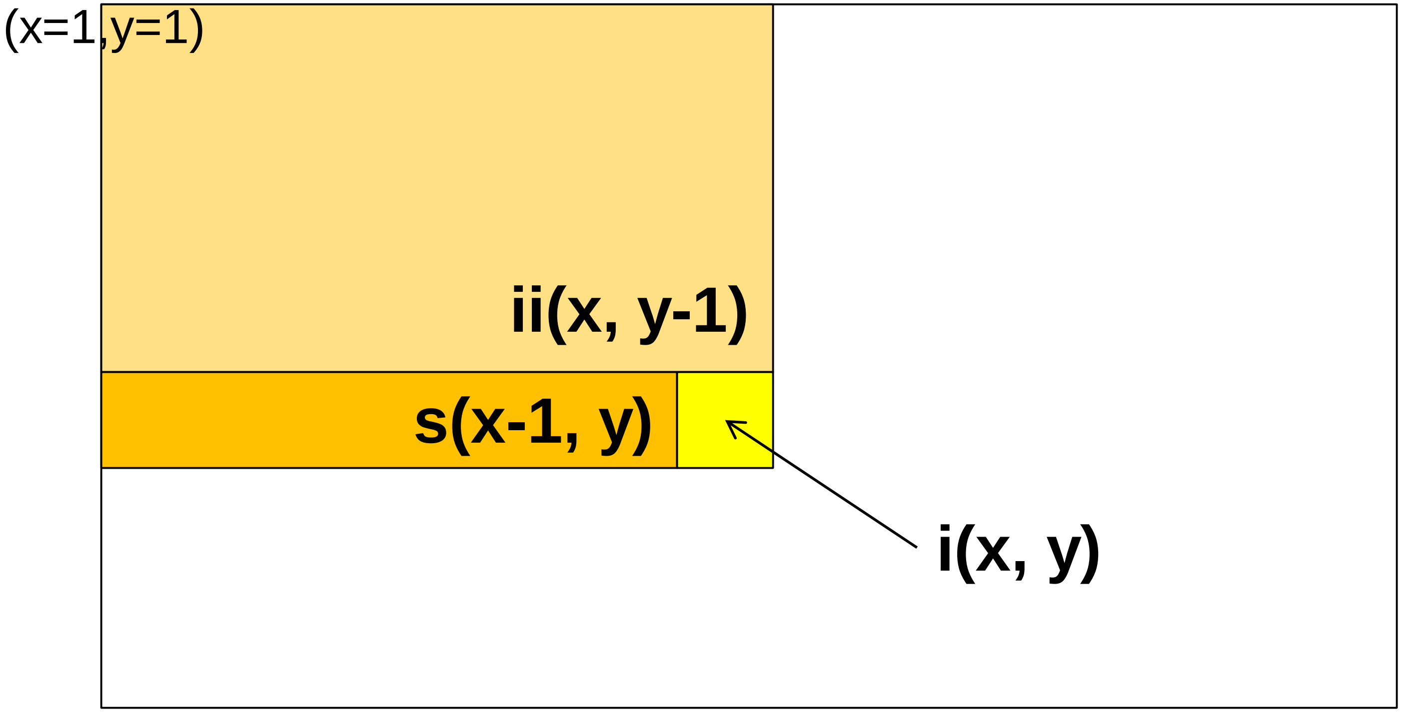


Examples

- The *integral image* = *sum of all* pixel values above and to the left of (x,y)
- Pixel P is at (x=3,y=2)
  - *integral image of P is =1+2+3+3+4+6*
- *integral image of Q is*
- *=1+2+3+3+4+6+5+2+4+0+2+3*



Computing the integral image [Lazebnik09]



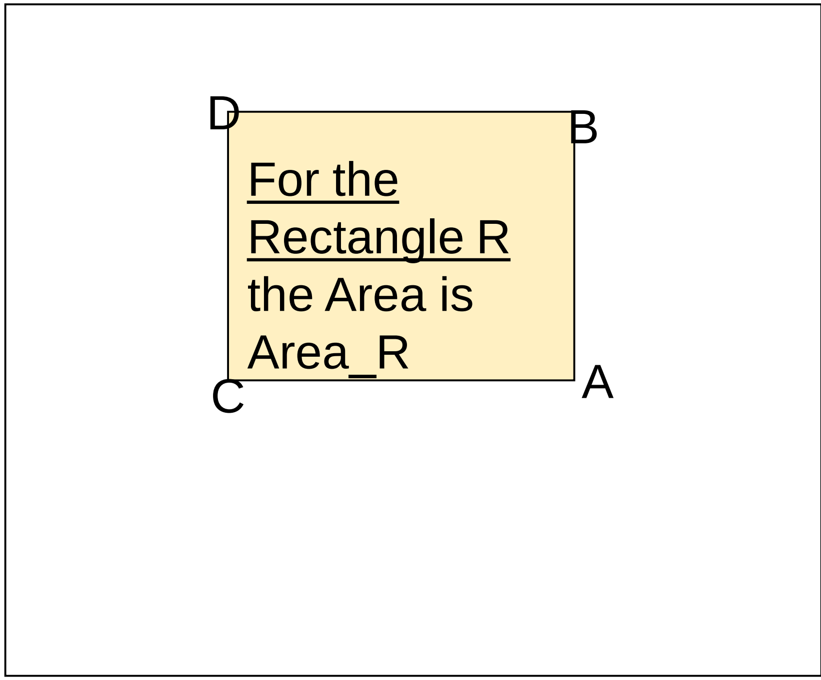
- Cumulative row sum: s(x, y) = s(x–1, y) + i(x, y)
- Integral image: ii(x, y) = ii(x, y−1) + s(x, y)
- MATLAB: ii = cumsum(cumsum(double(i)), 2);

Calculate sum within a rectangle

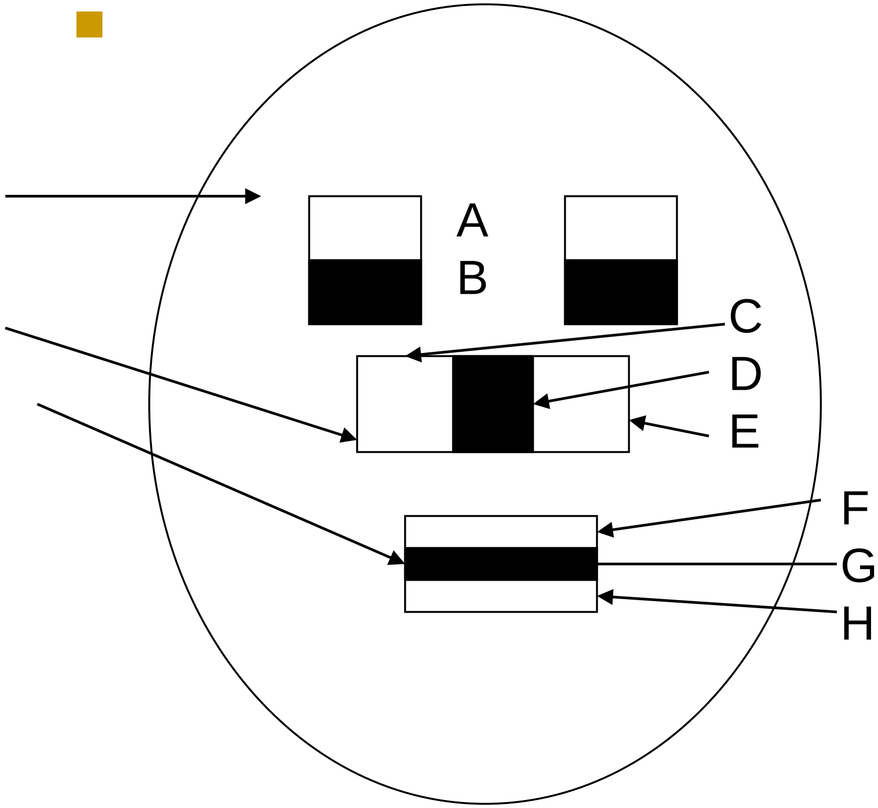
- A,B,C,D are the values of the integral images at the corners of the rectangle R.
- The sum of image values inside R is:

  Area_R = A – B – C + D

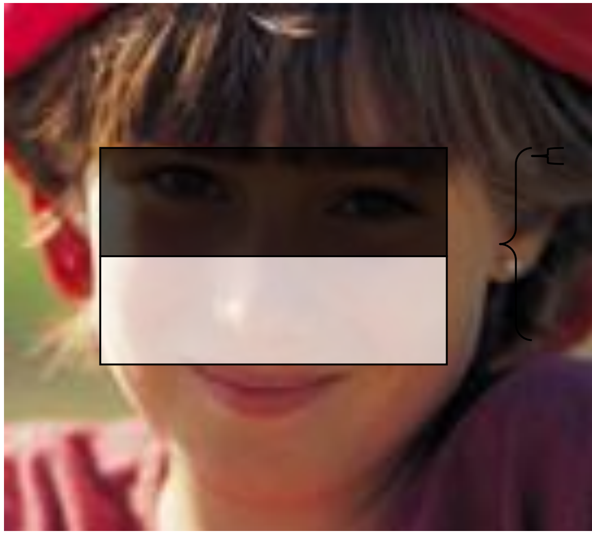
- If A,B,C,D are found , only 3 additions are needed to find Area_R
- Calculations of areas can reused for other windows.



Why do we need to find pixel sum of rectangles? Answer: We want to get face features

- You may consider these features as face features
  - Left Eye: (Area_A-Area_B)
  - Nose :(Area_C+Area_E-Area_D)
  - Mouth:(Area_F+Area_H-Area_G)
- They can be different sizes, polarity and aspect ratios
- 

Face feature and example



Pixel values inside the areas

Shaded area	10	20	4
	7	45	7
White area	216	102	78
	129	210	111

F=Feat_val =
pixel sum in white area - pixel sum in shaded area

Example

•Pixel sum in white area=
216+102+78+129+210+111=846

•Pixel sum in shared area=
10+20+4+7+45+7=93

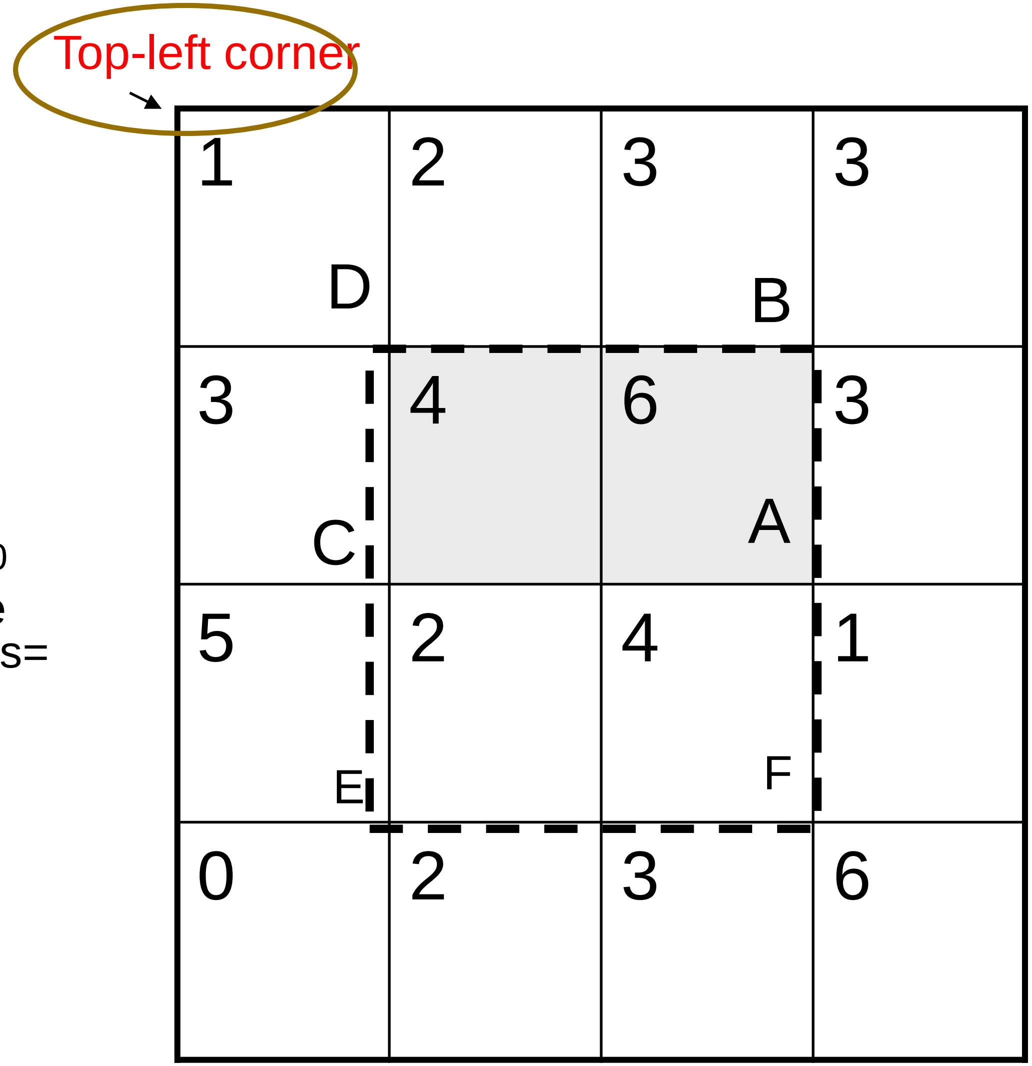
Feat_val=F=846-93=753

If F>threshold,
    feature= +1
Else
    feature= -1   End if;

If we can choose threshold =700 , so feature is +1.

A face

Definition: Area_X = sum of pixels in the rectangular area from the left-top corner to pixel X (including the top left corner and pixel X).

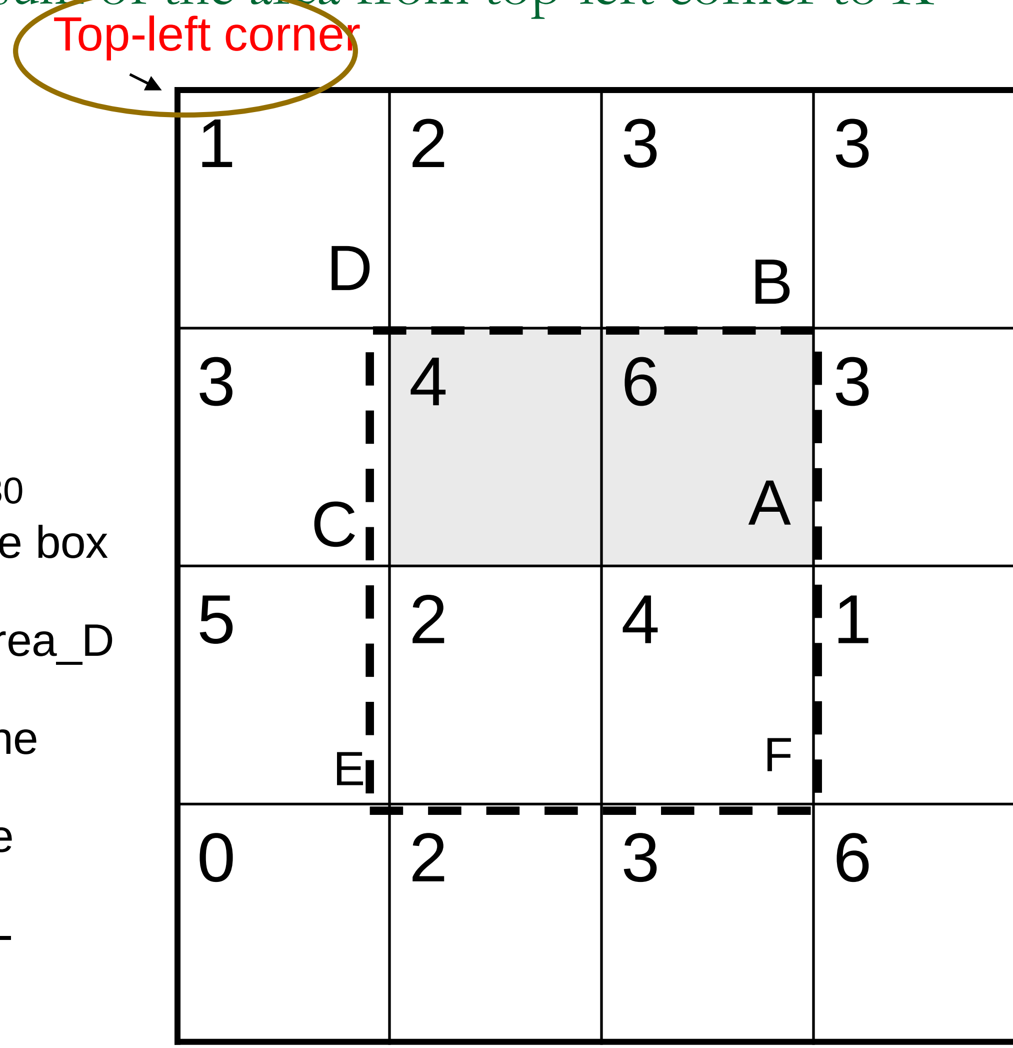


- Find the feature output of this image.
- Area_D=1
- Area_B=1+2+3=6
- Area_C =1+3=4
- Area_A=1+2+3+3+4+6=19
- Area_E=? 1+3+5=9
- Area_F=? 1+2+3+3+4+6+5+2+4=30
- Pixel sum of the area inside the box enclosed by the dotted lines=
- Area_F - Area_B - Area_E +Area_D =? 30-6-9+1=16

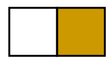
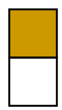
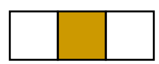
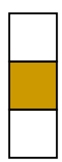
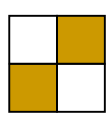
Class exercise 6.3

Definition: Area at X =pixel sum of the area from top-left corner to X= Area_X

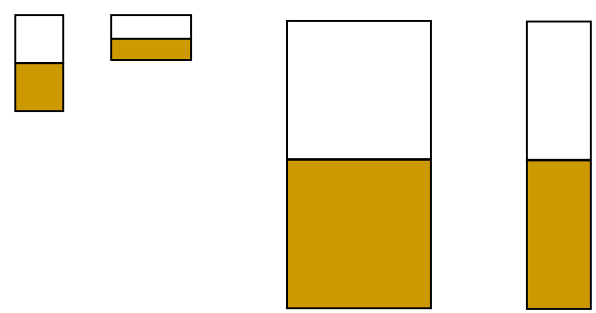
- Find the feature output of this image.
- Area_D=1
- Area_B=1+2+3=6
- Area_C =1+3=4
- Area_A=1+2+3+3+4+6=19
- Area_E=? 1+3+5=9
- Area_F=? 1+2+3+3+4+6+5+2+4=30
- Pixel sum of the area inside the box enclosed by the dotted lines=
- Area_F - Area_B - Area_E +Area_D =30-6-9+1=16
- WA=White area enclosed by the dotted line=?
- GA=Gray area enclosed by the dotted line=?
- (white area-shaded area)=WA-WG=?



4 basic types of <u>Rectangular Features</u> for (white_area)-(gray_area)

- <u>Type) Rows x columns</u>
- Type 1) 1x2 
- Type 2) 2x1 
- Type 3) 1x3 
- Type 4) 3x1 
- Type 5) 2x2 

- Each basic type can have difference sizes and aspect ratios.
- I.e. the following feature windows are of the same type (Type2) even they have different sizes, or aspect ratios
- Each rectangle inside is of the same dimension



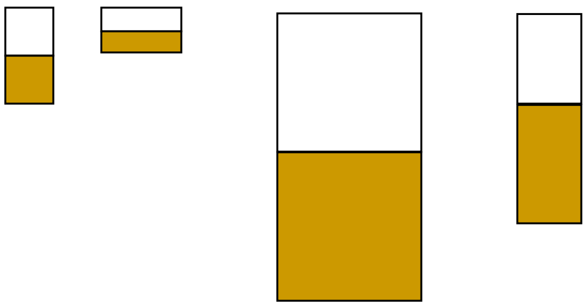
Faces can be any sizes,

Example: a face can be big or small , from to 24 x24 to 1024x1024,

- There are faces with different sizes

So, we need feature windows with different sizes.

- As long as white/gray areas have the relations
- The followings are Type2 <u>Rectangular Features</u>
  - <u>The white rectangle is above the shaded rectangle</u>
  - <u>White and shaded rectangle are of same dimension</u>



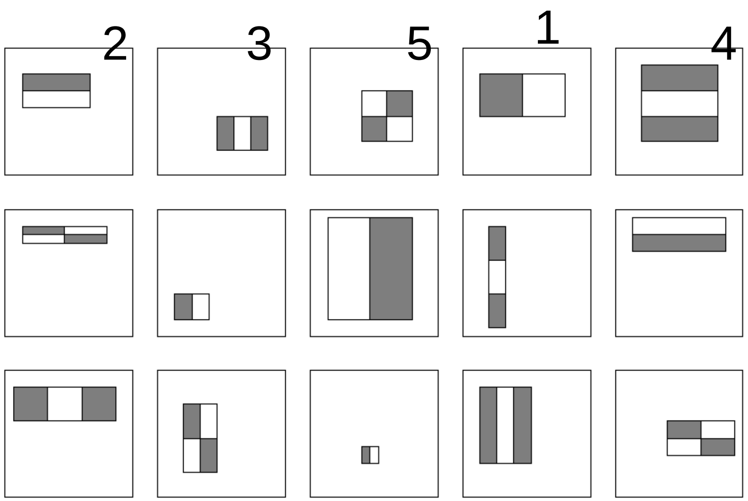
Class exercise 6.4

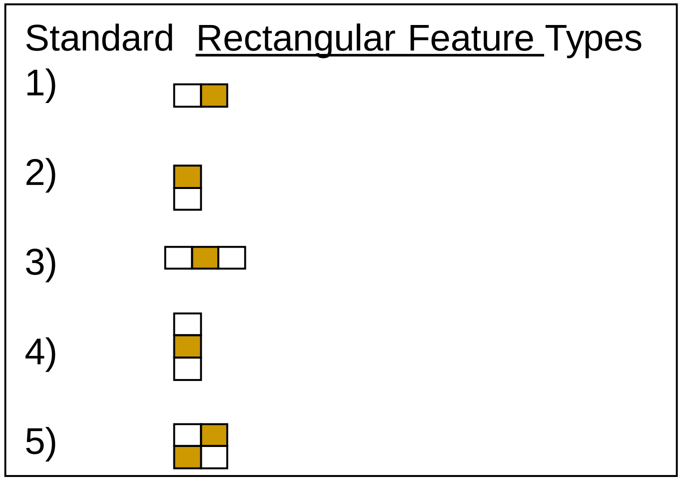
Feature selection [Lazebnik09]

- For a 24x24 detection region, the number of possible rectangle features is ~160,000!
- Name the types (type 1,2,3,4,5) of the rectangular features in the figures.

Some examples and their types

Fill in the types for the 2nd, 3rd rows

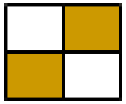




Class exercise 5: Features in a 24x24 (pixel) window

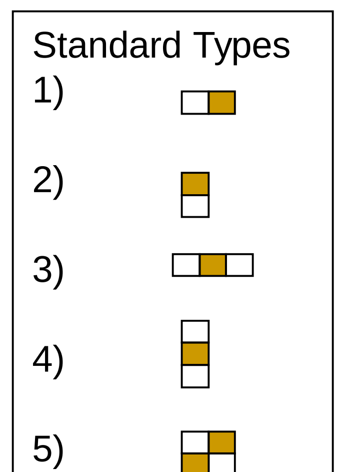
- Exercise 5a : How many rectangular features of all 5 types can be found a 24x24 pixel window?
- Answer: 162,336 (explain)
- Exercise 5b : How many type 1 features in a 24x24 (pixel) window?
- Answer:_43200 (explain)

Type 5)



Class exercise 6.6?

- Still keeping the 5 basic rectangular features types (1,2,3,4,5) (5 types: 2x1,1x2,3x1,1x3,2x2)
  - Find the number of rectangular features for a resolution of 36 x36 windows
  - Answer: 816264, explain your answer.



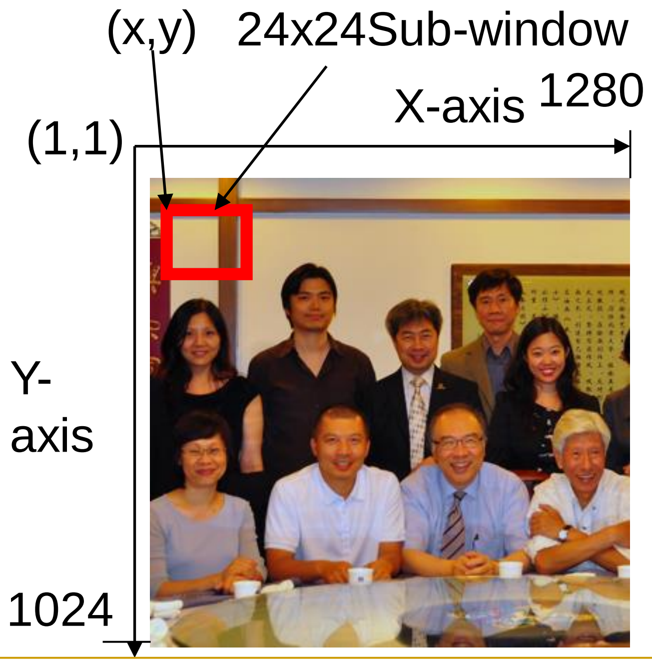
The Viola and Jones method
Technique 2:

AdaBoost for face detection

Class exercise 7: The detection challenge

- Use 24x24 base window
- For y=1;y<=1024;y++

  {For x=1;x<=1024;x++{
  - Set (x,y) = the left top corner of the 24x24 sub-window, different scales are needed to be considered too.
  - For the 24x24 sub-window, extract 162,336 features and see they combine to form a face or not.}
- }
- Exercise 7 : Discuss the number of operations required.
- Conclusion : too slow, solution use boosting



Answer 7:

Solution to make it efficient

- The whole 162,336 feature set is too large
  - Solution: select good features to make it more efficient.
  - Use: “Boosting”
- Boosting
  - Combine many small weak classifiers to become a strong classifier.
  - Training is needed.

Boosting for face detection

- Define weak learners based on rectangle features

value of rectangle feature

$$h_t(x) = \begin{cases} 1 & \text{if } p_t f_t(x) < p_t \theta_t \\ 0 & \text{otherwise} \end{cases}$$

window

threshold

P_t= polarity{+1,-1}

Face detection using Adaboost

- AdaBoost training
  - E.g. Collect 5000 faces, and 9400 non-faces. Different scales.
  - Use AdaBoost for training to build a strong classifier.
  - Pick suitable features of different scales and positions, pick the best few. (Take months to do , details is in [Viola 2004] paper)
- Testing
  - Scan through the image (any where), pick a window (any size ≥ 24x24) and rescale it to 24x24,
  - Pass it to the strong classifier for detection.
  - Report face, if the output is positive

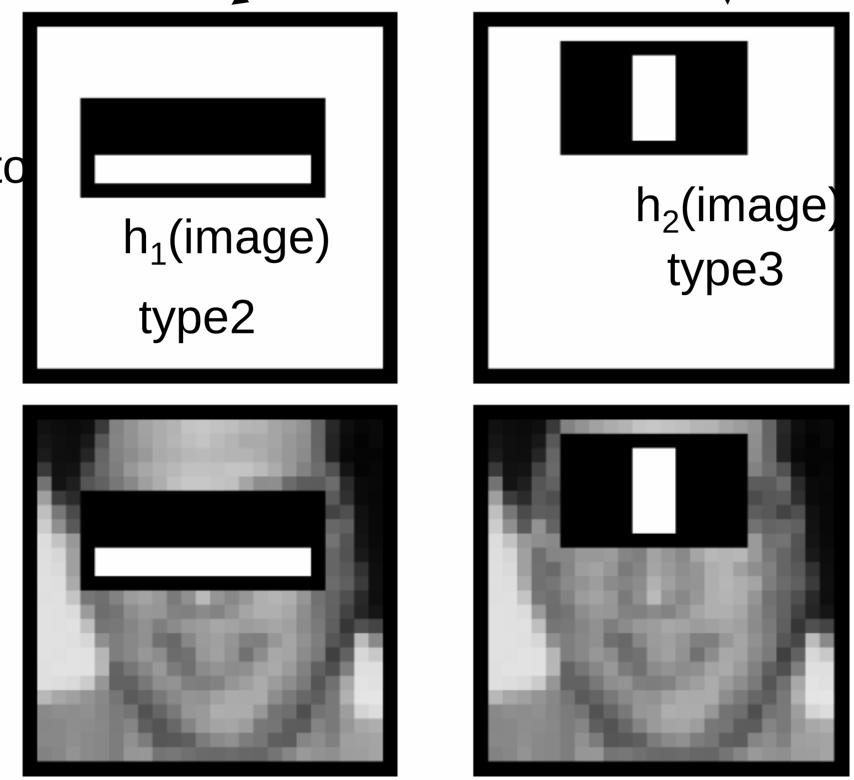
Boosting for face detection [viola2004]

- In the paper it shows that the following two features (obtained after training) in cascaded picked by AdaBoost have 100% detection rate and 50% false positive rate
- But 50% false positive rate is not good enough
- Approach [viola2004] :Attentional cascade

I.e. Strong classifier
H(face)=
Sign{$\alpha_1 h_1$(image)
+$\alpha_2 h_2$(image)}

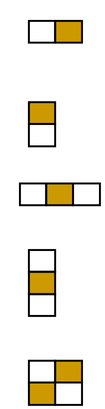
H(face)=+1→ face
H(face)=-1→non-face

Pick a window in the image and rescale it to 24x24 as "image"



Standard Types

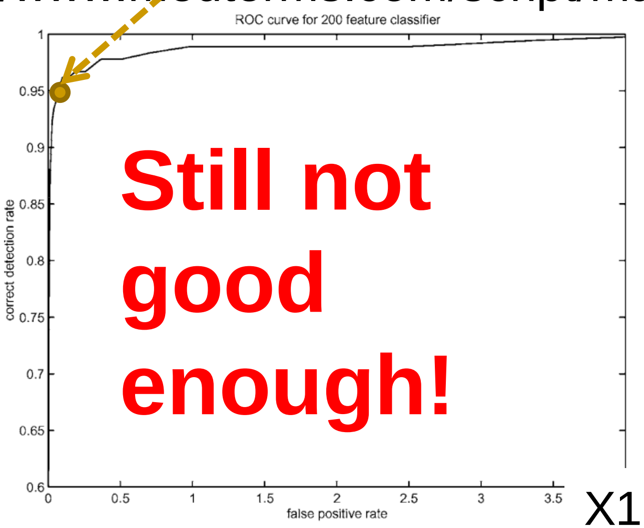
1)
2)
3)
4)
5)



Boosting for face detection

- An experiment shows: A 200-feature classifier can yield 95% detection rate and a false positive rate of 1 in 14084 (7.1×10^{-5} Still not good enough)
- Recall: False positive rate
  - The detector output is positive but it is false (there is actually no face). <u>*Definition of False positive*</u>*: A result that is erroneously positive when a situation is normal. An example of a false positive: a particular test designed to detect cancer of the toenail is positive but the person does not have toenail cancer.* (http://www.medterms.com/script/main/art.asp?articlekey=3377)

Correct Detection rate



$X10^{-3}$

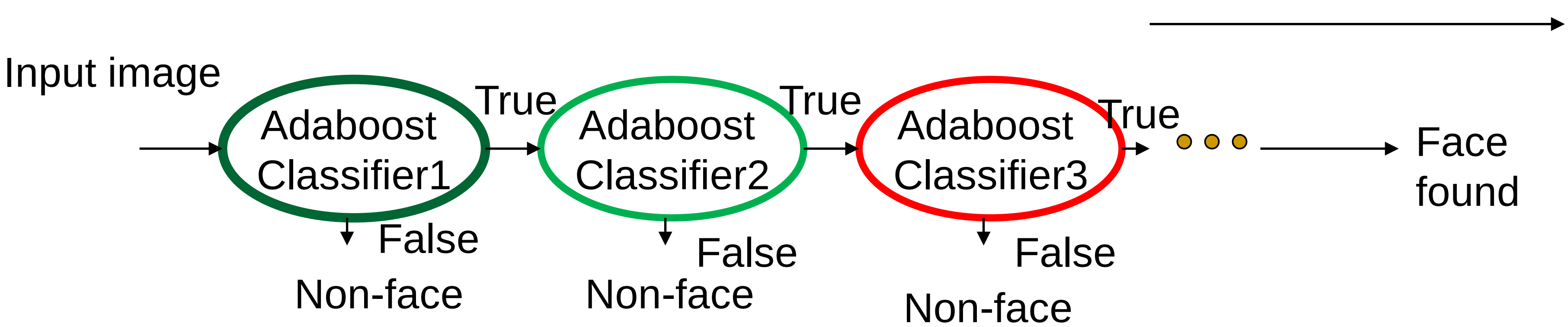
False positive rate

The Viola and Jones method
Technique 3:

Attentional cascade for fast rejection of non-face sub-windows

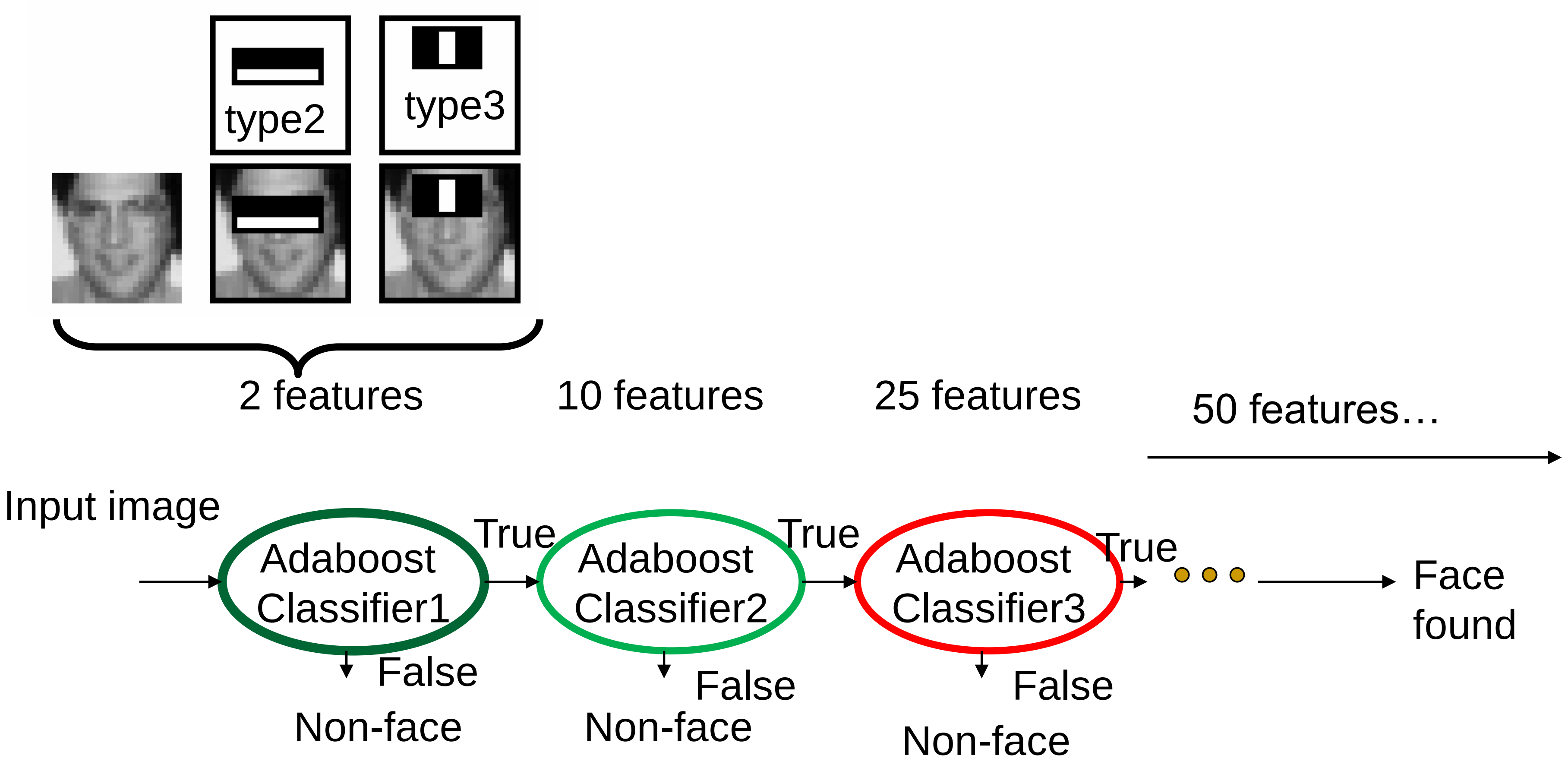
To improve false positive rate: Attentional cascade

- Cascade of many AdaBoost strong classifiers.
- Begin with simple classifiers to reject many negative sub-windows.
- Many non-faces are rejected at the first few stages.
- Hence the system is efficient enough for real time processing.



An example

- More features for later stages in the cascade [viola2004]

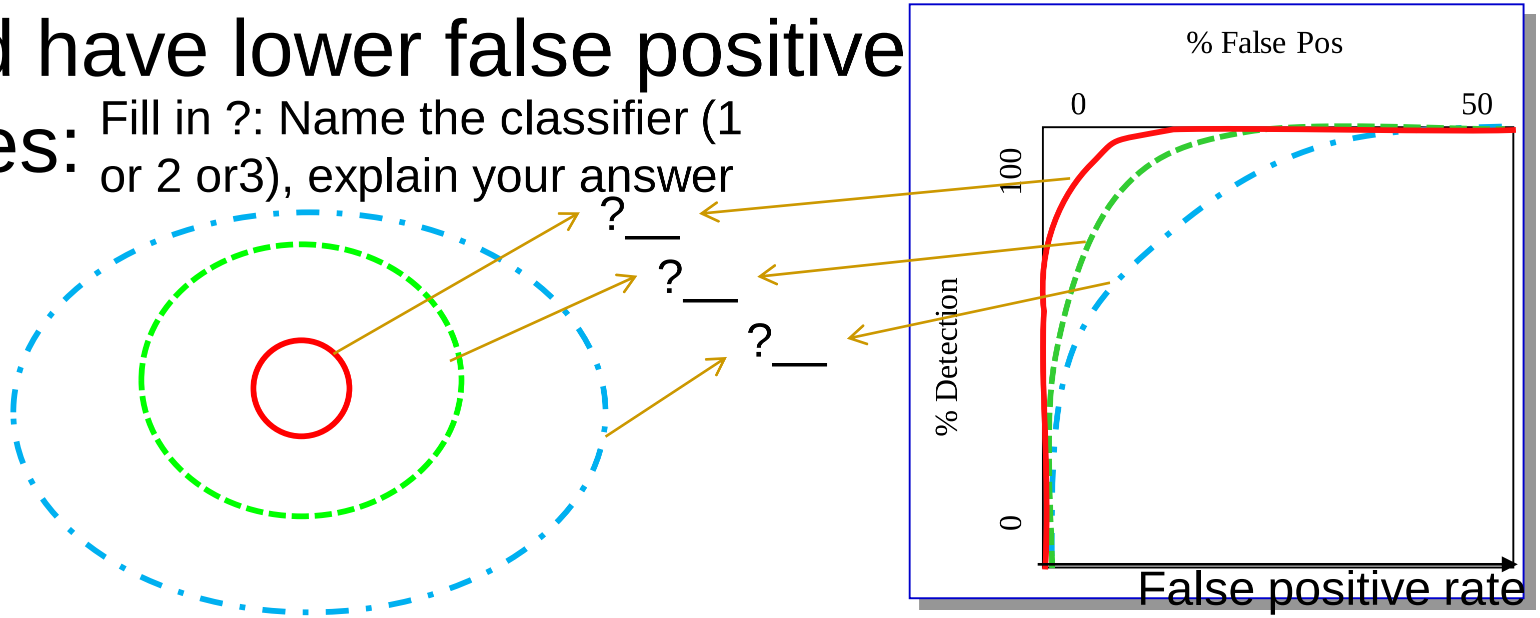


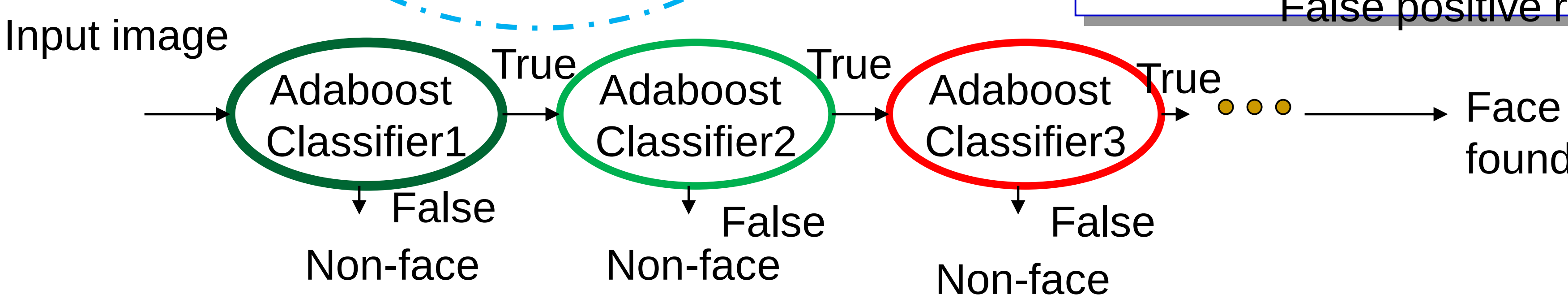
Class exercise 6.8: Attentional cascade

- Chain classifiers that are progressively more complex and have lower false positive rates:

Fill in ?: Name the classifier (1 or 2 or3), explain your answer

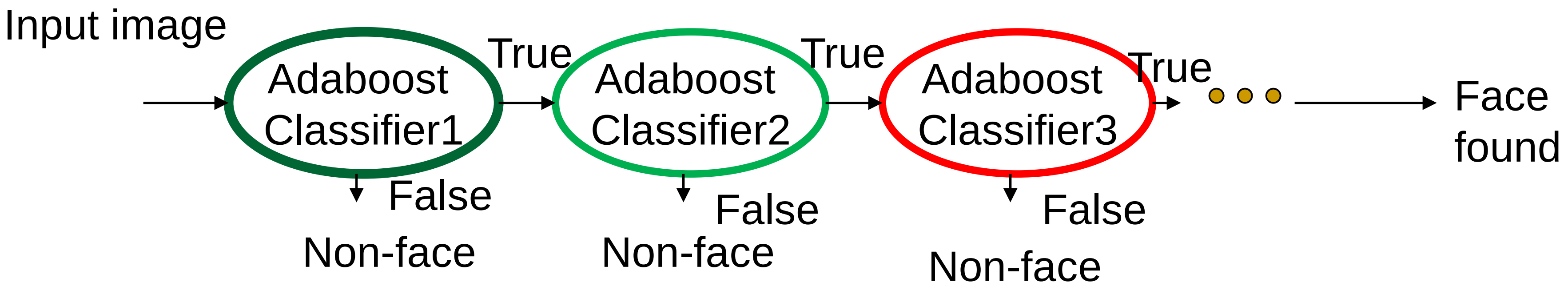
Receiver operating characteristic





Attentional cascade [Viola2004]

- Detection rate for each stage is 0.99 , for 10 stages,
  - overall detection rate is $0.99^{10} \approx 0.9$
- False positive rate at each stage is 0.3, for 10 stages
  - false positive rate $=0.3^{10} \approx 6\times10^{-6}$)



Detection process in practice [smyth2007]

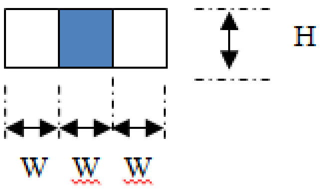
- Use 24x24 sub-window
- Scaling
  - scale the detection (not the input image)
  - Features evaluated at scales by factors of 1.25 at each level
  - Location : move detector around the image (1 pixel increments)
- Final detections
  - A real face may result in multiple nearby detections (merge them to become the final result)

Summary

- Learned
  - How to extract fece feature
  - How to apply adaboost for face detection
  - How to train up the system and how to detect face

Additional Exercise A1

- Definition: = Area_X = sum of pixels in the area from left-top corner to pixel X
- Based on the window in image1, answer the following questions.
- i) Find Area_A, Area_B, Area_C, Area_D, Area_E, Area_F
- In image1, calculate the number of Type-3 features found in each of the following different cases :
- W=1 pixel, H=1 pixel,
- W=2 pixels, H=2 pixels.

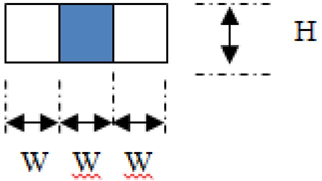
Type	Rows × Columns	Feature value	Features
Type -3	1 × 3	(Sum of pixles in shaded area) - (Sum of pixles in white area)	 Three rectangular blocks in a row. Width of each rectangle =W pixels. Height of each rectangle =H pixels.

2 A	7	**3**	**8**	**2**	**4 B**
0 C	4	2	3	5	5 D
1 E	3	6	8	2	8 F
8	0	3	5	3	2
1	3	5	7	4	1

Image 1

Answer A1

- Definition: = Area_X = sum of pixels in the area from left-top corner to pixel X
- Based on the window in Figure 1, answer the following questions.
- i) Find Area_A, Area_B, Area_C, Area_D, Area_E, Area_F
- Answer:
- Area_A=2
- Area_B=2+7+3+8+2+4=26
- Area_C=2
- Area_D=AreaB+4+2+3+5+5=26+4+2+3+5+5=45
- Area_E=3
- Area_F=Area_D+1+3+6+8+2+8=45+73



2 **A**	**7**	**3**	**8**	**2**	**4** **B**
0 **C**	4	2	3	5	5 D
1 **E**	3	6	8	2	8 F
8	0	3	5	3	2
1	3	5	7	4	1

- ii) Find the area inside the box CDFE based on the result in (i).
- Answer: Area_F-Area_B=73-26=47
- iii)Calculate the type 3 feature value in the area CDFE.
- Answer:(2+3+6+8)-(0+4+1+3)-(5+5+2+8)= -9
- iv)Calculate the number of features found in each of the following cases if W and H are the features are :
- W=1 pixel,H=1 pixels, answer=5x4=20
- W=2 pixels,H=2 pixels, Answer:4x1= 4

References

1. [viola2004] Paul A. Viola, Michael J. Jones: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2): 137-154 (2004) (PDF: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.137.4879&rep=rep1&type=pdf)
2. [viola2001] Paul A. Viola, Michael J. Jones, Rapid object detection using a boosted cascade of simple features CVPR 2001 (PDF: http://research.microsoft.com/en-us/um/people/viola/Pubs/Detect/violaJones_CVPR2001.pdf)
3. [Lazebnik09] www.cs.unc.edu/~lazebnik/spring09/lec23_face_detection.ppt
4. [stackoverflow] http://stackoverflow.com/questions/1707620/viola-jones-face-detection-claims-180k-features
5. [Jensen 2008] Ole Helvig Jensen," Implementing the Viola-Jones Face Detection Algorithm "Kongens Lyngby 2008 “, IMM-M.Sc.-2008-93,Technical University of Denmark Informatics and Mathematical Modeling
6. [smyth2007] Face detection using the viola Jones method ppt, UL Irvine (lecture notes of CS175 Fall 2007)
7. [yu tm]http://aimm02.cse.ttu.edu.tw/class_2009_1/PR/Lecture%207/Adaboost.ppt
8. [stackoverflow] http://stackoverflow.com/questions/1707620/viola-jones-face-detection-claims-180k-features

Appendix1

Advanced topics

Training

The face Adaboost detection system

Adaboost face detection Training algorithm [Jensen 2008]

Given : $(x_1,y_1),..(x_n,y_n)$, where $x_i \in X, Y=\{-1,+1\}$ for negative and positive examples

- Initialize weights :

$- w_{1,i} = 1/2M, M$ = number of positive example

$- w_{1,i} = 1/2L, L$ = number of positive example

For $t = 1,2,...,T$

{ Step1 : Normalize weights $w_{t,i} \leftarrow \dfrac{w_{t,j}}{\sum_{j=1}^{j=n} w_{t,j}}$

Step2 : Select the weak classfier with smallest weighted error

$$\text{find } \varepsilon_t = \min_{f,p,\theta} \sum_i^n w_i \left| h(x_i, f, p, \theta) - y_i \right|$$

Step3 : Define $h_t(x) = h(x, f_t, p_t, \theta_t)$, where f_t, p_t, θ_t are minimizer of ε_t at stage t

Step4 : update weigths

update the weights : $w_{t+1,i} = w_{t,i}\beta^{1-e_i}$

where $e_i = 0$ if example x_i is classfied correctly and $e_i = 1$ otherwise ,

and $\beta = \dfrac{\varepsilon_t}{1-\varepsilon_t}$

}

The final strong classifier is : $C(x) = \begin{cases} 1 \text{ if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2}\sum_{t=1}^{T}\alpha_t \\ 0 \quad \text{otherwise} \end{cases}$, where $\alpha_t = \log\dfrac{1}{\beta_t}$

Inside the main loop for training
For t=1,…T

- Step1
  - Init all weights
  - *Same weights all for samples at t=1*

Inside the main loop for training

For t=1,…T

-assume at stage *t*

- Step2: select the best weak classifier (weak learner) $\text{find } \varepsilon_t = \min_{f,p,\theta} \sum_i^n w_i \left| h(x_i, f, p, \theta) - y_i \right|$
- For all f (1,2,3,… 162,336 feature set)
- For all p (p=+1 or -1)
  - For different θ, (θ as low as possible to produce good result)
    - {

$$\varepsilon_{f,p,\theta} = \sum_i^n w_i \overbrace{\left| h(x_i, f, p, \theta) - y_i \right|}^{\text{Mistakenly classified}}$$

    - }

$$\{f, p, \theta\}_{best\_weak\_classifier} = \text{arguments}(\min\{\varepsilon_{f,p,\theta}\})$$

Step2 : more explanation

-assume at stage *t*

- Test every feature in the feature set {1,2,3,… 162,336 feature set}
- Test different polairty{+1,-1}: dark/white reversed.
- Try different θ (for simplicity start from 0.4), make it lower to see if performance (recognition, false-positive rates are improved.
- Output= {f_t (type of feature), p_t (polarity), θ_t (threshold)} which give the minimum error ε_t
- *{f_t , p_t , θ_t}= (minimizer of ε_t) at stage t*

Inside the main loop for training
For t=1,…T
-assume at stage *t*

- Step3

$$h_t(x) = h(x, f_t, p_t, \theta_t)$$

$$f_t, p_t, \theta_t \text{ are minimizer of } \varepsilon_t \text{ at stage } t$$

Inside the main loop for training

For t=1,…T

-assume at stage *t*

- step4

update the weights :

$$w_{t+1,i} = w_{t,i}\beta^{1-e_i}$$

where $e_i = 0$ if example x_i is classfied correctly and $e_i = 1$ ortherwsie,

and $\beta = \dfrac{\varepsilon_t}{1-\varepsilon_t}$

Inside the main loop for training

For t=1,…T

-assume at stage *t*

- step5

$$C(x) = \begin{cases} 1 \ \text{if} \ \sum_{\text{t}=1}^{\text{T}} \alpha_{\text{t}} h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_{\text{t}} \\ 0 \qquad \qquad \qquad \qquad \text{otherwise} \end{cases}$$

$$\text{where } \alpha_{\text{t}} = \log \frac{1}{\beta_t}$$

Appendix2

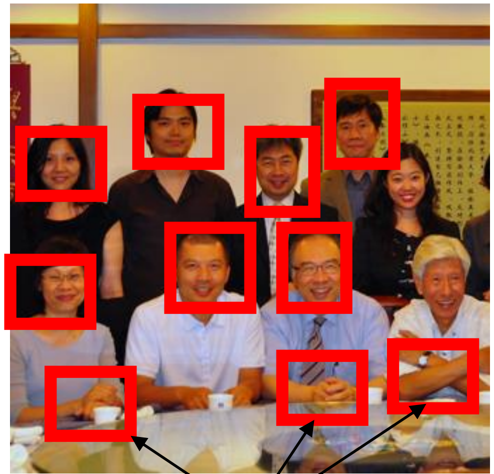
- Answers to exercises

Answer: Class exercise 6.1

- Detected results are in red frames
- What are the detection rate and false positive rate here?
  - Answer
    - detection rate=(7/9)*100%
    - false positive rate=(3/10)*100%

7 faces correctly detected in the picture, 9 actually faces exit in the image

10 windows reported to have faces , but in 3 windows they are not faces.



False positive results

Answer: Class exercise 6.2

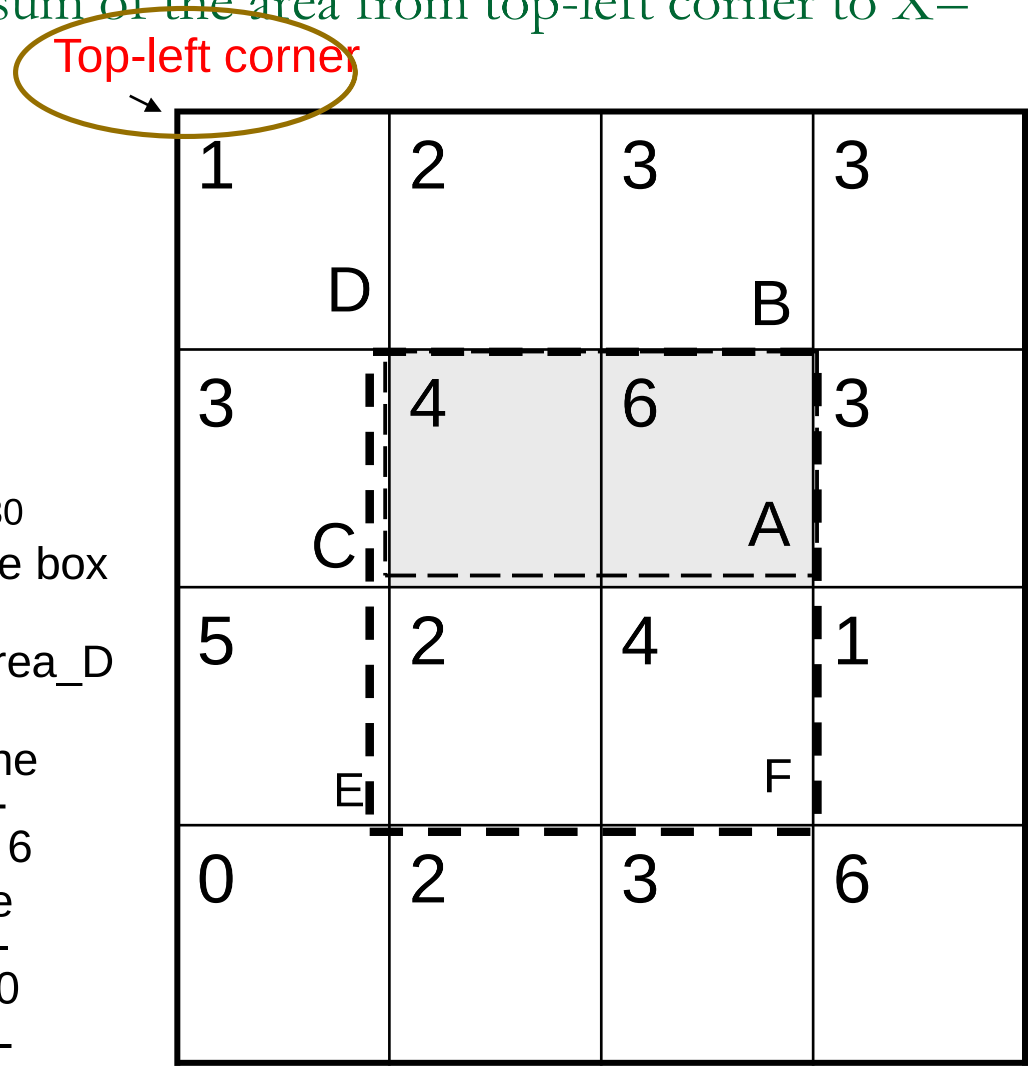
- Find the *Rectangle_Feature_value* (f) of the box enclosed by the dotted line
- *Rectangle_Feature_value f=*
- *∑ (pixels values in white area) – ∑ (pixels values in shaded area)*
- *f=(4+8)-(6+2)=12-8=4*

2	7	5	8
2	6	2	9
5	4	8	2
8	2	5	10

Answer: Class exercise 3

Definition: Area at X =pixel sum of the area from top-left corner to X= Area_X

- Find the feature output of this image.
- Area_D=1
- Area_B=1+2+3=6
- Area_C =1+3=4
- Area_A=1+2+3+3+4+6=19
- Area_E=? 1+3+5=9
- Area_F=? 1+2+3+3+4+6+5+2+4=30
- Pixel sum of the area inside the box enclosed by the dotted lines=
- Area_F - Area_B - Area_E +Area_D =? 30-6-9+1=16
- WA=White area enclosed by the dotted line= Area_F - Area_A - Area_E +Area_C=30-19-9+4= 6
- GA=Gray area enclosed by the dotted line= Area_A - Area_B - Area_C +Area_D=19-6-4+1=10
- (white area-shaded area)=WA-WG=6-10=-4



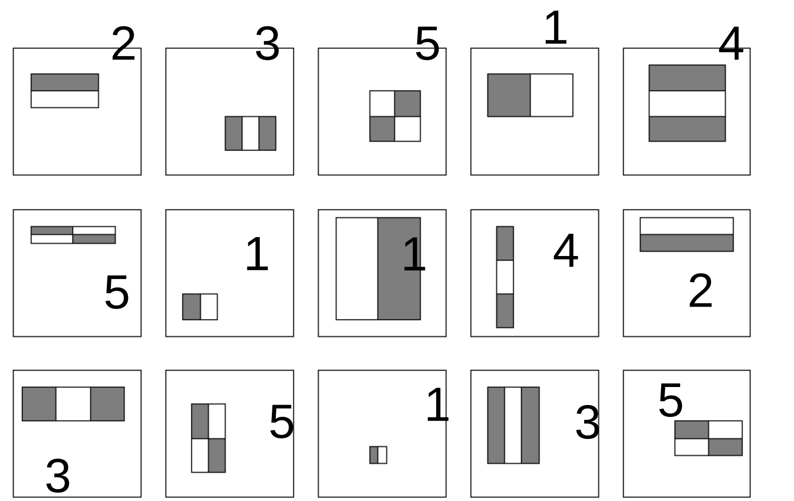
Answer: class exercise 4

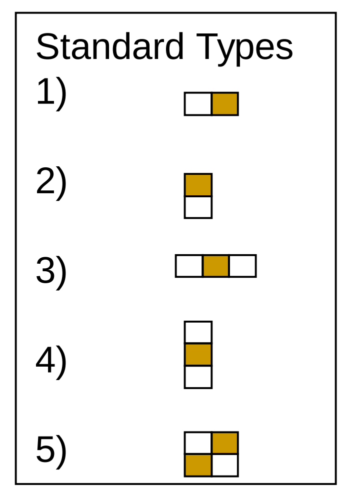
Feature selection [Lazebnik09]

- For a 24x24 detection region, the number of possible rectangle features is ~160,000!
- Name the types (type 1,2,3,4,5) of the <u>rectangular features</u> in the figures .
- Answer: see the labels in the diagram.

Some examples and their types

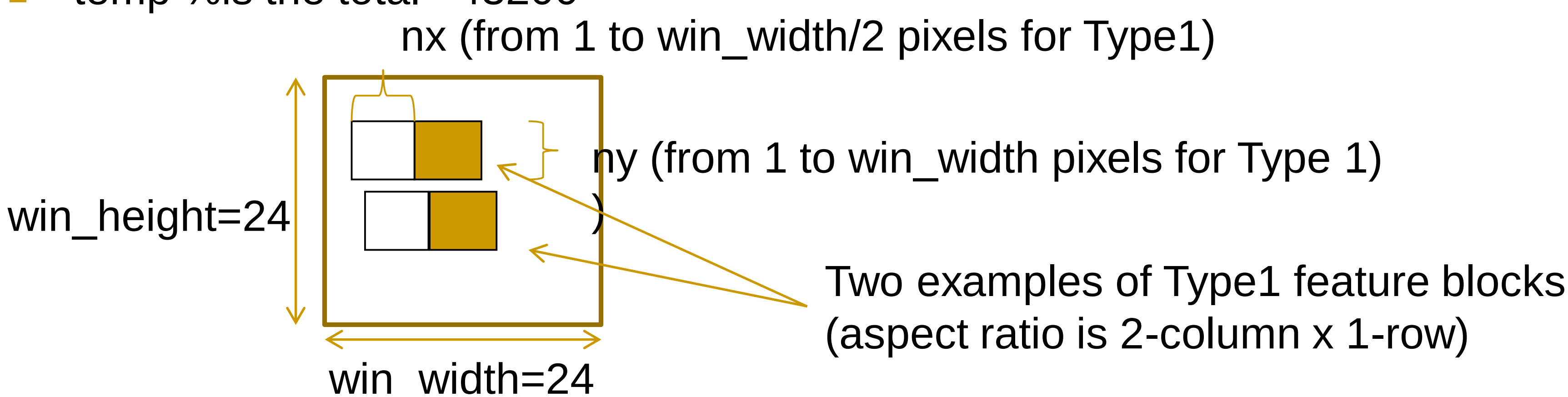
Fill in the types for the 2nd, 3rd rows





Answer5a1: Class exercise 5a : How many type 1 features in a 24x24 (pixel) window?

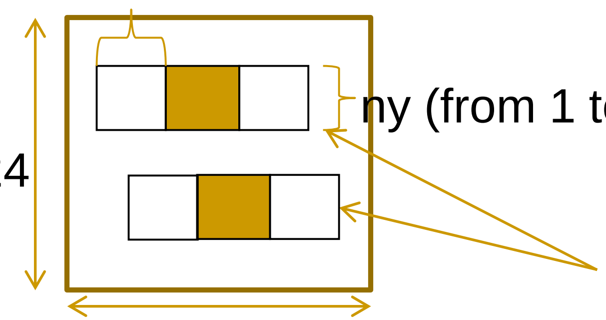
- temp=0; %Type1 feature: block aspect ratio is width=2 units, height=1unit
- for nx=1:win_width/2%nx=no. of x pixels in white area. Min =1,max=win_width/2
-     for ny=1:win_height%ny=no. of x pixels in white area. Min =1,max=win_width
-         number_of_blocks_x=(win_width-2*nx+1);%no.of x Blocks fit in win_width
-         number_of_blocks_y=(win_height-ny+1);%no.of y Blocks fit in win_height
-         temp=number_of_blocks_x*number_of_blocks_y+temp;
-     end
- end
- temp %is the total= 43200



Answer5a2: Class exercise 5a : How many type 3 features in a 24x24 (pixel) window?

- temp=0;
- %Type3: aspect ratio of the feature block, width=3 units, height=1unit
- for nx=1:win_width/3 %nx=no. of x pixels in white area.Min =1,max=win_width/3
- for ny=1:win_height %ny=no. of y pixels in white area.Min =1,max=win_width
- number_of_blocks_x=(win_width-3*nx+1);%no.of x Blocks fit in win_width
- number_of_blocks_y=(win_height-ny+1);%no.of y Blocks fit in win_height
- temp=number_of_blocks_x*number_of_blocks_y+temp;
- end
- end
- N_Type3=temp %answer= 27600

nx (from 1 to win_width/2 pixels for Type3)



ny (from 1 to win_width pixels for Type 3)

win_height=24

win_width=24

Example of Type3 feature blocks (aspect ratio is 3-column x 1-row)

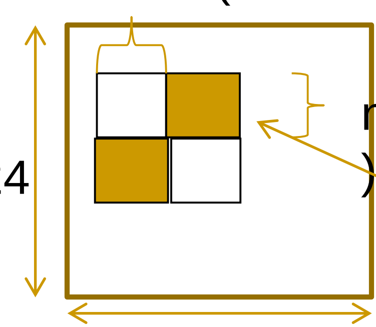
Answer5a3 Exercise 5a : How many type 5 features in a 24x24 (pixel) window?

- temp=0; %---
- %type5: aspect ratio of the feature block, width=2 units, height=2unit
- for nx=1:win_width/2%nx=no. of x pixels in white area.Min =1,max=win_width/2
-     for ny=1:win_height/2%ny=no. of y pixels in white area.Min =1,max=win_width/2
-         number_of_blocks_x=(win_width-2*nx+1);%no.of x Blocks fit in win_width
-         number_of_blocks_y=(win_height-2*ny+1);%no.of y Blocks fit in win_height
-          temp=number_of_blocks_x*number_of_blocks_y+temp;
-     end
- end
- N_Type5=temp %=20736

nx (from 1 to win_width/2 pixels for Type5)

ny (from 1 to win_width/2 pixels for Type 5)

win_height=24



win_width=24

An example of Type5 feature blocks (aspect ratio is 2-column x 2-row)

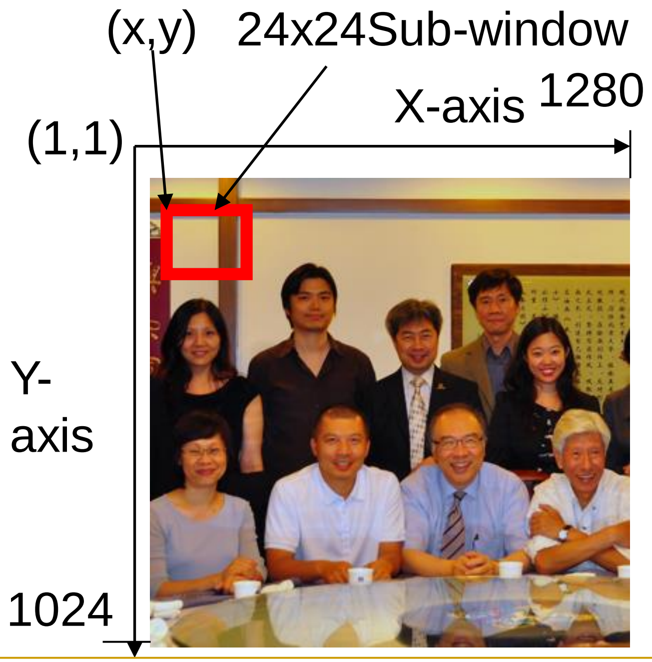
Answer for Exercise 5 and 6:Matlab: for a 24x24 windows, add all types N_type1x2+N_type3x2+N_type5=(43200x2+27600x2+20736)=162336

```
clear; temp=0;
%--matlab program to find number of features %(5 types (columns x rows):
%type1: 2x1; type2: 1x2; type3: 3x1; type 4: 1x3; type 5: 2x2)
%in Viola-Jones face detection cascaded Adaboost algorithm-
%%%%% 2x1 shape : (2 rows x 1 column, same as 1 row x 2 columns) , 2 types
%win_width=24%(you may choose 36 or 24 etc.)
win_width=24%(you may choose 36 or 24 or 12etc.)

win_height=win_width; %x=hornizontal direction; y=vertical direction
%Type1: aspect ratio of the feature block, width=2 units, height=1unit
for nx=1:win_width/2%nx=no. of x pixels of each square. Min =1,max=win_width/2
   for ny=1:win_height%ny=no. of y pixels  of each square. Min =1,max=win_width
      number_of_blocks_x=(win_width-2*nx+1);%no.of x Blocks fit in win_width
      number_of_blocks_y=(win_height-ny+1);%no.of y Blocks fit in win_height
       temp=number_of_blocks_x*number_of_blocks_y+temp;
   end
end
N_Type1=temp
N_Type2=N_Type1  % same as 2 rows x 1 column
pause

temp=0;
%Type3: aspect ratio of the feature block, width=3 units, height=1unit
for nx=1:win_width/3%nx=no. of x pixels  of each square.Min =1,max=win_width/3
   for ny=1:win_height%ny=no. of y pixels of each square.Min =1,max=win_width
      number_of_blocks_x=(win_width-3*nx+1);%no.of x Blocks fit in win_width
      number_of_blocks_y=(win_height-ny+1);%no.of y Blocks fit in win_height
       temp=number_of_blocks_x*number_of_blocks_y+temp;
   end
end
N_Type3=temp
N_Type4=N_Type3  % same as 3 rows x 1 column
pause
%
temp=0; %-------------------------------------------------------------------
%type5: aspect ratio of the feature block, width=2 units, height=2unit
for nx=1:win_width/2%nx=no. of x pixels  of each square.Min =1,max=win_width/2
   for ny=1:win_height/2%ny=no. of y pixels  of each square.Min =1,max=win_width/2
      number_of_blocks_x=(win_width-2*nx+1);%no.of x Blocks fit in win_width
      number_of_blocks_y=(win_height-2*ny+1);%no.of y Blocks fit in win_height
       temp=number_of_blocks_x*number_of_blocks_y+temp;
   end
end
N_Type5=temp
'total'
N_ALL=N_Type1+N_Type2+N_Type3+N_Type4+N_Type5
%Result= 162336 if width =24
%Result= :  816264 if width =36 ( ??or 704004??)
```

Answer 7: The detection challenge

- Use 24x24 base window
- For y=1;y<=1024;y++

  {For x=1;x<=1024;x++{
  - Set (x,y) = the left top corner of the 24x24 sub-window, different scales are needed to be considered too.
  - For the 24x24 sub-window, extract 162,336 features and see they combine to form a face or not.
  - } }
- Exercise 7 : Discuss the number of operations required.
- Conclusion : too slow, solution use boosting



Answer 7:
- possible locations of (x,y)=1024x1280.
- Each (x,y) location, for i=1,2,3..obtain sub-images: each has size (24ix24i) with left-top corner at (x,y) as long as x+24i<1024
- For a sub-image, shrink it to a 24x24 window.
- For each 24x24 window, it has 162336 features to be calculated.

Answer: Class exercise 6.8: Attentional cascade

- Chain classifiers that are progressively more complex and have lower false positive rates: Fill in ?: Name the classifier (1 or 2 or3), explain your answer

Receiver operating characteristic