

# Úvod do rozpoznávání, obrazové příznaky

doc. Ing. Josef Chaloupka, Ph.D.



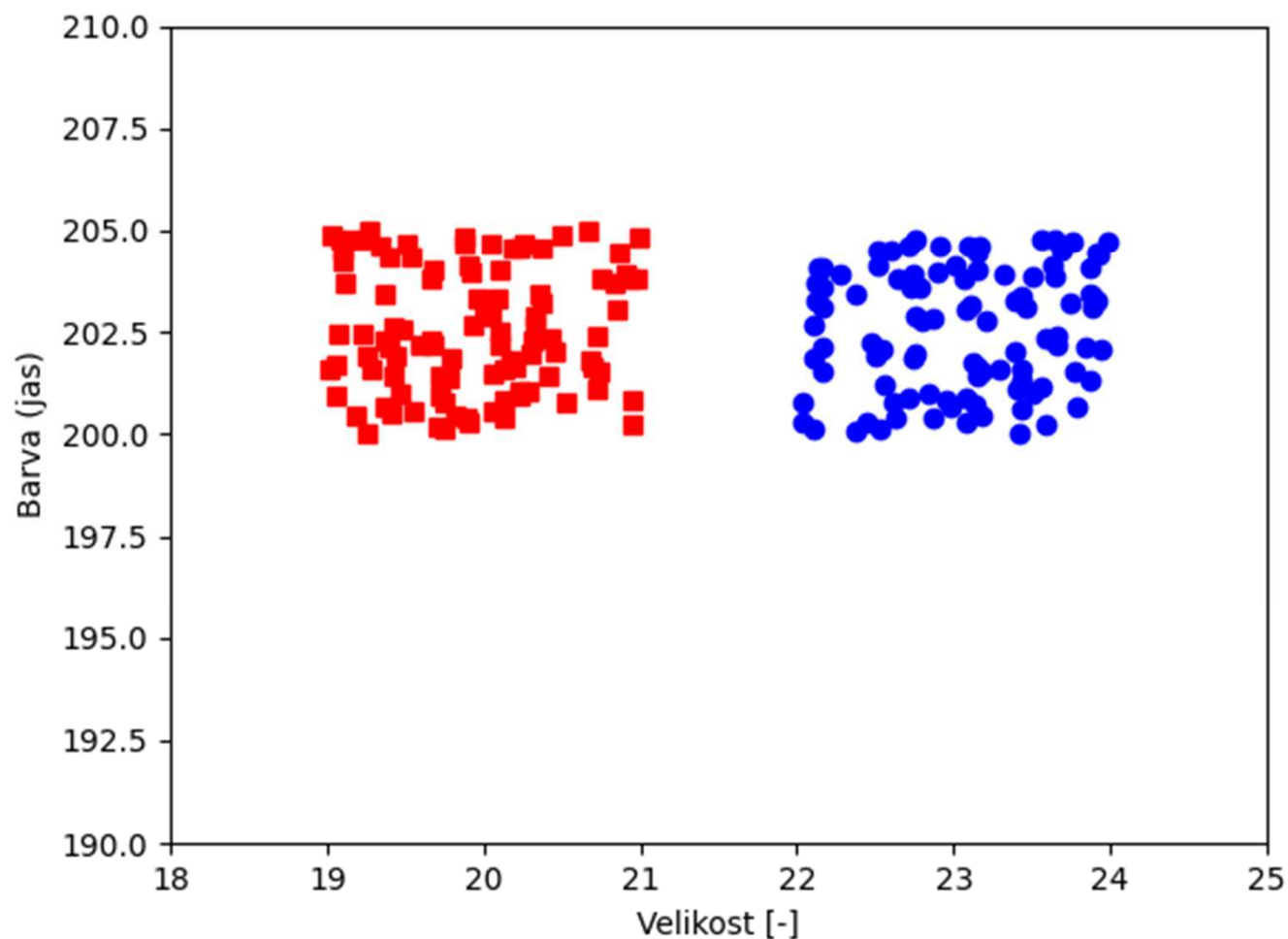
# Volba, výběr a redukce příznaků



Jaké příznaky?

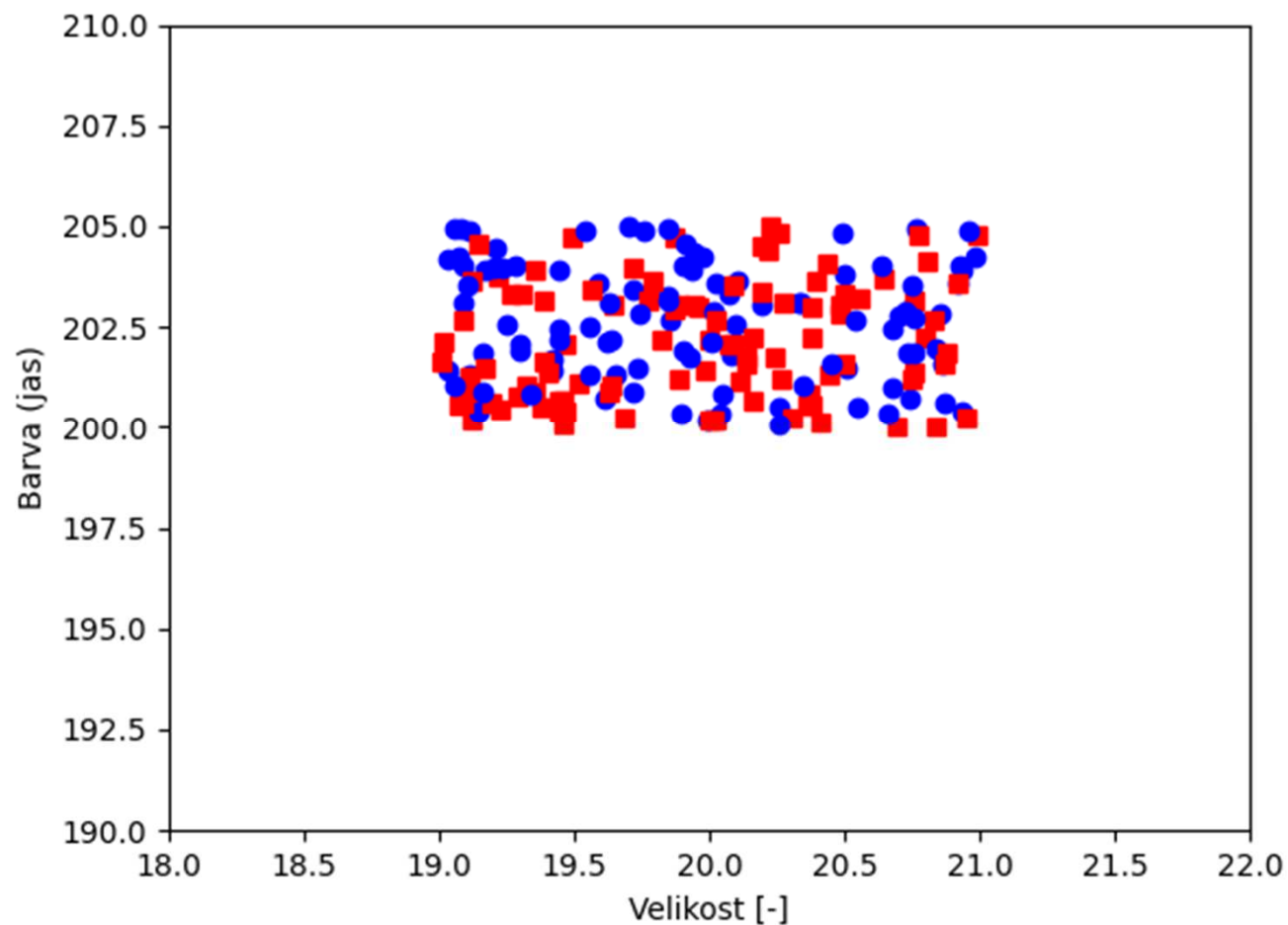
# Volba, výběr a redukce příznaků

Koruna, dvojkoruna (třídy pro rozpoznávání (psy, kočky, králíci...))



Jaké příznaky?

# Volba, výběr a redukce příznaků



Jaké příznaky?

# Volba, výběr a redukce příznaků

## Praktické zkušenosti ukazují, že:

- „kvalita“ příznaků významně ovlivňuje úspěšnost rozpoznávání
- „za určitých podmínek“ lze s více příznaky dosáhnout lepších výsledků
- větší počet příznaků ovšem přináší více výpočtů, delší časy

## Jak najít vhodné příznaky?

- obecná a exaktní odpověď neexistuje
- vychází se většinou z intuice a z dostupnosti různých metod
- často se raději volí větší počet příznaků (a z nich se pak případně analyticky vybírají ty nejdůležitější)

# Volba, výběr a redukce příznaků

## Požadavky na příznaky:

- *praktičnost* - dostupnost a použitelnost při klasifikaci
- *reprezentativnost* - příznaky musí dobře reprezentovat objekty jednotlivých tříd
- *diskriminativnost* – musí umožnit co nejlepší rozlišení mezi třídami
- *nekorelovanost* – příznaky by mezi sebou měly mít co nejmenší vazbu

**Příklad:** rozměr, objem, hmotnost – mohou být u některých předmětů značně korelované příznaky (příčemž každý další již nenese žádnou novou informaci)

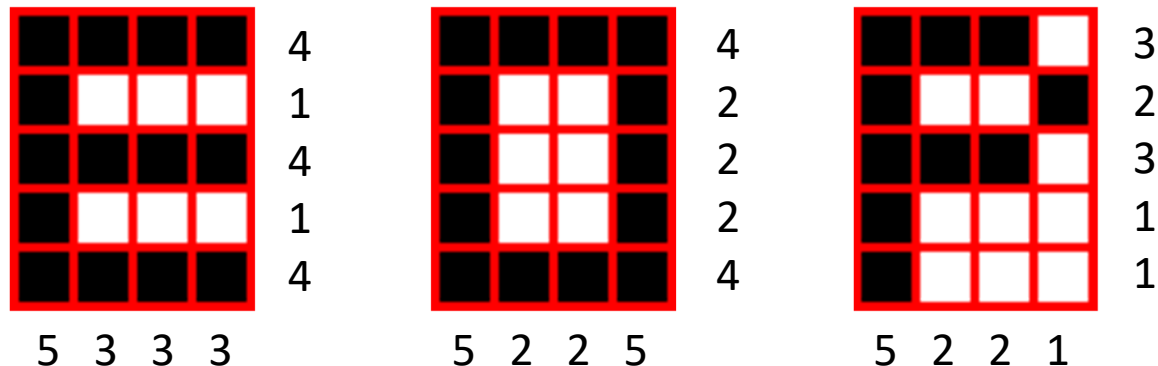
# Jednoduché příznaky pro rozpoznávání obrazů

- **Velikost:** počet obrazových elementů (pixelů), které oblast obsahuje, skutečná velikost objektu >>> pokud je znám převod pixel na m
- **Eulerovo číslo (genus):** z počtu souvislých oblastí  $S$  a počtu děr  $N$ , nemění se při použití geometrických transformací obrazu

$$E = S - N$$

- **Projekce:**

horizontální:  $p(x) = \sum_{y=0}^{M-1} f(x, y)$       vertikální:  $p(y) = \sum_{x=0}^{N-1} f(x, y)$

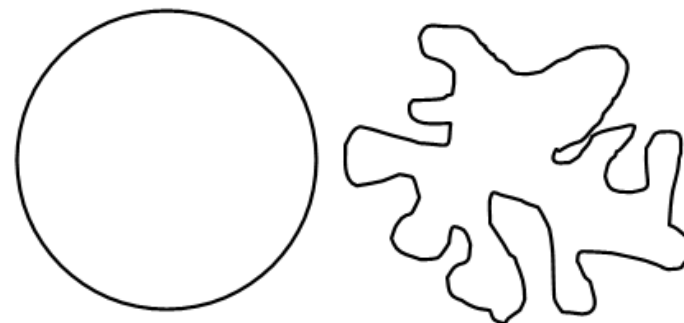




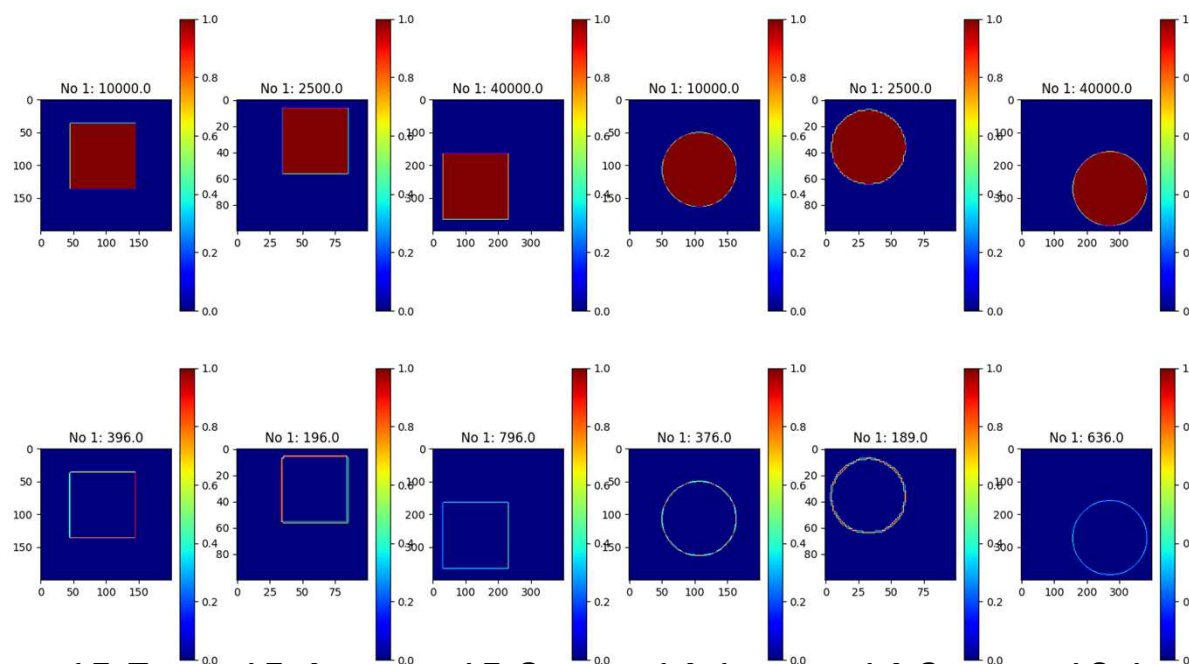
# Jednoduché příznaky pro rozpoznávání obrazů

## ● Nekompaktnost:

$$\text{nekompaktnost} = \frac{(\text{délka hranice oblasti})^2}{\text{velikost}}$$



kompaktní objekt    nekompaktní objekt



Nekompaktnost >>> 15,7

15,4

15,8

14,1


14,3


10,1

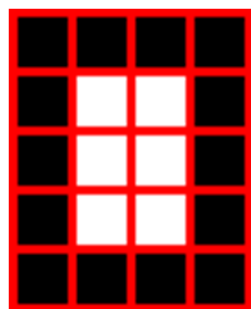


# Jednoduché příznaky pro rozpoznávání obrazů

- Řetězové kódy

	1	
4		2
	3	

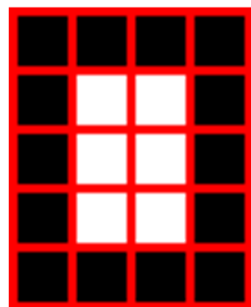
1	2	3
8		4
7	6	5



4, 4, 4, 6, 6, 6, 6, 8, 8, 8, 2, 2, 2, 2



4, 4, 5, 7, 8, 7, 6, 2, 2, 2, 2



4, 6, 8, 2



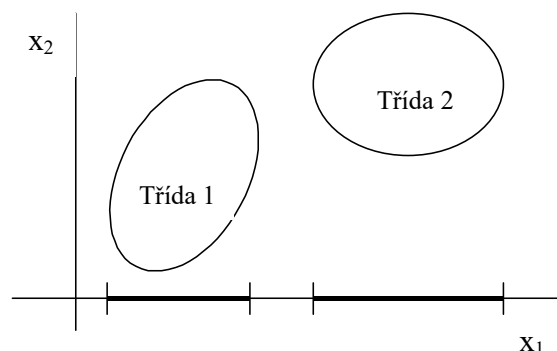
4, 6, 8, 2

# Redukce počtu příznaků

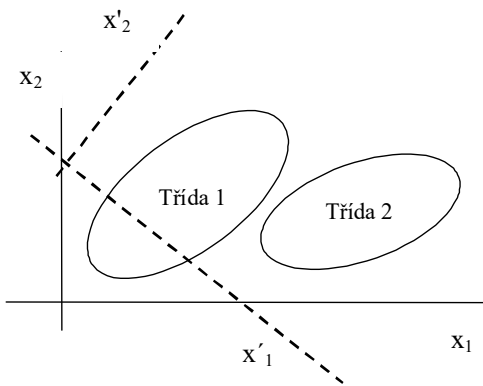
**Cíl:** Z většího počtu příznaků vybrat pouze ty nejvýznamnější z hlediska rozpoznávání.

**Účel:** Snížit zátěž (výpočetní, časovou) vlastního klasifikačního procesu.

**Příklady:**



Lze vystačit pouze s příznakem  $x_1$ , příznak  $x_2$  je v této úloze redundantní



Lze transformovat obrazový prostor a v něm počet příznaků redukovat

# Redukce počtu příznaků

## Principy redukce počtu příznaků:

- transformace a výběr nových příznaků – **extrakce příznaků**
- výběr příznaků podle individuální či skupinové významnosti – **selekce příznaků**

# Redukce počtu příznaků

## Metody založené na transformaci obrazového prostoru

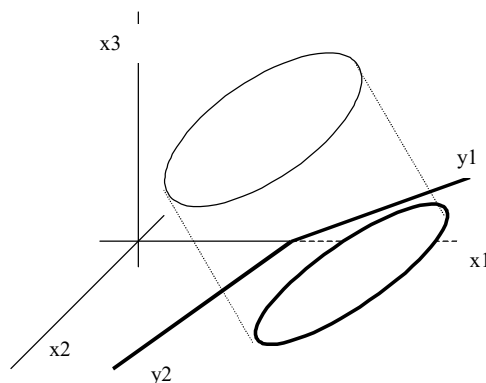
- vycházejí z Karhunen-Loevova rozvoje

### Idea:

a) původní  $n$ -rozměrné příznakové vektory  $x$  převést na  $m$ -rozměrné vektory  $y$  pomocí vhodné lineární transformace  $T$ .

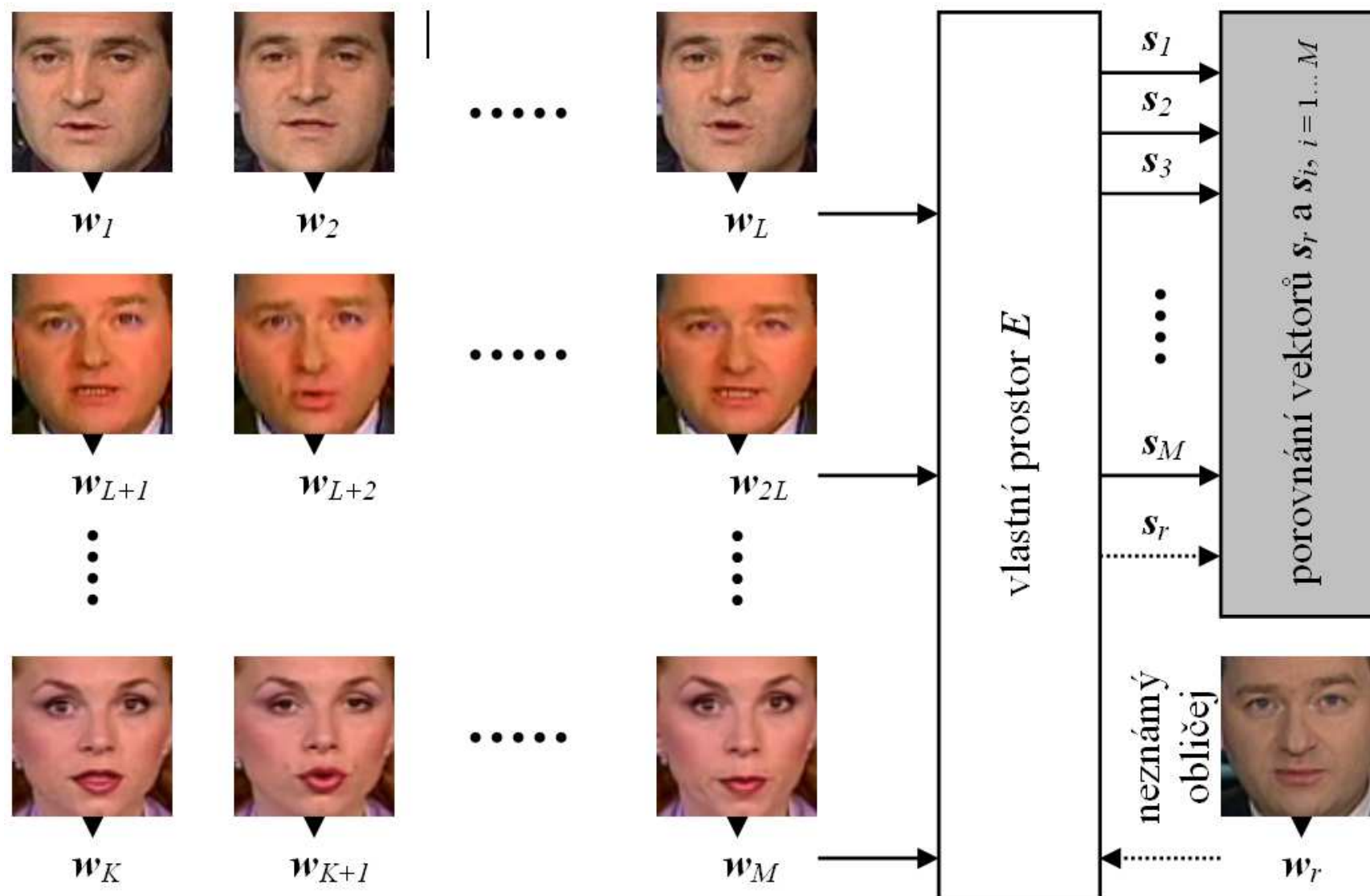
$$y = Tx \quad \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} T_{11} & \cdots & T_{1n} \\ \vdots & \vdots & \vdots \\ T_{m1} & \cdots & T_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad T \text{ je matice } n \times m$$

*interpretace: původní vektory  $x$  se promítají na  $y$  v prostoru s nižší dimenzí*



b)  $T$  se hledá tak, aby vzdálenost  $|y - x|$  (měřená na všech obrazech trénovací množiny) byla minimální – Karhunen-Loevův rozvoj

# Redukce počtu příznaků - PCA



# Redukce počtu příznaků - PCA

**Př. 9 známých obrázků o velikosti 64 x 64 pixelů, 1 neznámý, postup:**

## **Trénovací část**

- 1) Obrázky převedeny na šedotónové, z obrázku vytvořen vektor o délce 4096 – seřazení sloupců (nebo řádků) matice obrazu za sebe
- 2) Ze známých obrázků (vektorů) vytvořena matice  $W_p$  – velikost 4096 x 9
- 3) Z řádků matice  $W_p$  spočítán průměrný vektor  $w_p$  – délka 4096
- 4) Vytvoření matice  $W$  – od sloupců  $W_p$  odečten  $w_p$
- 5) Vytvoření kovarianční matice  $C = W^T * W$  – velikost 9 x 9
- 6) Z matice  $C$  spočítány vlastní čísla a jím náležející vlastní vektory
- 7) Z vlastních vektorů vytvořena matice  $E_p$  – vlastní vektory seřazeny podle velikosti (od největšího k nejmenšímu) vlastního čísla – velikost 9 x 9
- 8) Vytvoření matice (vlastní prostor – EigenSpace)  $E = W * E_p$  – velikost 4096 x 9
- 9) Projekce známých vektorů do vlastního prostoru  $PI = E^T * W$

## **Testovací část**

- 1) Převedení neznámého obrázku do stupně šedi a vytvoření vektoru  $w_p$
- 2) Vektor  $w_u = w_p - w_p$
- 3) Projekce neznámého vektoru  $PT = E^T * w_u$
- 4) Porovnání známých příznakových vektorů  $PI(i)$  a neznámého  $PT$  – např. dle minimální vzdálenosti

# Redukce počtu příznaků

## Metody založené na transformaci obrazového prostoru

### Poznámky:

- 1) Existuje několik variant výše uvedené metody
  - a) s využitím autokorelační matice (preferuje vliv umístění obrazů v prostoru)
  - b) s využitím disperzní matice (preferuje vliv rozptylů)
  - c) s nebo bez respektování rozložení jednotlivých tříd
- 2) Metoda je dobře teoreticky rozpracována, avšak často jen pro speciální případy
- 3) Metoda je výpočetně náročná, a to jak ve fázi trénování (výpočet transformace), tak i při vlastním rozpoznávání (přepočítávání příznakových vektorů).
- 4) **Nové příznaky jsou jen těžko interpretovatelné.**
- 5) Při snižování počtu příznaků se nebere v úvahu vlastní proces rozpoznávání



# Rozpoznávání objektů – minimální vzdálenost

V městských blocích

$$d_B(s_r, s_i) = \sum_{m=1}^M |s_r(m) - s_i(m)|$$

Euklidova vzdálenost

$$d_E(s_r, s_i) = \sqrt{\sum_{m=1}^M (s_r(m) - s_i(m))^2}$$

Kosinová vzdálenost

$$d_C(s_r, s_i) = \frac{\sum_{m=1}^M s_r(m) s_i(m)}{\sqrt{\sum_{m=1}^M s_r(m)^2 \sum_{m=1}^M s_i(m)^2}}$$

Mahalanobisova vzdálenost

$$d_M(s_r, s_i) = - \sum_{m=1}^M \frac{1}{\sqrt{\lambda_i}} s_r(m) s_i(m)$$

# Rozpoznávání objektů – minimální vzdálenost

## **Pravidlo nejbližšího souseda - NN** (Nearest Neighbour)

neznámý vzorek se zařadí do té třídy, k jejímuž představiteli má nejmenší vzdálenost

## **Pravidlo k nejbližších sousedů - kNN** (k Nearest Neighbours)

neznámý vzorek se zařadí do té třídy, jejíž představitelé jsou nejvíce zastoupeny v uspořádané k-tici nejbližších sousedů

## **Reprezentace pomocí etalonů:**

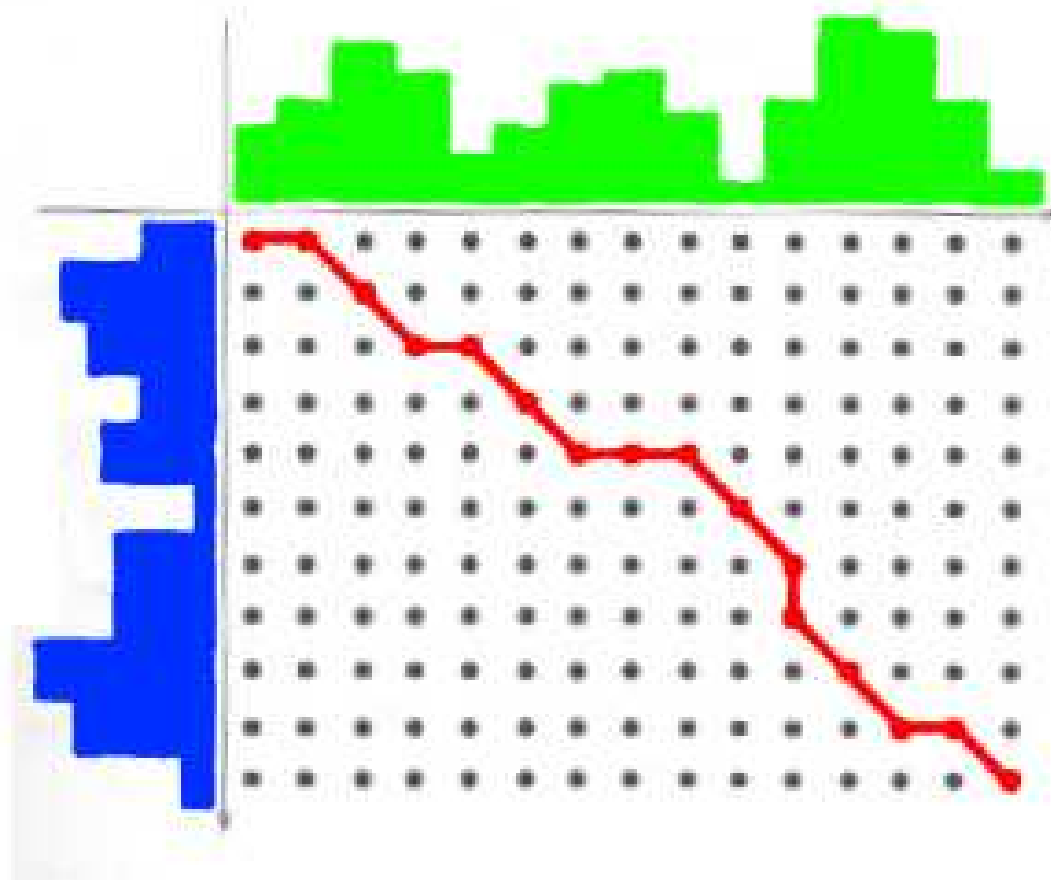
Každá třída je reprezentována **etalonem** - vzorkem třídy, který ji nejlépe reprezentuje ve smyslu minimální vzdálenosti.

Etalon je buď skutečným prvkem třídy, nebo může vzniknout výpočtem, např. průměrováním, z příznakových vektorů třídy.

Při klasifikaci se měří vzdálenosti  $|\mathbf{x} - \mathbf{e}_r|$  a vybere se  $T_r$ , aby  $|\mathbf{x} - \mathbf{e}_r| = \min_{s=1,2..R} |\mathbf{x} - \mathbf{e}_s|$

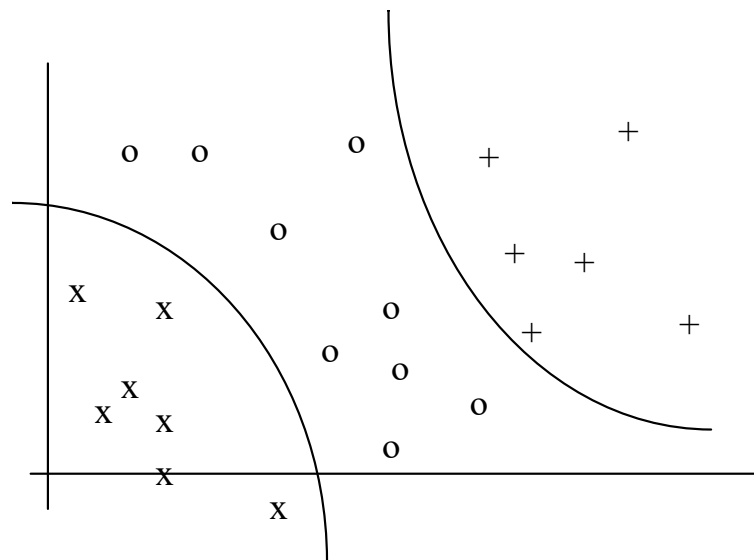
# Rozpoznávání objektů – minimální vzdálenost

Dynamické borcení času: DTW - Dynamic time warping



# Rozpoznávání objektů – Metoda diskriminačních funkcí

vychází z předpokladu, že obrazový prostor lze rozdělit na disjunktní části pomocí  
rozdělujících nadploch:  
v E2 ..... rozděluje křivky  
v E3 ..... rozděluje plochy



Rozděluje nadplochy lze určit pomocí **diskriminačních funkcí**  $g_1, \dots, g_R$ ,

přičemž  $g_r$  je vybrána tak, aby pro všechna platilo  $g_r(x) > g_s(x) \quad s = 1, \dots, R, \quad s \neq r$

Rozděluje plocha mezi třídami  $T_r$  a  $T_s$  je dána rovnicí:  $g_r(x) - g_s(x) = 0$

# Rozpoznávání objektů – metoda max. pravděpodo.

(též nazývaná metoda minimální chyby)

Princip: každá třída je reprezentována

a) **apriorní** pravděpodobností třídy  
pravděpodobností výskytu prvků této třídy  $P(T_r)$

musí platit 
$$\sum_{r=1}^R P(T_r) = 1$$

b) **podmíněnou hustotou pravděpodobností**  $p(\mathbf{x} | T_r)$   
udává rozložení pravděpodobnosti vektoru příznaků  $\mathbf{x}$  pro třídu  $T_r$

Trénování: pro každou třídu se na trénovací množině určí (odhadnou) výše uvedené pravděpodobnosti

Rozpoznávání: aplikace Bayesova pravidla

$$P(T_r | \mathbf{x}) = \frac{p(\mathbf{x} | T_r)P(T_r)}{p(\mathbf{x})}$$

$P(T_r | \mathbf{x})$  ... **aposteriorní** pravděpodobnost, že  $\mathbf{x}$  patří do třídy  $T_r$

$p(\mathbf{x}) = \sum_{i=1}^R p(\mathbf{x} | T_i)P(T_i)$  .... absolutní pravd. hustota rozložení vektoru příznaků  
(nezávisle na třídě)

# Rozpoznávání objektů – metoda max. pravděpodo.

## c) Metoda maximální pravděpodobnosti – příklad

V útulku se nachází 70% psů a 30% koček.

20% koček a 10% psů je černých.

Z dálky kamerou snímáme černé zvíře – jaká je pravděpodobnost, že to je pes?

- **apriorní** pravděpodobnost:

třída - psy  $P(T_1) = 70\%$ , kočky  $P(T_2) = 30\%$   $\sum_{r=1}^R P(T_r) = 1$

- **podmíněnou hustotou pravděpodobností**

náhodně vybraný pes je černý  $p(\mathbf{x} | T_1) = 10\%$

náhodně vybraná kočka je černá  $p(\mathbf{x} | T_2) = 20\%$

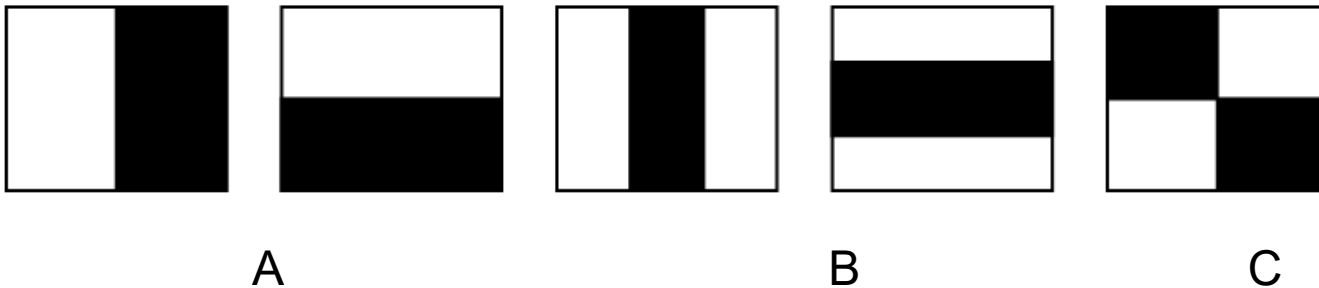
náhodně vybrané zvíře je černé  $p(\mathbf{x}) = 0,7 \times 0,1 + 0,3 \times 0,2 = 0,13$

pozorované černé zvíře je pes  $p(T_1 | \mathbf{x}) = (0,1 \times 0,7) / 0,13 = 54\%$

pozorované černé zvíře je kočka  $p(T_2 | \mathbf{x}) = (0,2 \times 0,3) / 0,13 = 46\%$

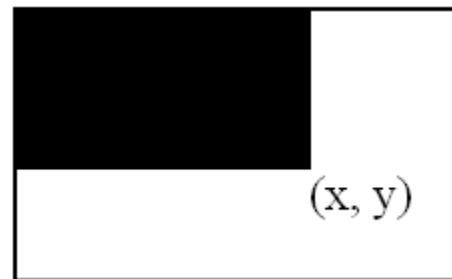
# AdaBoost a “Haar-like” příznaky

- Viola-Jonesův detektor (VJD) obličejů využívá příznaky podobné Haarovým báзовým funkcím a klasifikátor založený na AdaBoost algoritmu



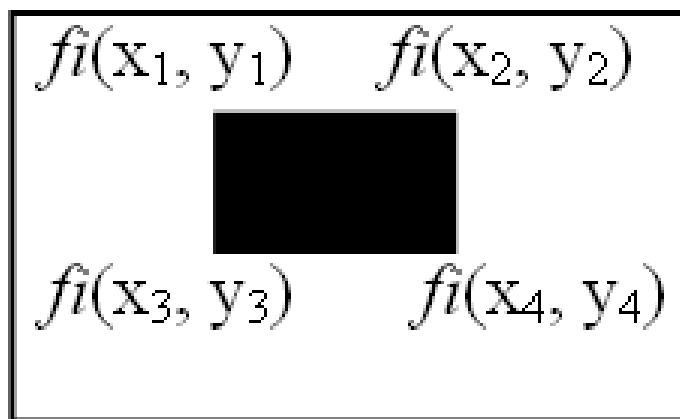
příznak - suma hodnot obrazových bodů nacházející se pod bílou oblastí 2D funkce mínus suma hodnot obrazových bodů pod černou oblastí funkce, zjednodušení:

**Integrovaný obraz**





# AdaBoost a “Haar-like” příznaky

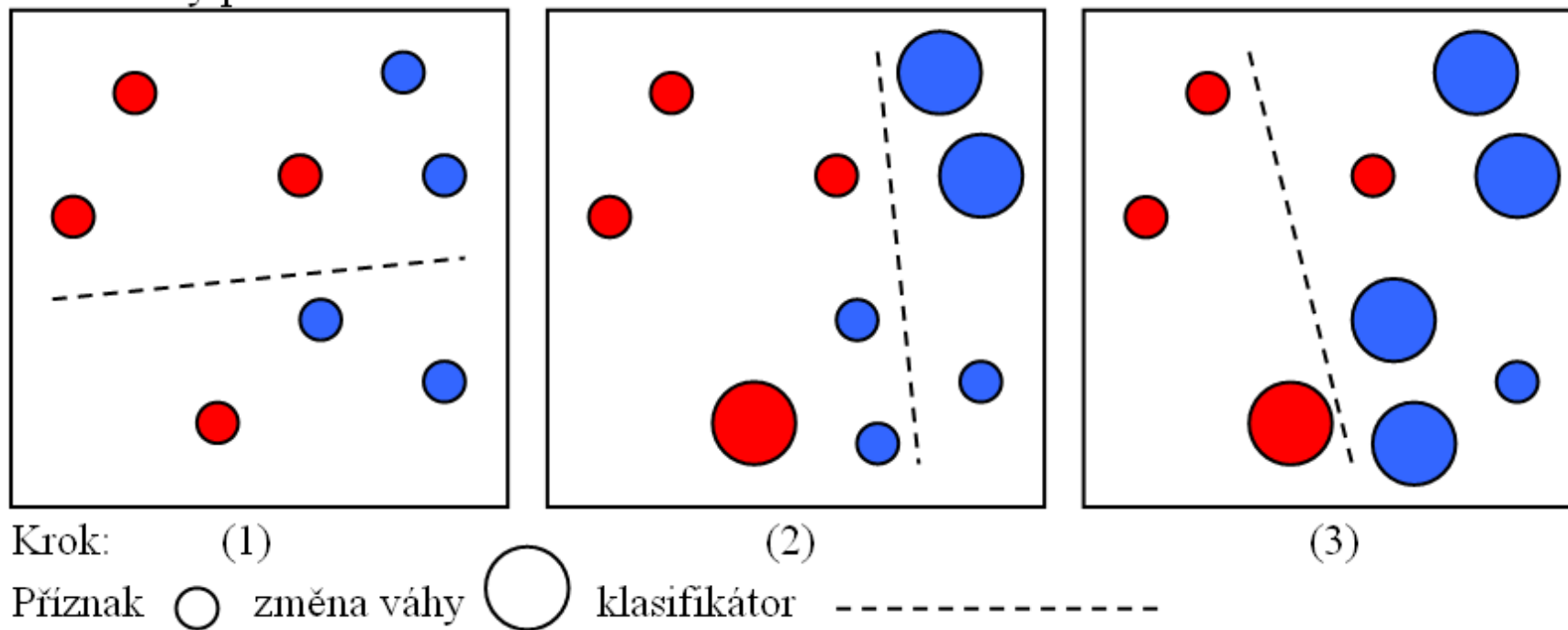


- suma byla spočítána ze čtyř hodnot na základě využití dvou operací sčítání a jedné operace odčítání:  $(f_i(x_1, y_1) + f_i(x_4, y_4)) - (f_i(x_3, y_3) + f_i(x_2, y_2))$
- Pro příznak typu A (obr. 2.8) by poté bylo potřeba použít 6 hodnot z integrálního obrazu, pro příznak B by to bylo 8 a 9 pro příznak typu C

# AdaBoost a “Haar-like” příznaky

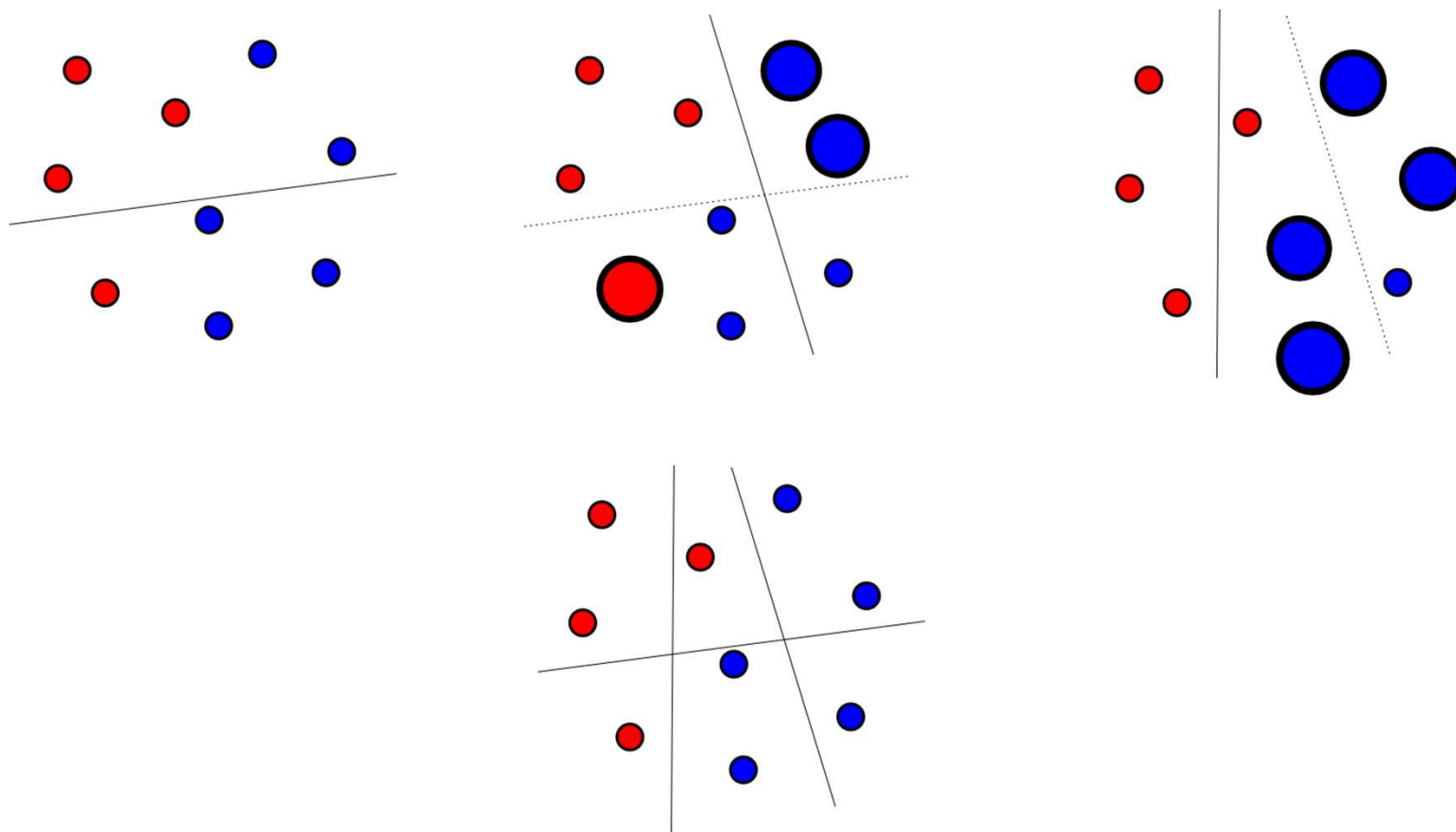
- Algoritmus AdaBoost

Příznakový prostor



# AdaBoost a “Haar-like” příznaky

- Algoritmus AdaBoost



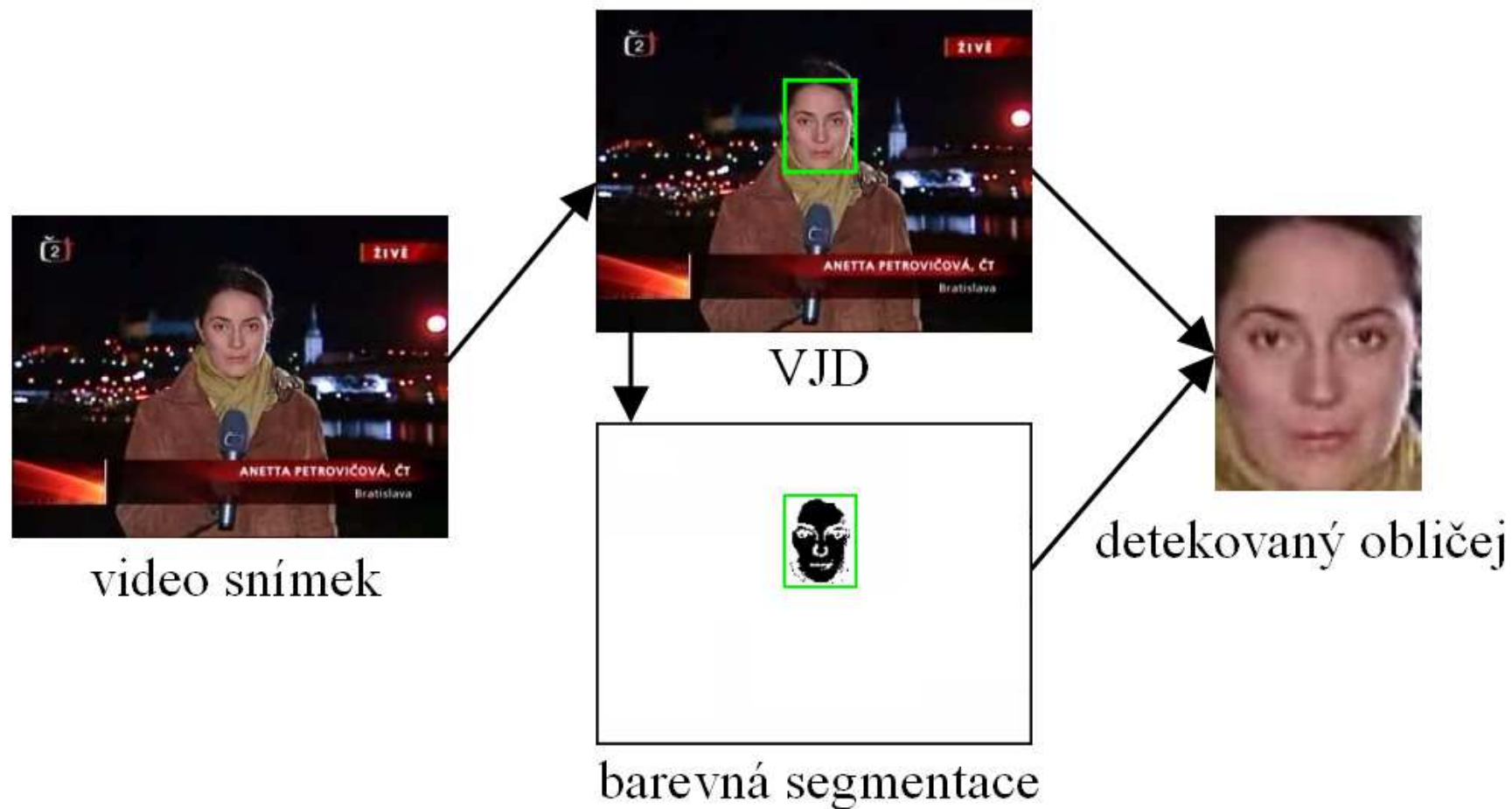
# AdaBoost a “Haar-like” příznaky

- Trénovací data:
- 5000 obličejů o rozměrech 24x24
- Trénování je výpočetně náročné
- V detekční fázi velmi rychlé,  
>200 FPS pro 640x480.

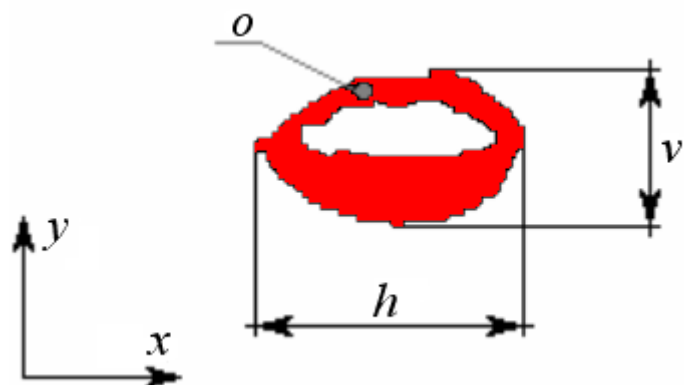


[Viola, Jones \(2001\): Robust Real-time Object Detection](#)

# AdaBoost a “Haar-like” příznaky



# Využití dynamiky chování rozpoznávaných objektů



horizontální rozšíření rtů  $h$ :

$$h = \max_{y=0..N-1} \sum_{x=0}^{M-1} f(x, y)$$

Vertikální rozšíření rtů  $v$ :

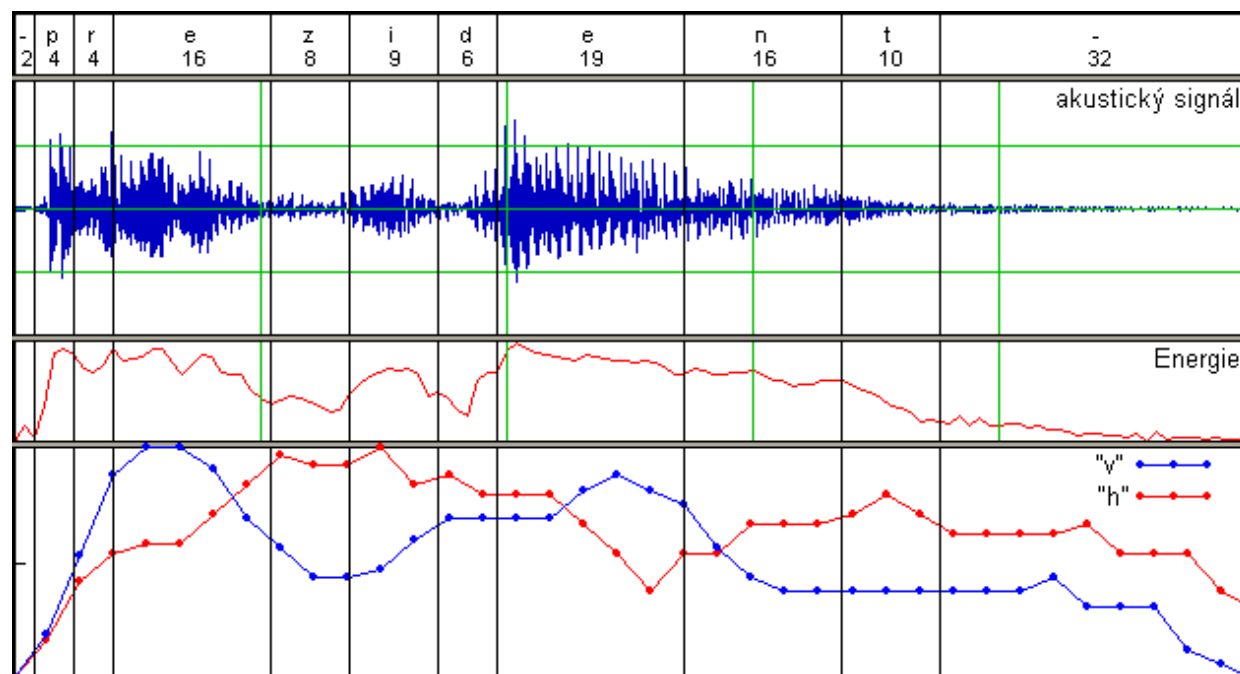
$$v = \max_{x=0..M-1} \sum_{y=0}^{N-1} f(x, y)$$

oblast rtů  $o$ :

$$o = \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x, y)$$

zaokrouhlení rtů  $r$ :

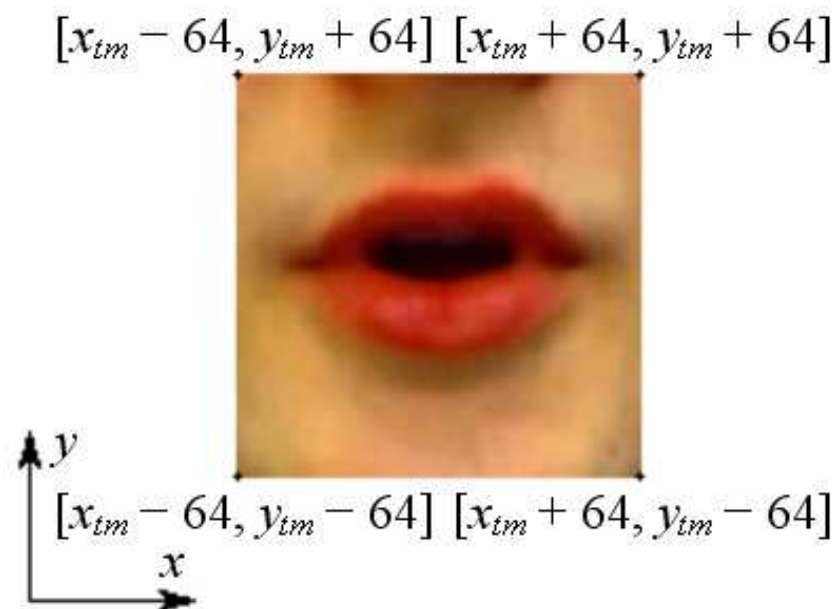
$$r = \frac{v}{h}$$



# Využití dynamiky chování rozpoznaných objektů

## 2D Diskrétní kosinová transformace:

$$F(u,v) = \frac{2c(u)c(v)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f(m,n) \cos\left(\frac{2m+1}{2N}u\pi\right) \cos\left(\frac{2n+1}{2N}v\pi\right) \quad c(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{pro } k=0 \\ 1 & \text{pro } k>0 \end{cases}$$

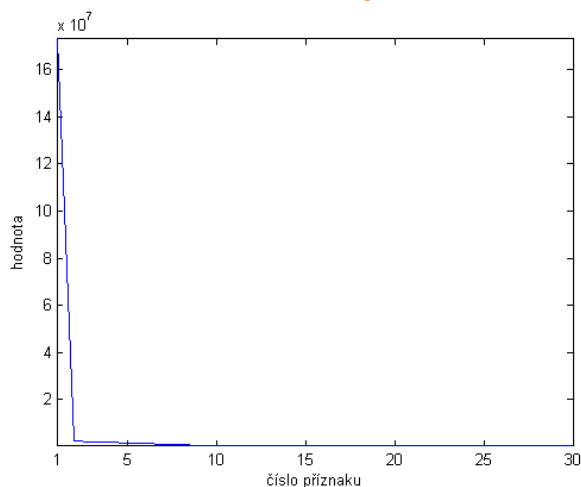


N nejvyšších hodnot energie:  $E(u,v) = F(u,v)^2$

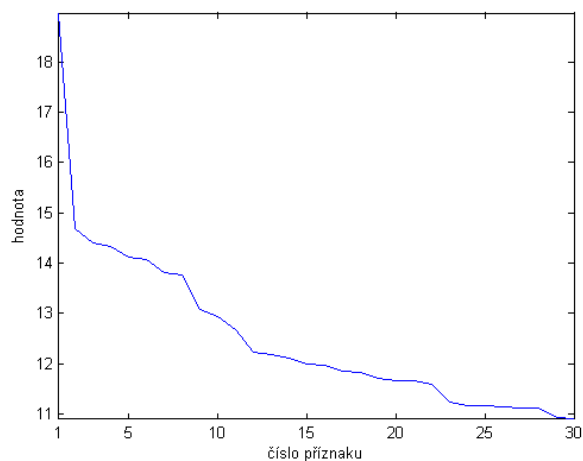


# Využití dynamiky chování rozpoznaných objektů

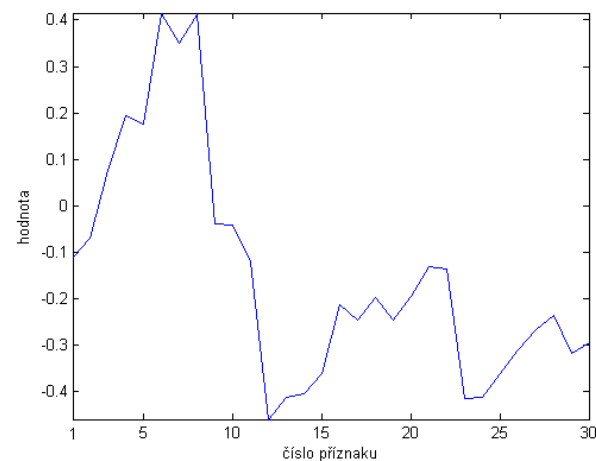
## Normalizace příznakového vektoru:



$E(u,v)$



$\log(E(u,v))$

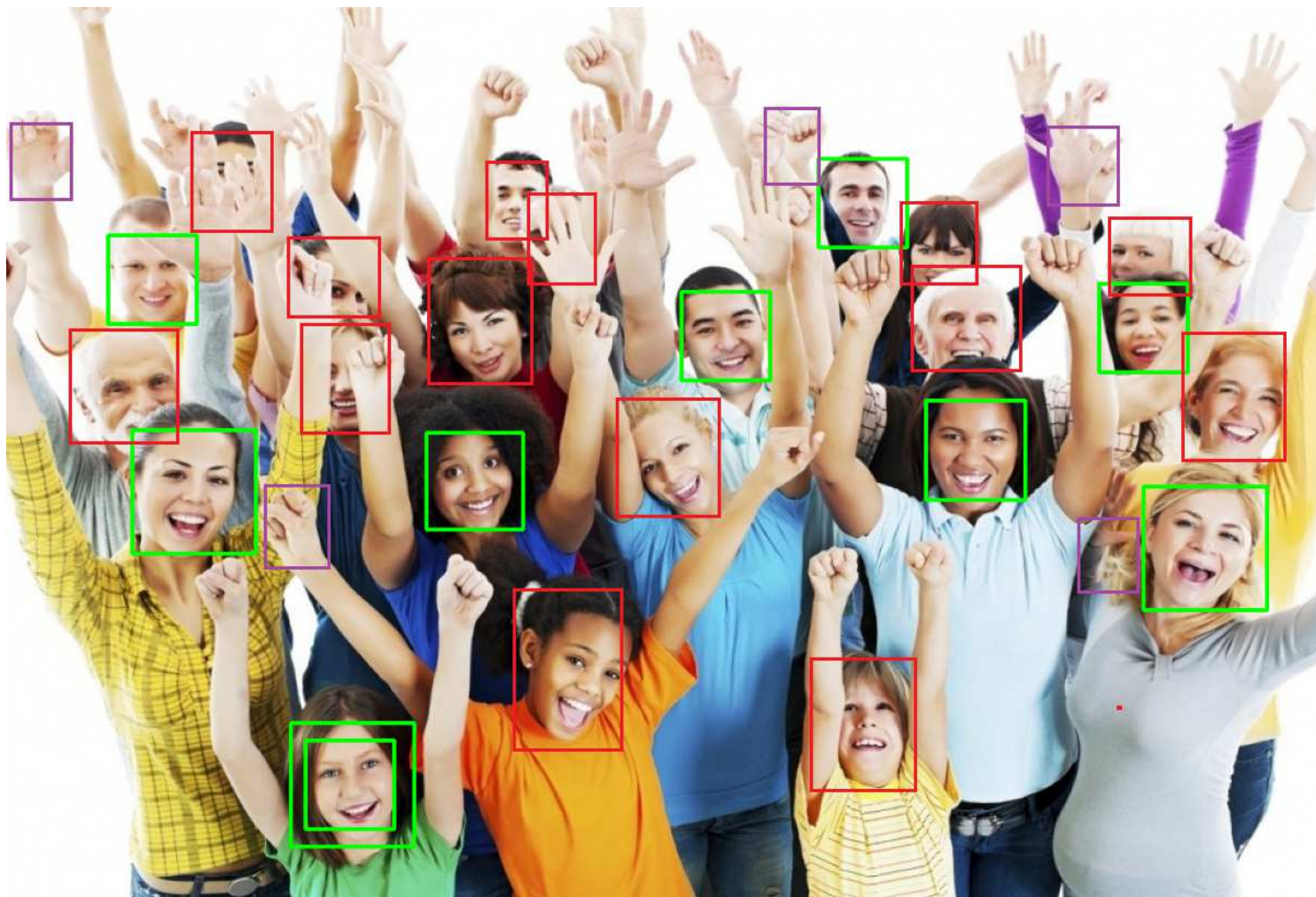


$\log(E(u,v)) - FMS$

## Výpočet dynamických příznaků:

$$\begin{aligned} x'[n] &= x[n] - x[n-1] &<<< \text{rychlost} \\ x''[n] &= x'[n] - x'[n-1] &<<< \text{zrychlení} \end{aligned}$$

# Vyhodnocení výsledků rozpoznávání



$N = 23$ ; YES = 9; NO = 14; MULTIPLE = 1; ADD = 4

$ACC = YES / N = (9 / 23) * 100 = 39\%$

# Vyhodnocení výsledků rozpoznávání

## Accuracy, Precision, Recall, F1

		Klasifikátor	
		Positive	Negative
Realita	Positive	True P.	False P.
	Negative	False N.	True N.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} = 8 / 20 = 0,4$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = 4 / 5 = 0,8$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} = 4 / 6 = 0,67$$

		Klasifikátor	
		jablko	hruška
Realita	jablko	   	
	hruška	 	  

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \cdot 0,8 \cdot 0,67 / (0,8 + 0,67) = 0,73$$

Accuracy - Kolik klasifikací bylo pravdivých

Precision - Kolik pozitivních klasifikací bylo pravdivých

Recall - Jak dobře klasifikátor dokáže najít všechny pozitivní předpovědi.

F1 – Jak je klasifikátor efektivní. Precision nebo Recall >>> 0, F1 >>> 0

# Vyhodnocení výsledků rozpoznávání

Accuracy, Precision, Recall, F1



TP = 4; TN = 0; FP = 0; FN = 0  
Acc = 1; P = 1; R = 1; F1 = 1

		Klasifikátor	
		Positive	Negative
Realita	Positive	True P.	False P.
	Negative	False N.	True N.

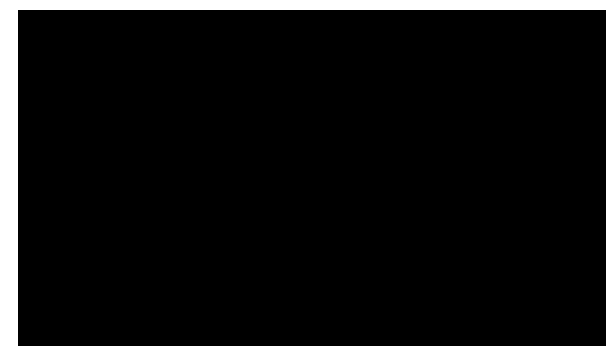
		Klasifikátor	
		Obličej	No
Realita	Obličej	True P.	False P.
	No	False N.	True N.



TP = 4; TN = 0; FP = 0; FN = 1  
Acc = 0,8; P = 1; R = 0,8; F1 = 0,89



TP = 0; TN = 0; FP = 4; FN = 0  
Acc = 0; P = 0; R = 0; F1 = 0



TP = 0; TN = 1; FP = 0; FN = 0  
Acc = 1; P = 0; R = 0; F1 = 0

# Vyhodnocení výsledků rozpoznávání



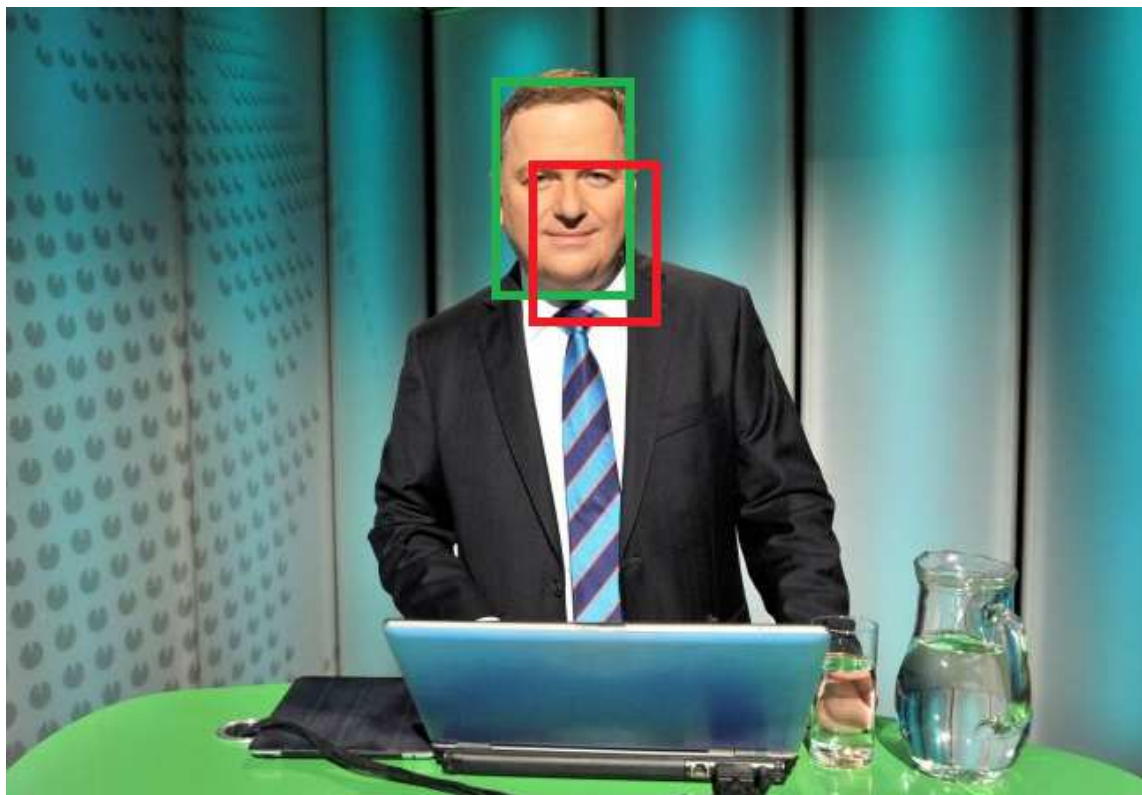
OK!




???



# Vyhodnocení výsledků rozpoznávání

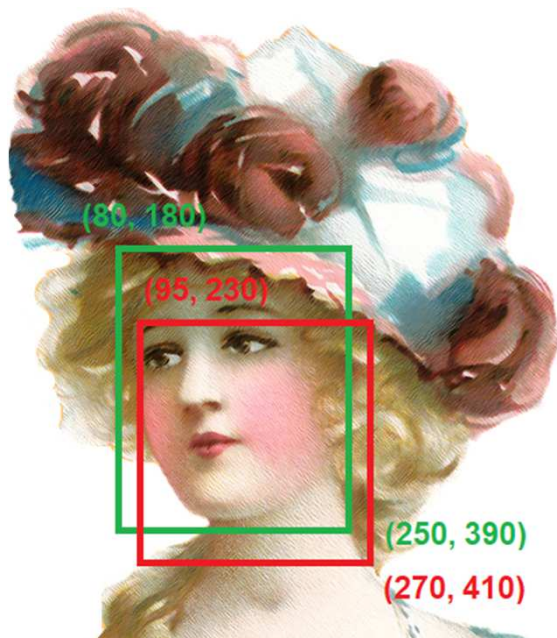
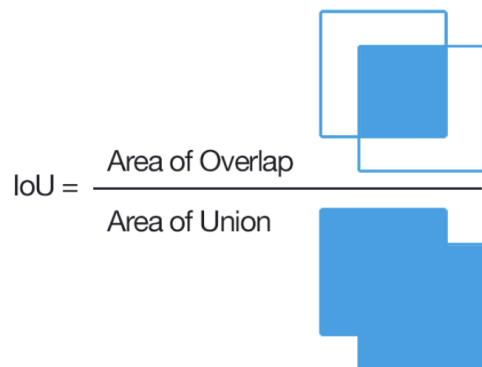


Intersection over Union (IoU)

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$
The diagram illustrates the IoU calculation with two overlapping squares. The top part shows the two squares with their intersection shaded in blue. The bottom part shows the union of the two squares, which is the combined area of both, also shaded in blue.

# Vyhodnocení výsledků rozpoznávání

Intersection over Union (IoU)



REF = (xf1, yf1, xf2, yf2) = (80, 180, 250, 390)

REC = (xc1, yc1, xc2, yc2) = (95, 230, 270, 410)

1)  $X1 = \max(xf1, xc1) = \max(80, 95) = 95$

$Y1 = \max(yf1, yc1) = \max(180, 230) = 230$

$X2 = \min(xf2, xc2) = \min(250, 270) = 250$

$Y2 = \min(yf2, yc2) = \min(390, 410) = 390$

2) IF (X2 < X1) OR (Y2 < Y1) >>> IoU = 0

ELSE:

3)  $|REF \cap REC| = (X2 - X1) * (Y2 - Y1)$

$= (250 - 95) * (390 - 230) = 24800$

4)  $|REF| = (xf2 - xf1) * (yf2 - yf1)$

$= (250 - 80) * (390 - 180) = 35700$

$|REC| = (xc2 - xc1) * (yc2 - yc1)$

$= (270 - 95) * (410 - 230) = 31500$

5)  $|REF \cup REC| = |REF| + |REC| - |REF \cap REC|$

$= 35700 + 31500 - 24800 = 42400$

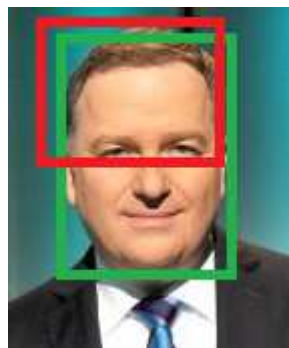
6)  $IoU = \frac{|REF \cap REC|}{|REF \cup REC|} = 24800 / 42400 = 0,58$



# Vyhodnocení výsledků rozpoznávání



IoU = 0,8



IoU = 0,4

T = 0,5 IF IoU < T >>> FP  
ELSE >>> TP

mAP@IoU=0,5

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

**mean** je průměr přes jednotlivé AP  
pro různé T u IoU

COCO challenge: 10x IoU 0.5:0.05:0.95

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^N \text{AP}_i$$

mean Average Precision (mAP)

**Average Precision** např. pro IoU 0,5 se vypočítá jako plocha pod křivkou Precision – Recall, kde měníme práh spolehlivosti (související s IoU) od 0 do 1

Precision - Recall curve

