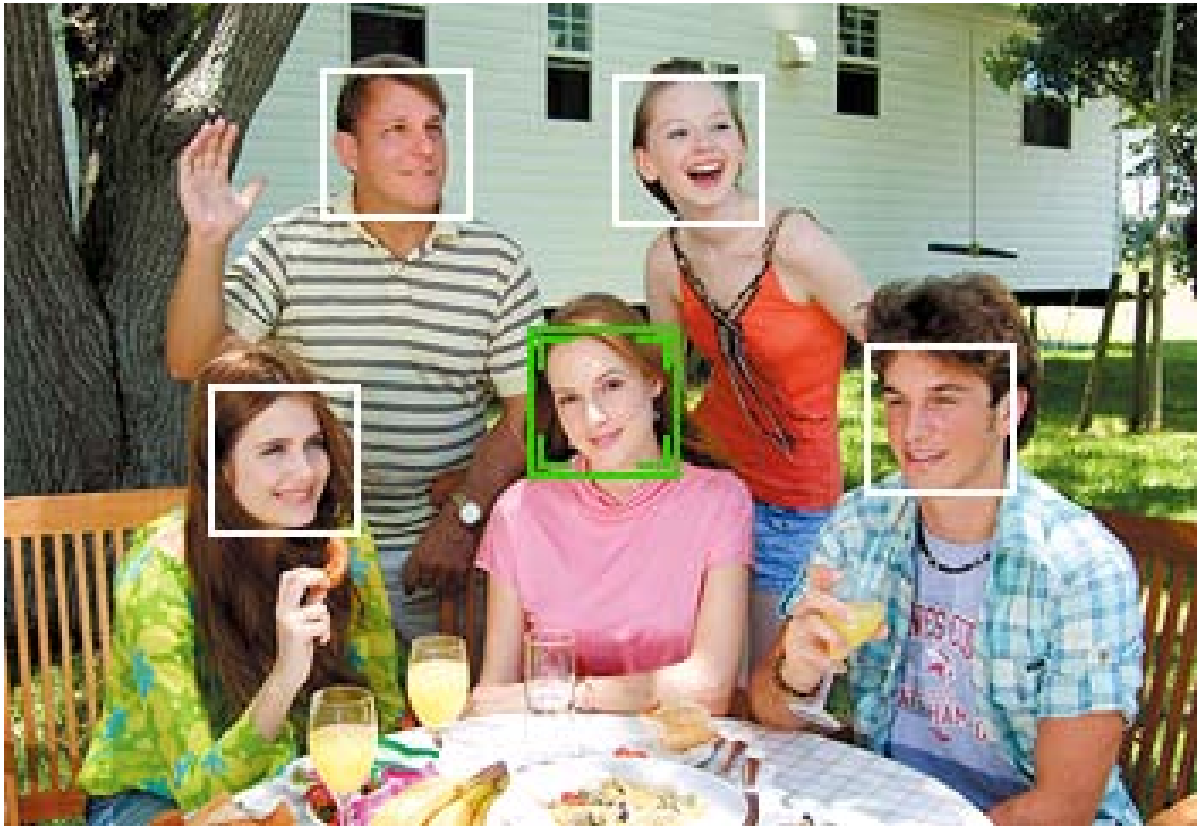
OPPA European Social Fund
Prague & EU: We invest in your future.

Detekce objektů pomocí scanning window



Jiří Matas

Center for Machine Perception

Department of Cybernetics, Faculty of Electrical Engineering

Czech Technical University, Prague

Osnova přednášky:

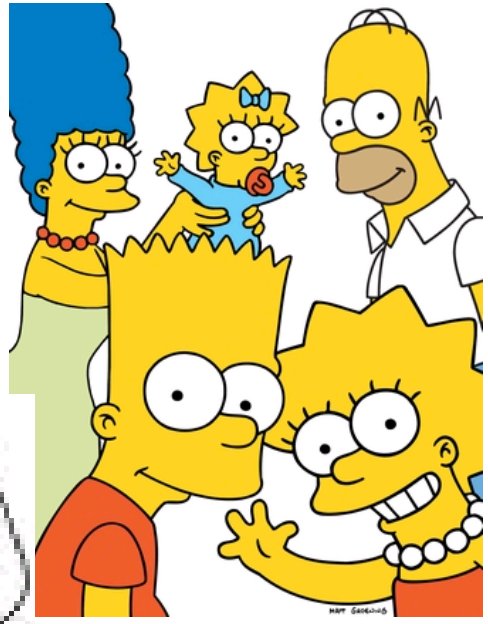
- Formulace úlohy detekce třídy objektů.
- Rěšení pomocí „scanning window“, tj. metodou detekce se strukturou navrženou ve článku Viola-Jones [2001].

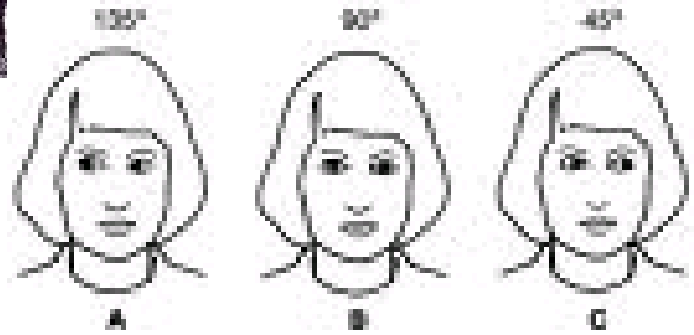
- Metodu vyložíme na příkladu „detekce obličejů“. Metoda byla úspěšně použita na mnoho tříd, např.:
  - automobil, pohled zezadu; SPZ; znak-neznak
  - chodci; obličej, přední pohled; oči

- Na rozdíl od úloh „image retrieval“ a „object recognition“ detekujeme (rozpoznáváme) objekty, které jsme nikdy neviděli, ale víme, že jsou z dané třídy

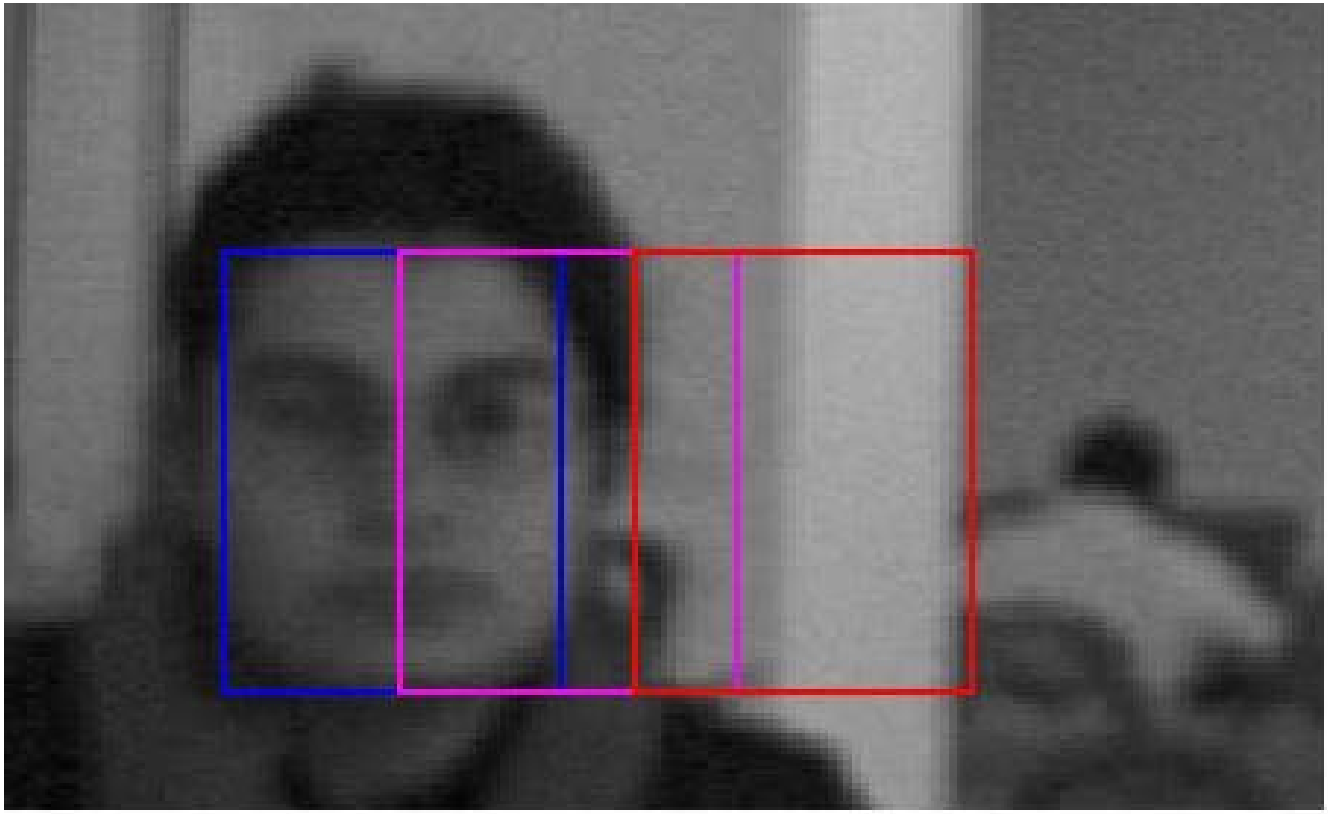
Formulace úlohy „Detekce obličejů“ I.

Definice obličeje: není to tak jednoduché





Co je ještě obličej



Formulace úlohy „Detekce obličejů“ II.

- Co je to obličej? Typická definice:
  - uživatel dodá L_o výřezů, které považuje za obličeje
  - uživatel dodá L_n obrázků, v kterých nejsou obličeje (*vzdáváme nalezení nějaké formální definice obličeje typu „část těla zdravého jedince druhu homo sapiens, která …)*
- Jak specifikovat pozici obličeje : pozice obličeje je dána obdélníkem
- Obdélník je považován za správnou lokalizaci tehdy (např.):
  - jsou-li uvnitř obě oči a pusa, a vzdálenost očí je větší jak půl hrany delší strany obdélníka (*obličej musí mít „oči“, „ústa“*)
  - a tento obličej nemá lepší vysvětlení jiným obdélníkem
- Poznámka: nejedná se o „objevné, hluboké “ definice, ale zachycují požadavky typických aplikací. Lze mít jiné definice, ale musím si je ujasnit. „Standardní význam“ slova nestačí, není dost přesný. *(zkuste najít definici muž-žena, pojmy se zdají dost jasné ….)*

Formulace úlohy „Detekce obličejů“ III.

- Detektor OBLIČEJů v obrázku je algoritmus (funkce), který má na vstupu obrázek, a na výstupu seznam obdélníků, o kterých se domnívá, že obsahují OBLIČEJ.
- Detektor je typicky implementován pomocí klasifikátoru výřezů. Klasifikátor výřezu (oken) je algoritmus (funkce), který má na vstupu obdélníkový výřez, a na výstupu jeden bit, obličej-neobličej.
- Rozhodnutí ale nejsou v jednotlivých oknech nezávislá (viz „ … jiný výřez nevysvětluje obličej lépe“ na předchozí straně).
- V praxi řeší aplikace funkce „potlačení nemaxim“ (non-maximum suppression).
  Správné řešení: úlohy formulovat ne jako klasifikaci výřezů, ale jako značkovací problém (labelling problem).
- Jaké jsou charakteristiky detektoru obličejů?
  Jaké jsou požadované vlastnosti?

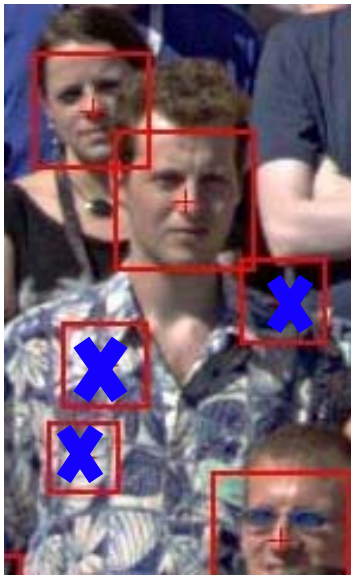
Chyby detektoru obličejů

Chyby detektoru obličejů

1. detekce chybí (false negative)
2. detekce přebývá (false positive)
3. lokalizace

Problémy:

- chyba lokalizace nebo 1. + 2., kde je hranice?
- co je to vlastně obličej?



Požadavky na detektor, měření kvality.

1. vysoká úspěšnost detekce na obličejích, tj. malý počet přehlédnutých obličejů
2. nízký počet „halucinací“ obličejů, tj. malý počet falešných detekcí
3. co nejpřesnější lokalizace
4. rychlost detekce (mnoho aplikací nutně vyžaduje práci v reálném čase, např. ostření fotoaparátu na základě detekce obličeje)
5. další požadavky?

Poznámky.

- Požadavky 1. a 2. „jdou pro sobě“. 1. lze splnit tak, že všechny výřezy jsou klasifikovány jako obličej, 2. pokud obličej nikdy nehlásím. Těžké je udržet nízký počet 1. a 2. zároveň
- Požadavek 4. jde proti 1. a 2. Rychlé rozhodování dělá více chyb (viz. třeba šachy)
- Požadavek 4. jde proti 3. (viz třeba střelba)

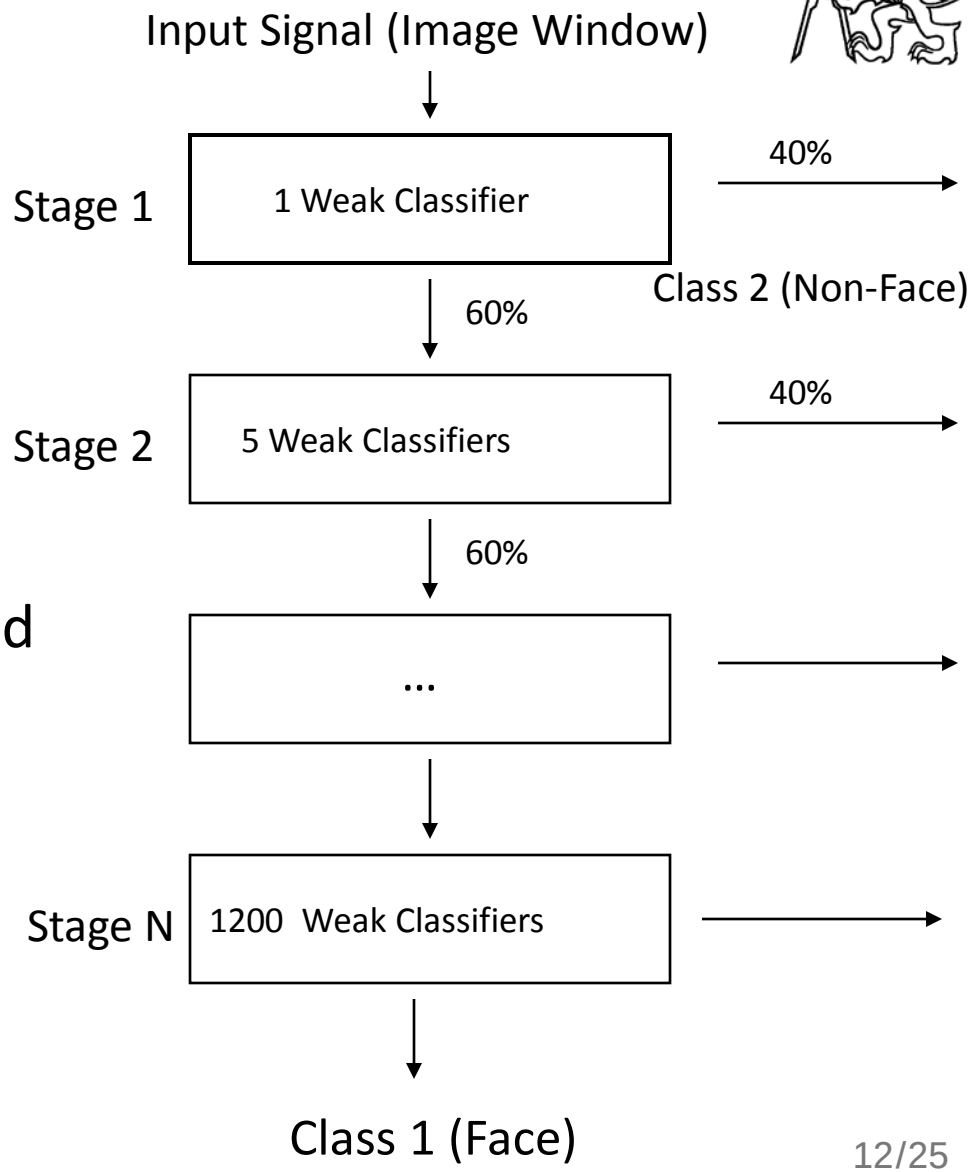
Detekce obličejů metodou Viola – Jones (2001)

- Problém detekce obličejů implementován jako ověření „všech“ nenatočených čtvercových oken v obrázku, a přesto fungoval už v roce 2001 v reálném čase na standardním počítači!
  (Lze zobecnit na množinu obdélníkových oken, i natočených)
- Do té doby jsem si myslel, že to není možné, neboť:
  - oken je hodně *(odhadněte počet v 10 Mpix obrázku!)*
  - v každém okně se vyhodnocuje složitá funkce, která musí rozpoznat všechna pozadí od všech obličejů
- V roce 2001 už bylo zřejmé, že vyučované klasifikátory dosahují lepších výsledků než „ručně“ vytvořené algoritmy (vyučovaný klasifikátor nalézá rozhodovací funkci na základě příkladů)
- Už před rokem 2001 existoval velmi dobrý naučený detektor (Schneiderman-Kanade, 1998). Obrázek zpracovával přes noc, učil se měsíce.

Detekce obličejů metodou Viola – Jones (2001)

Průlom #1: Sekvenční rozhodování

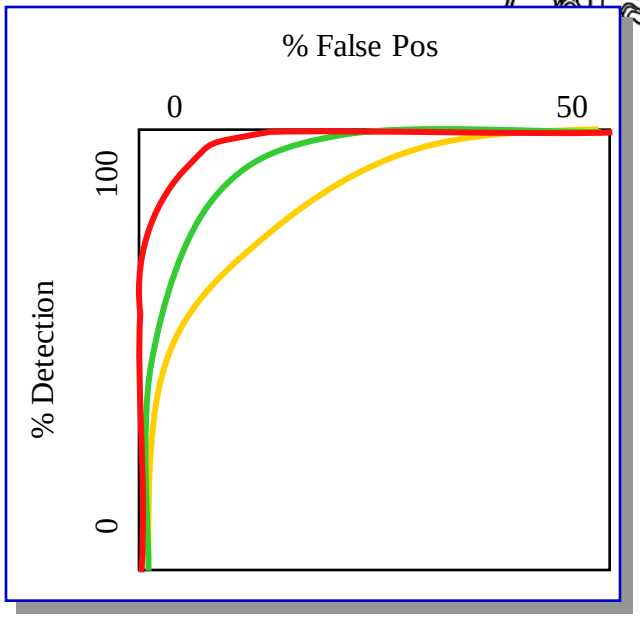
- VJ si uvědomili, že rychlost detektoru závisí na rychlosti klasifikace *neobličejů (*je jich o mnoho řádu více než obličejů*),* navrhli **kaskádu klasifkátorů**
- v kaskádě se rozhododuje sekvenčně
- sekvenční klasifikace je známa od 40 let, ale učení pro sekvenční klasifikátory bylo opomíjeno.
- algoritmus WaldBoost kvazi-optimálně řeší kompromis mezi rychlostí a přesností u klasifikátoru typu AdaBoost.

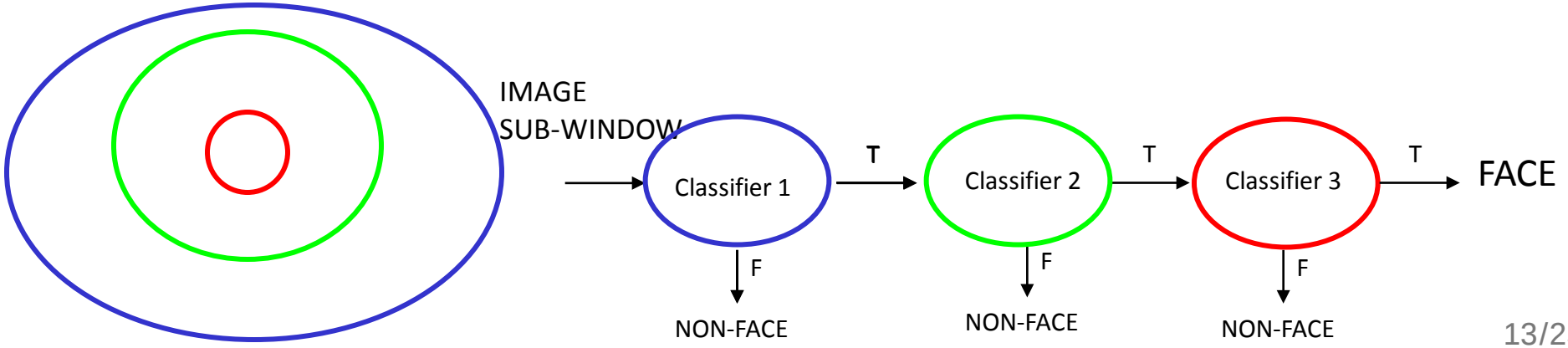


Detekce obličejů metodou Viola – Jones (2001)

Průlom #2: Bootstrap

- VJ učili složitější klasifikátory jen na příkladech, které prošly jednoduššími klasifikátory na počátku kaskády.
- Výsledek: kaskáda se mohla učit na řádově větším počtu příkladů
- V mnoha problémech platí: čím víc dat při učení máš, tím jsi lepší

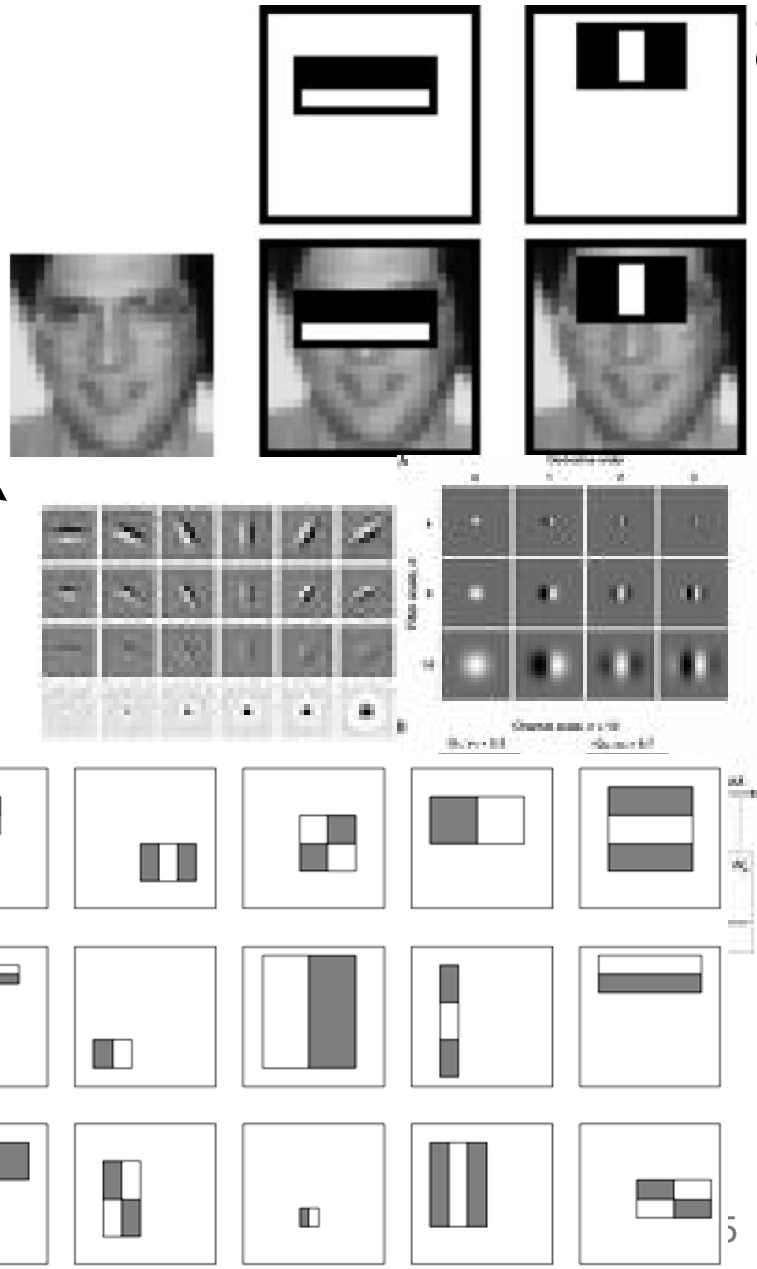




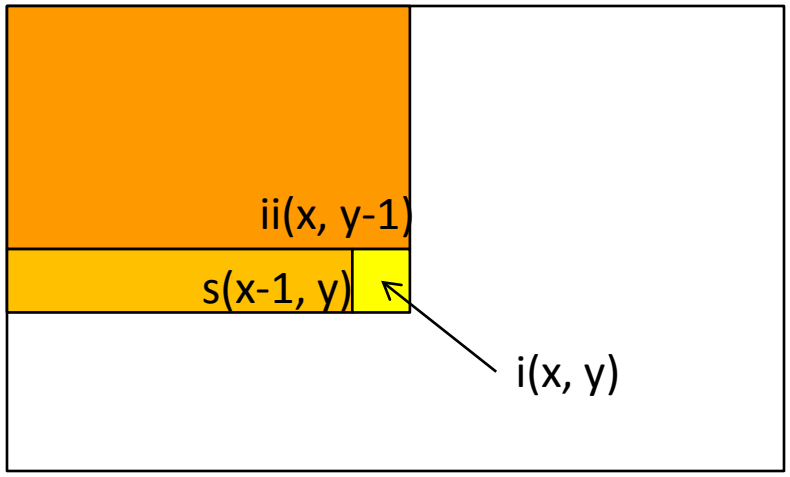
Detekce obličejů metodou Viola – Jones (2001)

Průlom #3: Rychlé vyčislitelné příznaky

- VJ použili k popisu obličejů příznaky, které lze vypočítat extrémně rychle.
- Rozdíly jasů se používaly často, ale jako hladké funkce (rozdíly Gaussiánu, Gaborovy filtry)
- Aproximace po částech konstantními funkcemi, tzv. Haar wavelety, je o několik řádů rychlejší, a jen o málo méně vhodné
- Pro výřez 24x24 pixelů, existuje ~160,000 možných příznaků, které vybrat?



Rychlý výpočet Haar waveletů



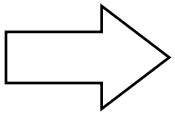
řádkový součet: $s(x, y) = s(x-1, y) + i(x, y)$

Integralní obrázek : $ii(x, y) = ii(x, y-1) + s(x, y)$

MATLAB: ii = cumsum(cumsum(double(i)), 2);

Obrázek

0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0

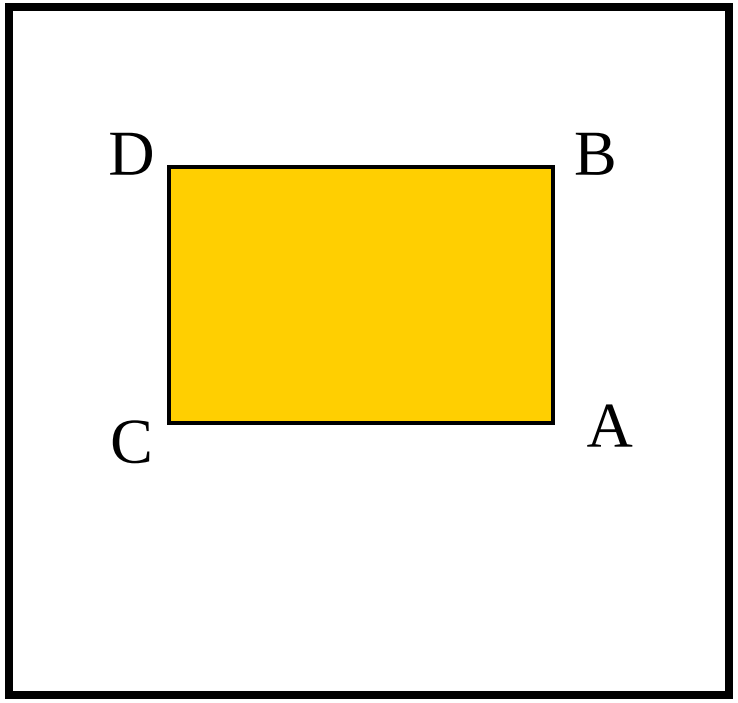


Integrální obrázek

0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21

Rychlý výpočet Haar waveletů

- Hodnoty A,B,C,D vyčteme z integrálního obrázku



- Součet hodnot jasů v původním obrázku lze vypočítat takto:

  sum = A – B – C + D

- Na každý obdélník potřebujeme jen 3 sčítání!

- Poznámka: pomocí integrálního obrázku lze počítat poměrně velkou třídu konvolucí (polynomy, sinus, cosinus)
- Příznaky lze napočítat pro libovolné měřítko stejně rychle.

Detekce obličejů metodou Viola – Jones (2001)

Průlom #4: Moderní metodu strojového učení

AdaBoost (Schapire a Freund, 1997), která učí klasifikátor a zároveň vybírá příznaky.

- Výhody AdaBoostu:
  - Teoreticky podložený (dobrá generalizace)
  - Velmi malá chyba v mnoha aplikacích
  - Velice jednoduchý (“just 10 lines of code” [R. Schapire])

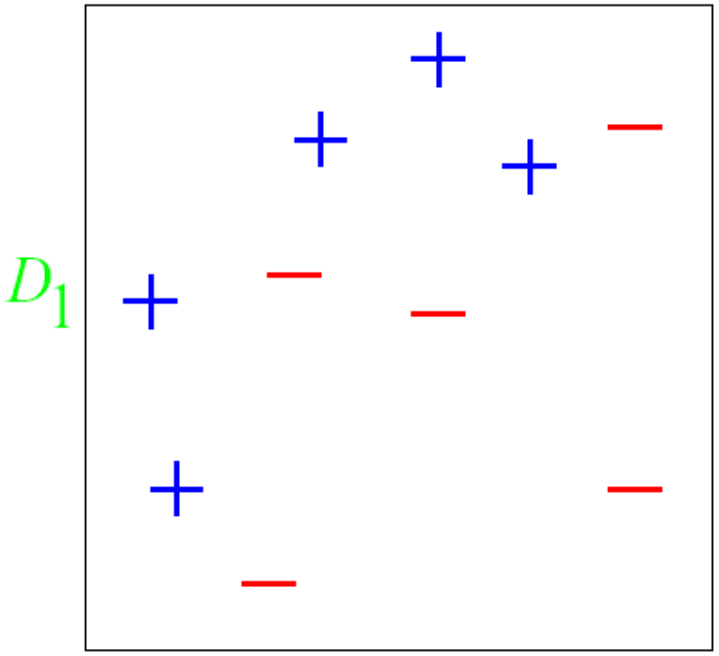
- R. Schapire and Y. Freund získali v roce 2003 Godelovu cenu (jedna z nejprestižnějších cen v teoretické computer science)

Poznámky :

- odstup od výsledku v oblasti strojového učení k jeho použití v počítačovém vidění je velmi krátký

- k průlomu v počítačovém vidění došlo přenosem znalosti

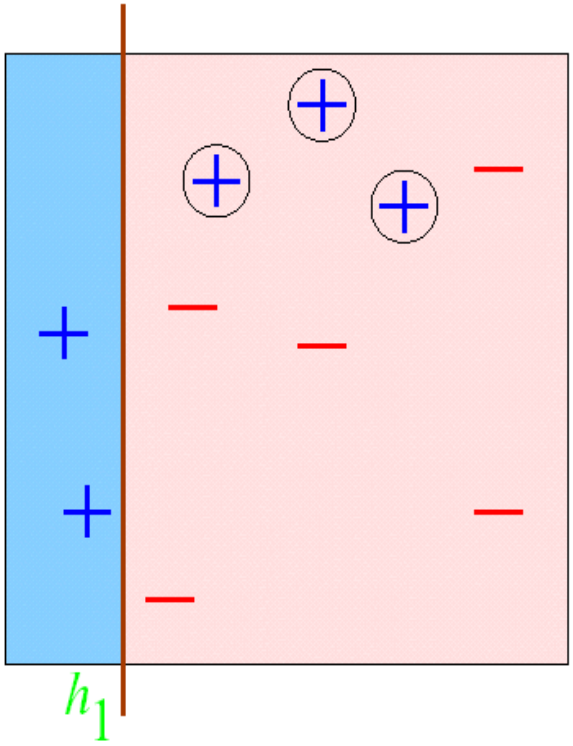
AdaBoost - příklad



Taken from "**A Tutorial on Boosting" by** Yoav Freund and Rob Schapire

AdaBoost - příklad

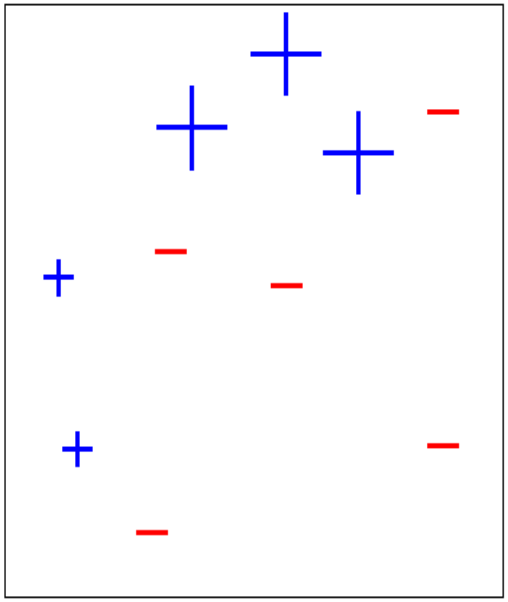
ROUND 1



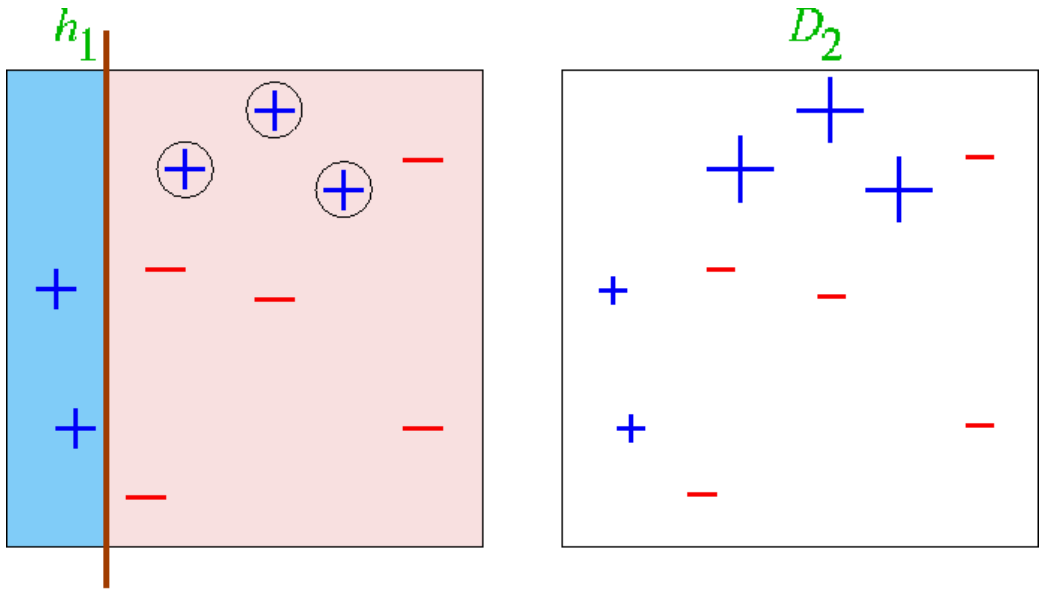
$\varepsilon_1 = 0.30$

$\alpha_1 = 0.42$

D_2



První klasifikátor

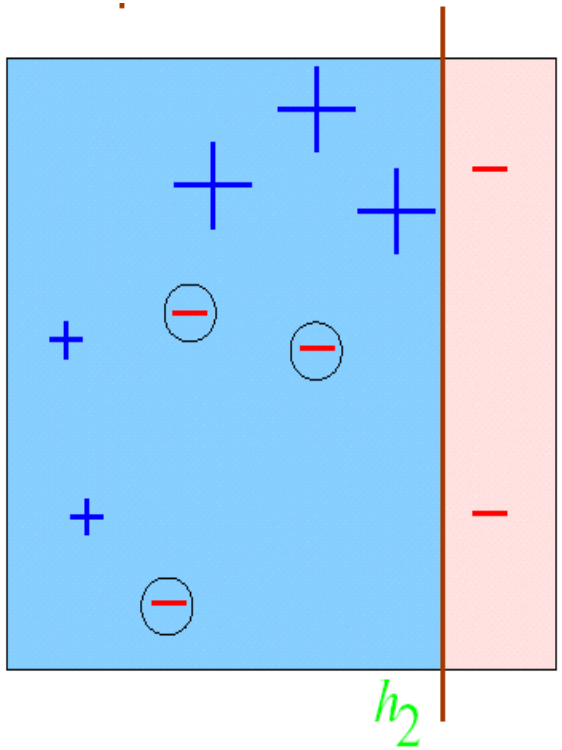


$\varepsilon_1 = 0.30$

$\alpha_1 = 0.42$

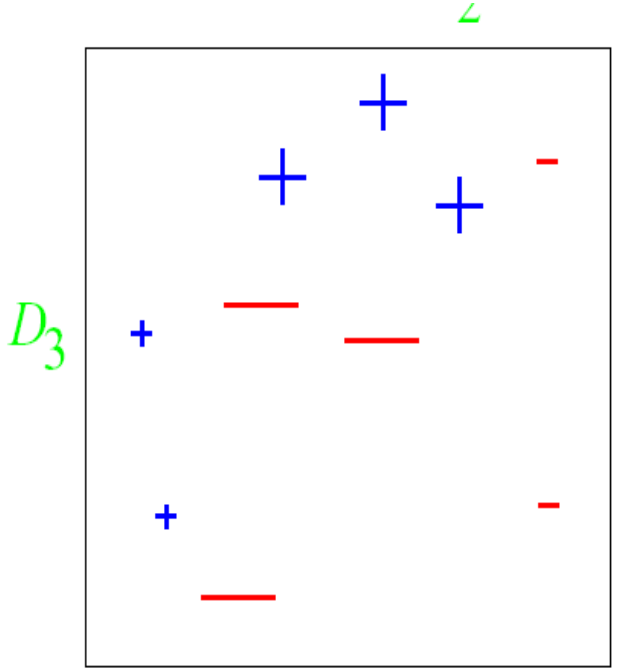
AdaBoost - příklad

ROUND 2

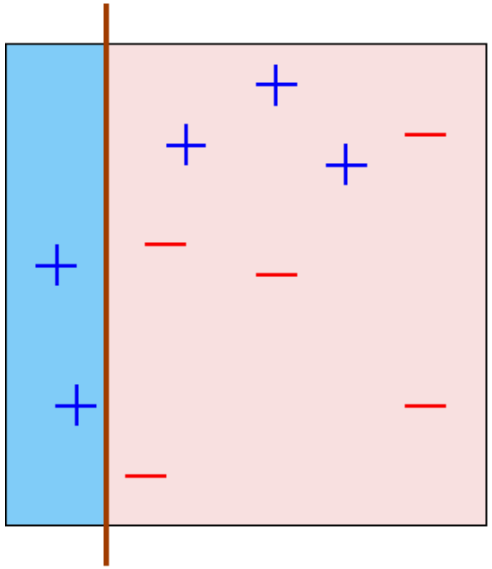


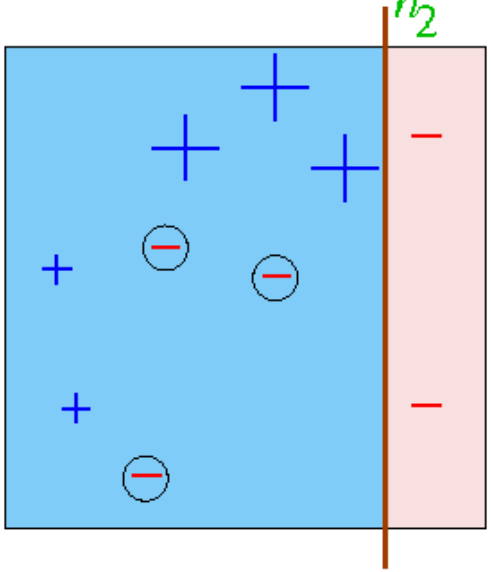
$\varepsilon_2 = 0.21$

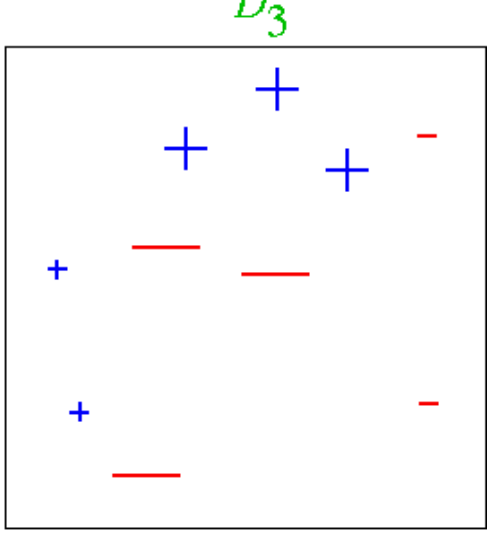
$\alpha_2 = 0.65$



Vybrány dva klasifikátory



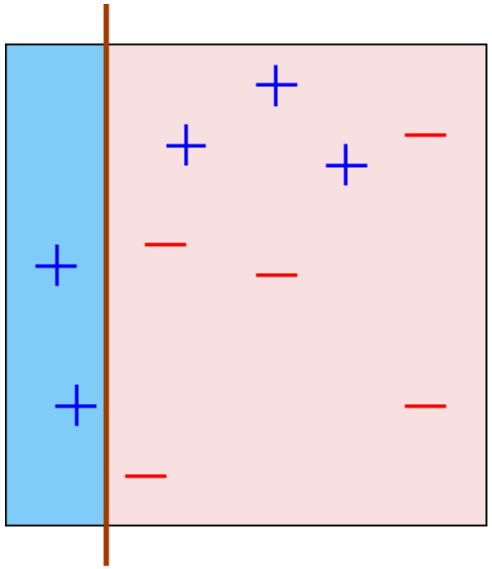


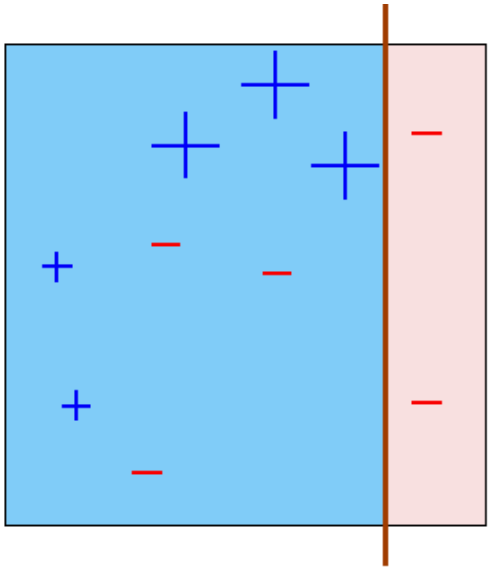


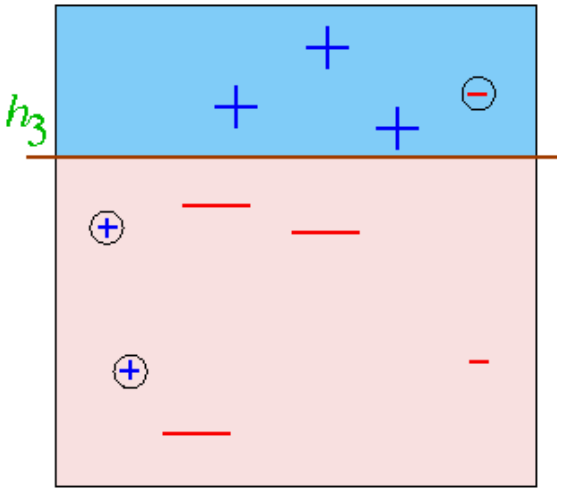
$\varepsilon_2 = 0.21$

$\alpha_2 = 0.65$

Vybrány dva klasifikátory



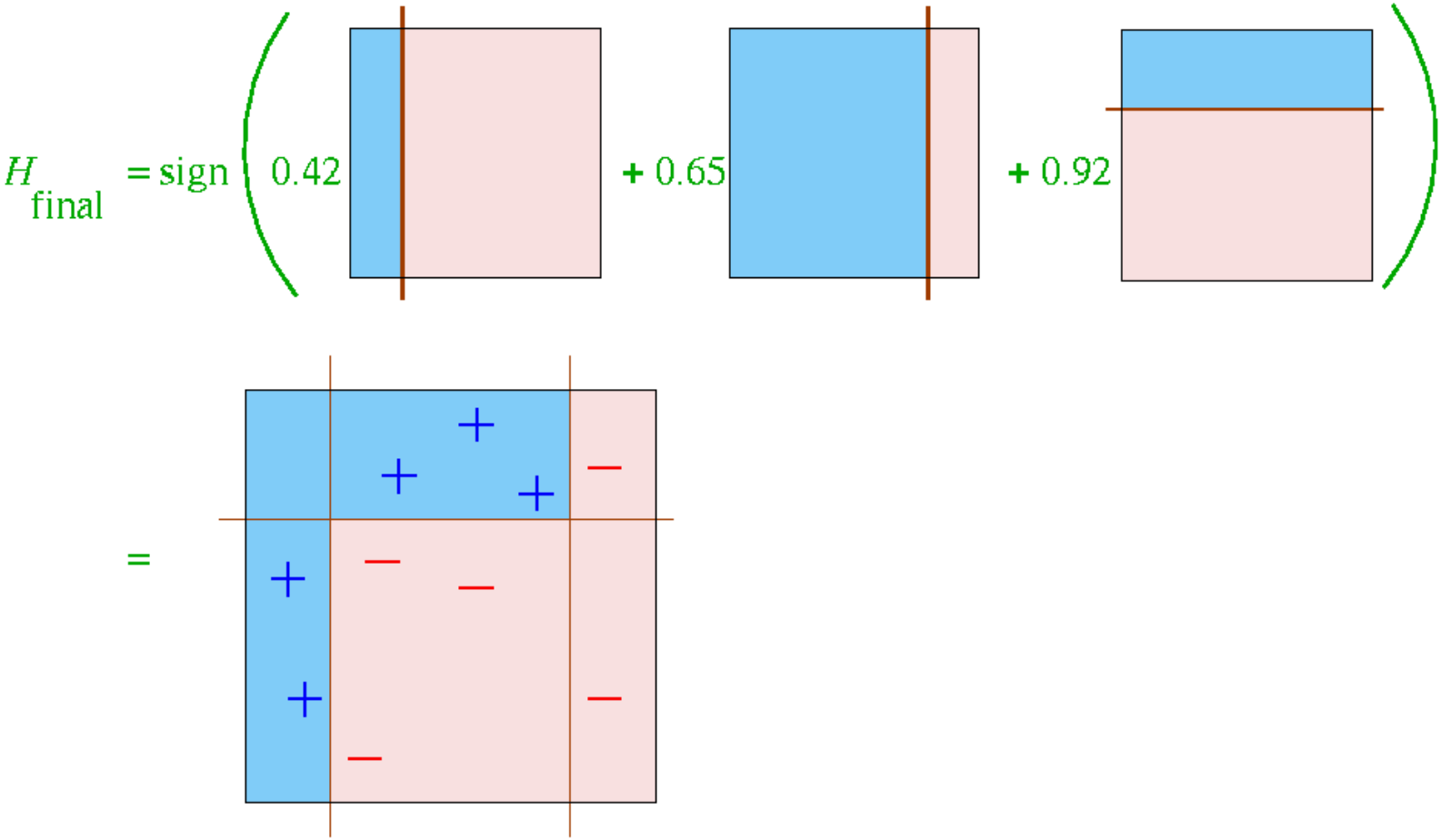




$\varepsilon_3 = 0.14$

$\alpha_3 = 0.92$

Výsledný klasifikátor



AdaBoost: Algoritmus

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train base learner using distribution D_t.
- Get base classifier $h_t : X \to \mathbb{R}$.
- Choose $\alpha_t \in \mathbb{R}$.
- Update:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization f
tion).

typical $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$

where $\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$

the weights of incorrectly classified examples are increased so that the base learner is forced to focus on the hard examples in the training set

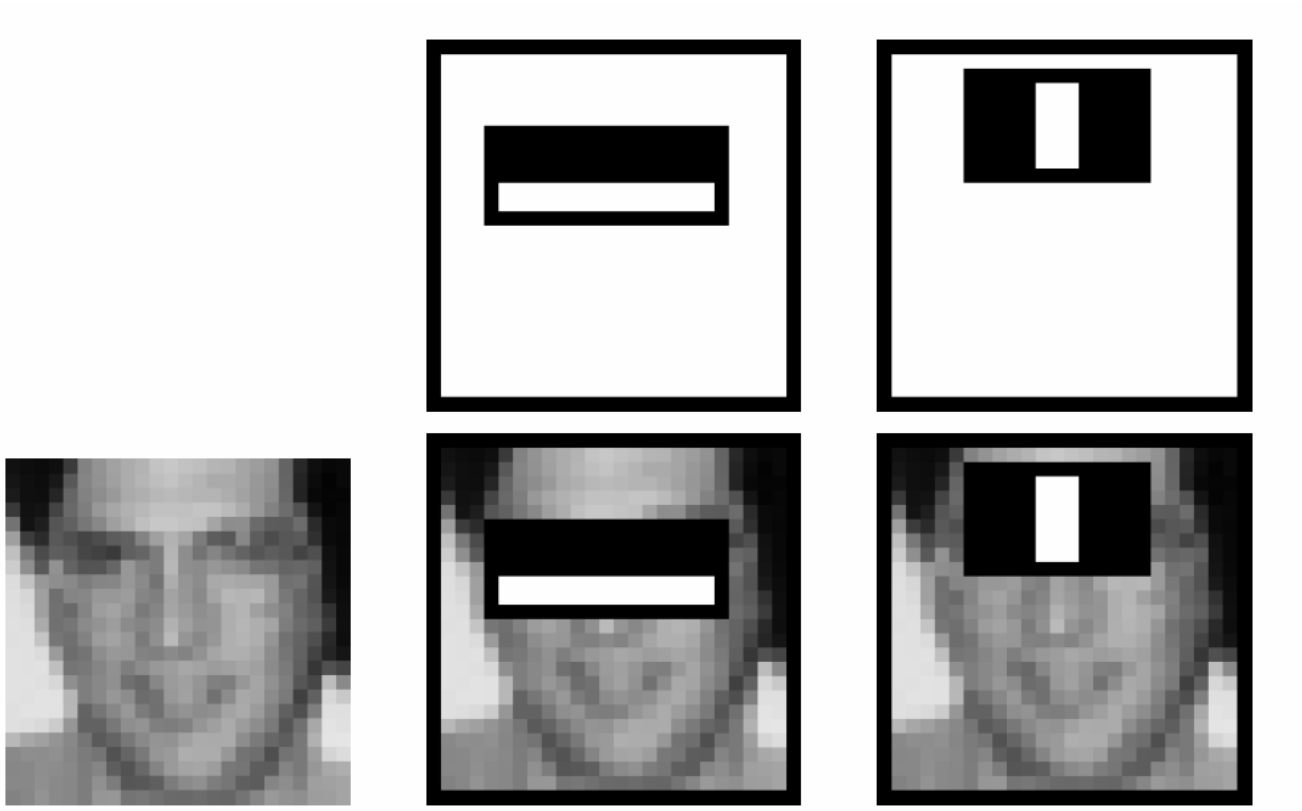
Output the final classifier:

$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

From [R. Schapire, NE&C03]

Detector obličejů Adaboost

- První dva příznaky vybrané metodou Adaboost:



Tyto dva příznaky mají 100% detekci a 50% falešných poplachů.

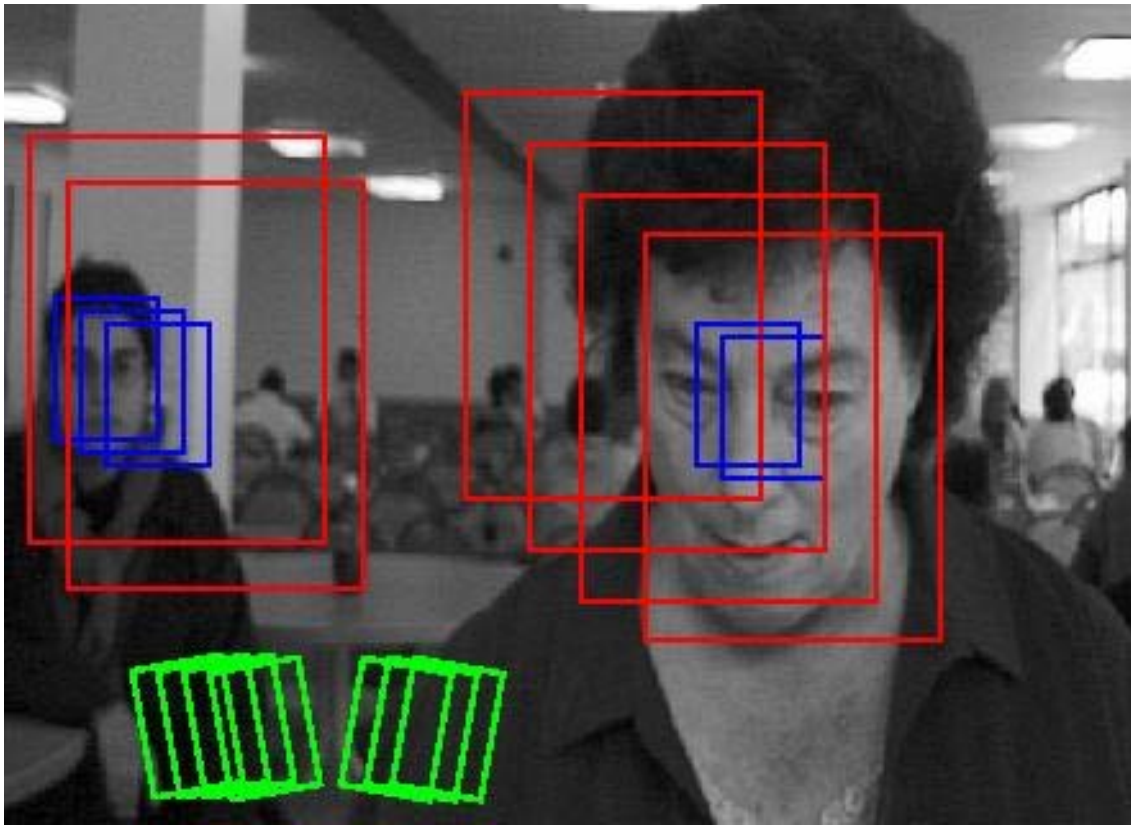
Diskretrizovaný prostor testovaných oken

Klasifikátor okna lze naučit tak, že nemusím testovat *každé okno*

Stačí přibližně:

- posun o 10% šířky okna
- zvětšení o 15% šířky okna
- rotovat o +- 15 stupňů

Poznámka:
o rychlosti (počtu oken) rozhoduje velikost nejmenšího detekovaného obličeje. Celkový čas je součet geometrické řady s $q=1/1.15^2$;$s \approx 4t_0$



Příznaky používané pro V-J detektory

- Výhoda měřítkové nezávislosti Haarových příznaků není příliš důležitá. Pro jakékoliv příznaky mohu vytvořit pyramidu, většinu času výpočtu zabere zpracování nejvyššího rozlišení (viz. pozn.)
- Používá se proto celá řada příznaků, mezi nejpopulárnější patří
  - LBP(local binary patterns), které dosahují např. v úloze detekce obličejů lepší výsledky než Haarovy příznaky
  - HOG (histogram of oriented gradients), $\approx$ zobecnění SIFTu na mřížku NxM, dobré výsledky v detekci chodců
- Příznaky lze v AdaBoostu kombinovat, tj. nemusím řešit úlohu „nejlepších přiznaků“
- Příznaky ale musí mít přibližně stejnou výpočetní složitost, nebo musí byt alg. AdaBoost upraven.

Poznámka:
o rychlosti (počtu oken) rozhoduje velikost nejmenšího detekovaného obličeje. Celkový čas je součet geometrické řady s $q=1/1.15^2$;$s \approx 4t_0$

Slabiny detektorů typu V-J

- Použití pro třídu objektů s velkým rozsahem aspektů je problematické, s dimenzí prostoru rychle roste počet oken (ze 5D: 2D pozice, 1D měřítko, orientace, aspekt)
- … a le v aplikacích často existují silná omezení (měřítko je funkcí polohy, např. sledování dopravy).
- Učení detektoru může trvat hodně dlouho,
  složitost je typicky O(TxFxL), kde T je velikost trénovací množiny, F počet příznaků vyžadujících nový průchod dat a L je délka klasifikátoru AdaBoost
- Nemá žádný model geometrie a fotometrických změn (až na jasovou normalizaci okna)

Ukázky aplikací V-J detektoru

Boosted Classifier for Car Detection,
David C. Lee, Carnegie Mellon University

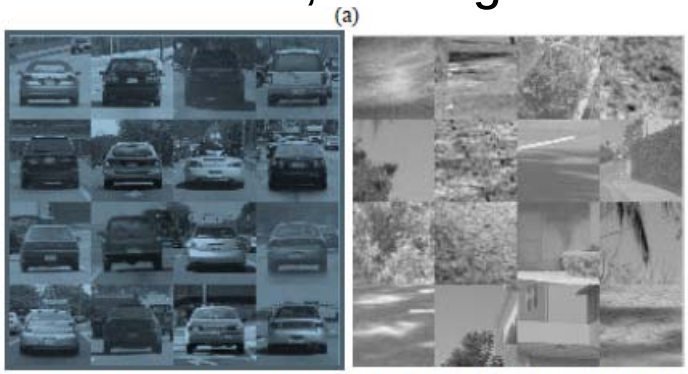


Figure 1 (a) Images collected while driving around Pittsburgh (b) cropped regions of cars (c) cropped regions of non-cars

Figure 2 First three filters selected by Adaboost

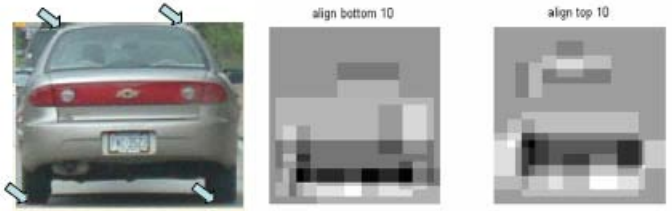


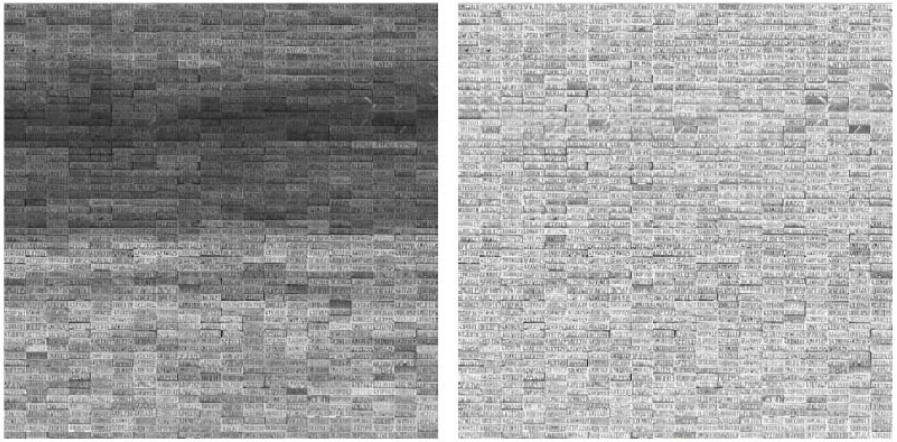
Figure 8 Two ways to align images of cars

Figure 10 The first row and the left most image of second row shows results with no errors. The three rightmost images on the second row show examples where there are either false detections or missed detection.

Ukázky aplikací V-J detektoru

CENTER FOR MACHINE PERCEPTION

License Plate Detection Using AdaBoost, Louka Dlagnekov
Department of Computer Science & Engineering UC San Diego



(a) (b)

Figure 2: (a) 1,200 of 1,520 training examples. (b) Same images variance normalized.

Figure 10: Detected license plate locations on a frame from a live video stream.

Figure 8: Examples of image windows misclassified as license plates.

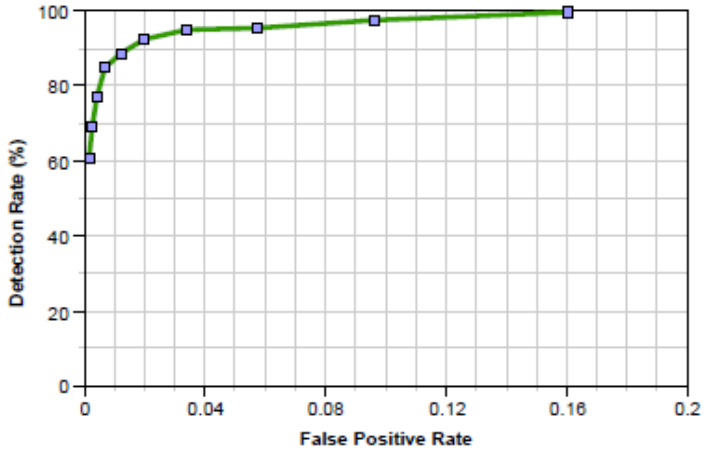
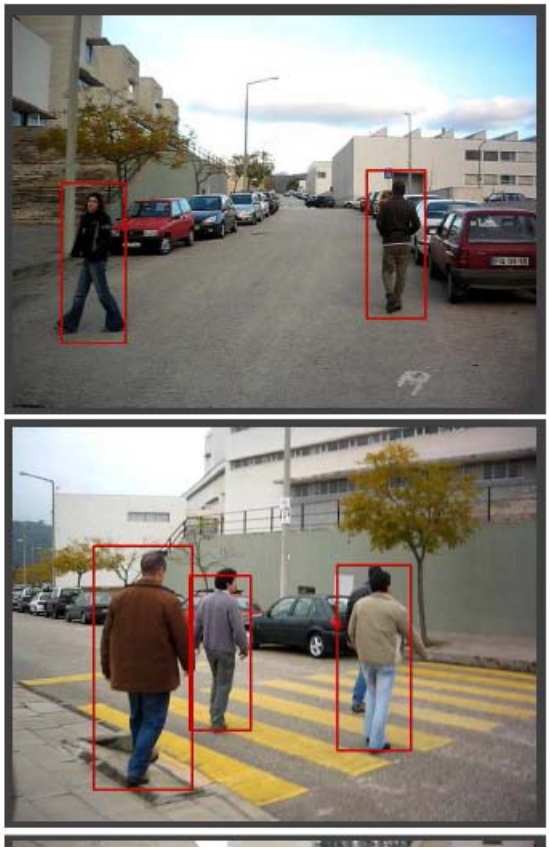


Figure 9: Receiver Operating Characteristic (ROC) curve for the 100 feature classifier.

Ukázky aplikací V-J detektoru



Detekce obličejů – pár poznámek

- V roce 2001 byl publikován článek V-J. V roce 2009 je implementován (samozřejmě ve velmi vylepšené verzi) v mnoha digitálních fotoaparátech.

- Pro detekci ve videosekvencích (např. chodců) lze použít i příznaky napočtené na více snímky (např. změny jasu v čase

- Otázky?

Thank you for your attention.

OPPA European Social Fund
Prague & EU: We invest in your future.