

Metode za brojanje ljudi u slici



Luka Novak

Sadržaj

- Uvod
- Računalni vid
- Detekcija objekata prije konvolucijskih modela
- Viola Jones algoritam
- Detekcija objekata konvolucijskim modelima
- YOLO
- Rezultati
- Zaključak

1.

Računalni vid

Vidjeti, prepoznati, razumjeti

- ▷ Čovjek prima 2/3 informacija osjetilom vida
- ▷ 13 milisekundi potrebno da prepozna što je na slici
- ▷ 3 milisekunde da čovjek prepozna majku

Kategorije

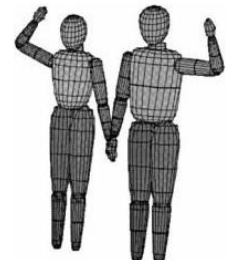
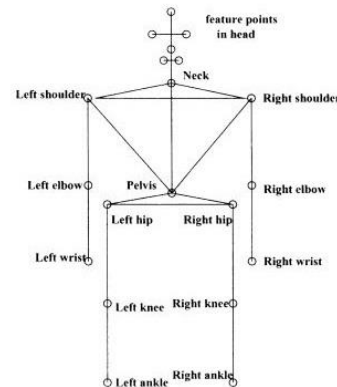
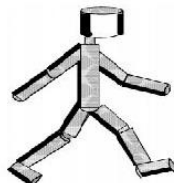
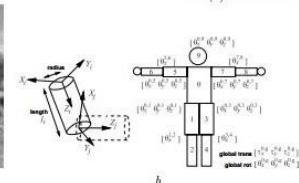
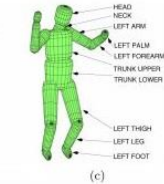
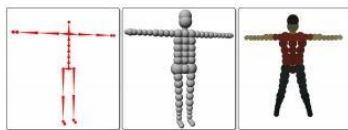
- ▷ Klasifikacija
- ▷ Lokalizacija
- ▷ Detekcija
- ▷ Praćenje
- ▷ Segmentacija
- ▷ Super-resolution
- ▷ Style transfer
- ▷ Colourisation

2.

Detekcija objekata prije konvolucijskih modela

Pristupi kroz povijest

- ▷ Poravnanje
- ▷ Modeli temeljeni na izgledu
- ▷ Klizeći prozor
- ▷ Lokalne značajke
- ▷ Parts-and-shapes modeli



3.

Viola Jones algoritam

Algoritam

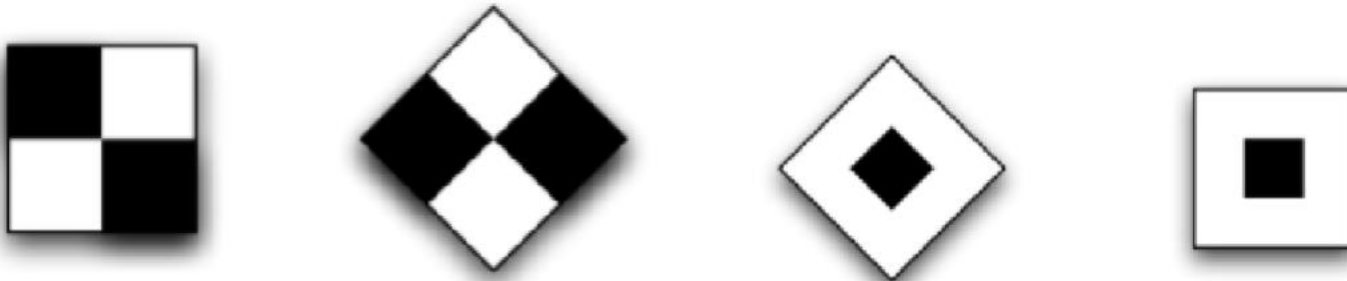
- ▷ Paul Viola i Michael Jones
- ▷ *Rapid object detection using a boosted cascade of simple features, 2001*
- ▷ Temeljen na klizećem prozoru
- ▷ Koristi kaskade Haarovih značajki

Haarove značajke



(a) Značajke rubova

(b) Linijske značajke



(c) Dijagonalne značajke

(d) Radijalne značajke

Izračun na razini piksela

$$\Delta = dark - white = \frac{1}{n} \sum^n I_{dark}(x) - \frac{1}{n} \sum^n I_{light}(x)$$

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

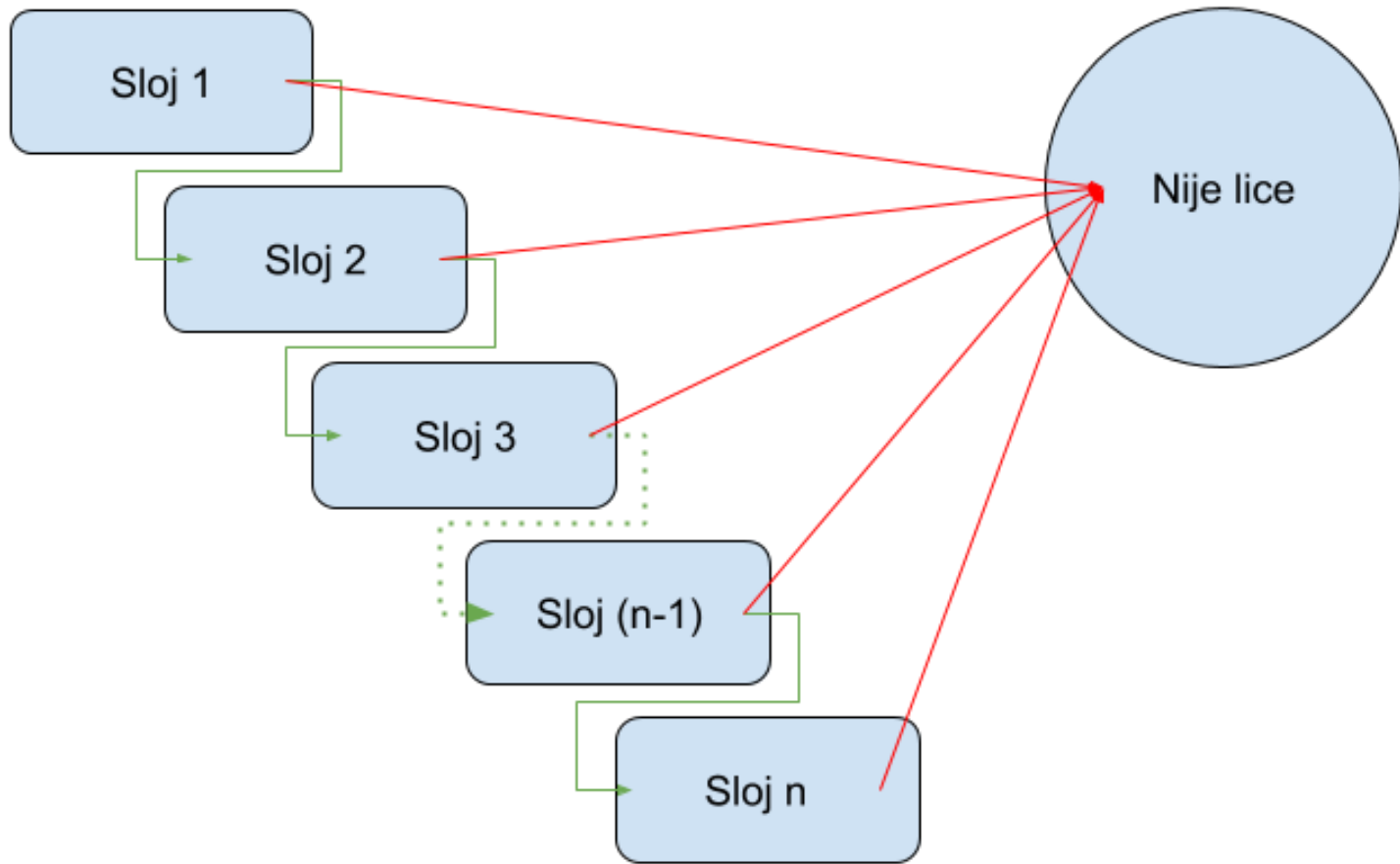
(a) Idealne vrijednosti piksela

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

(b) Stvarne vrijednosti piksela

Slika 3.2: Izračun za linijske Haarove značajke

Haarove kaskade



Postavke algoritma

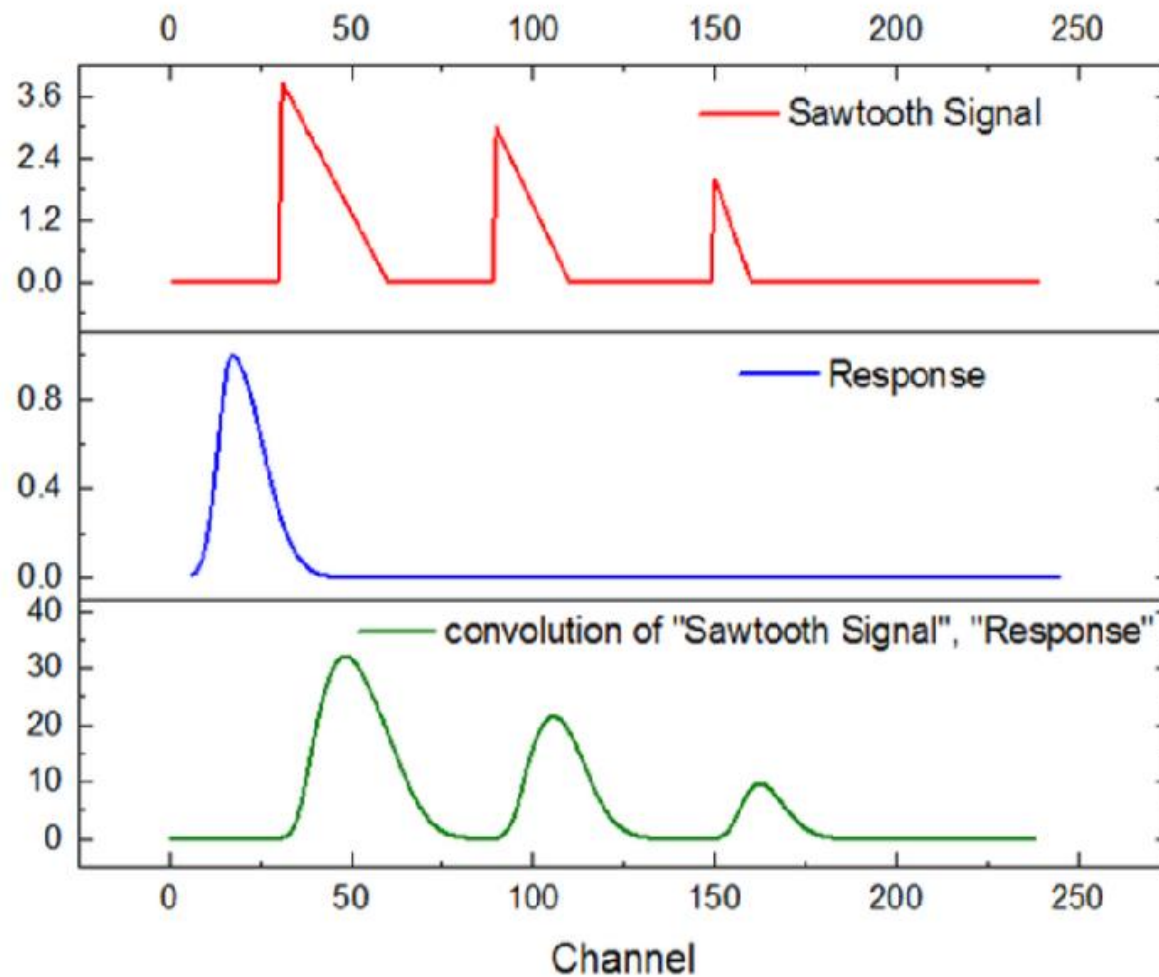
- ▷ 160000 značajki
- ▷ Neke odgovaraju više ovisno o domeni
- ▷ 200 značajki za točnost od 95%
- ▷ Konačne postavke → 6000 parametara
- ▷ U prosjeku, oko 10 značajki evaluirano po slici



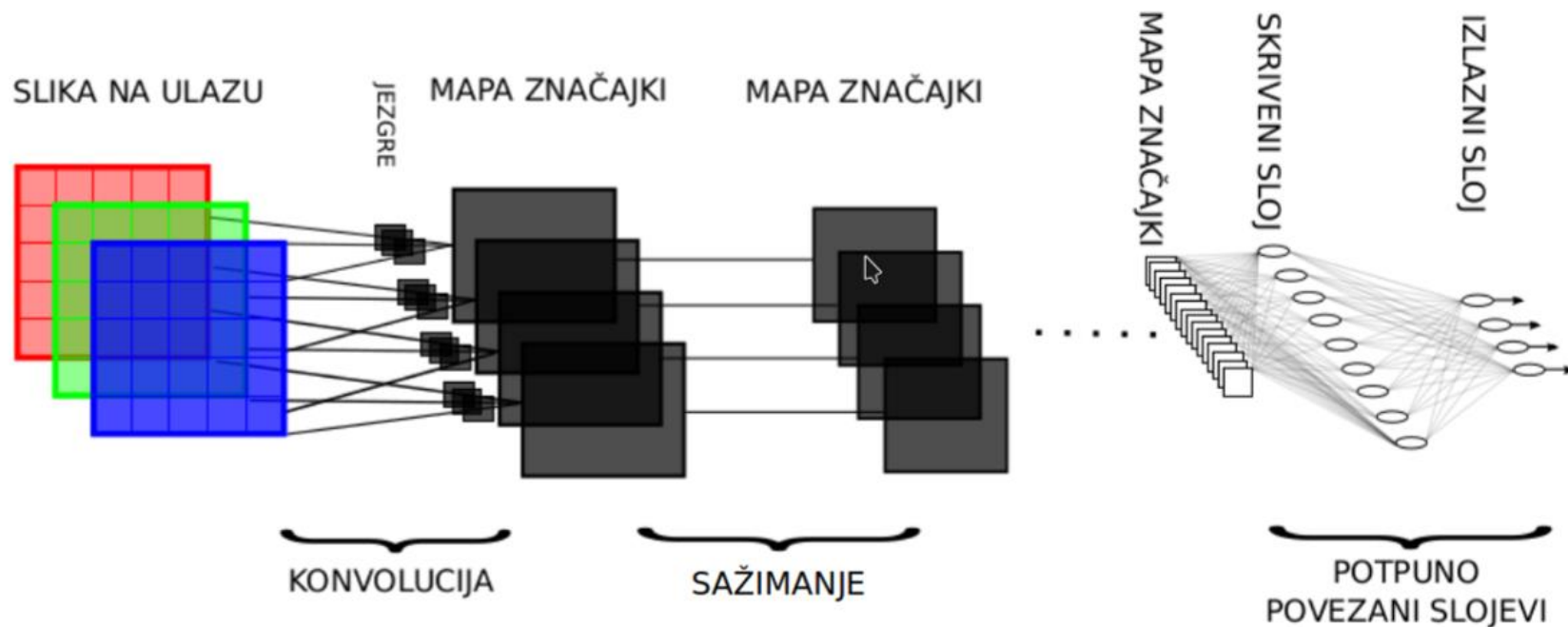
4.

Detekcija objekata konvolucijskim modelima

Konvolucija



Struktura konvolucijske neuronske mreže



5. YOLO

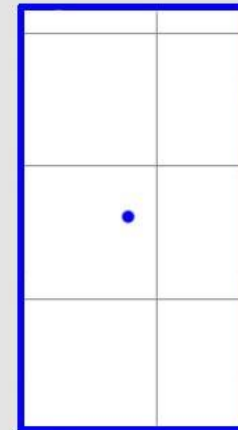
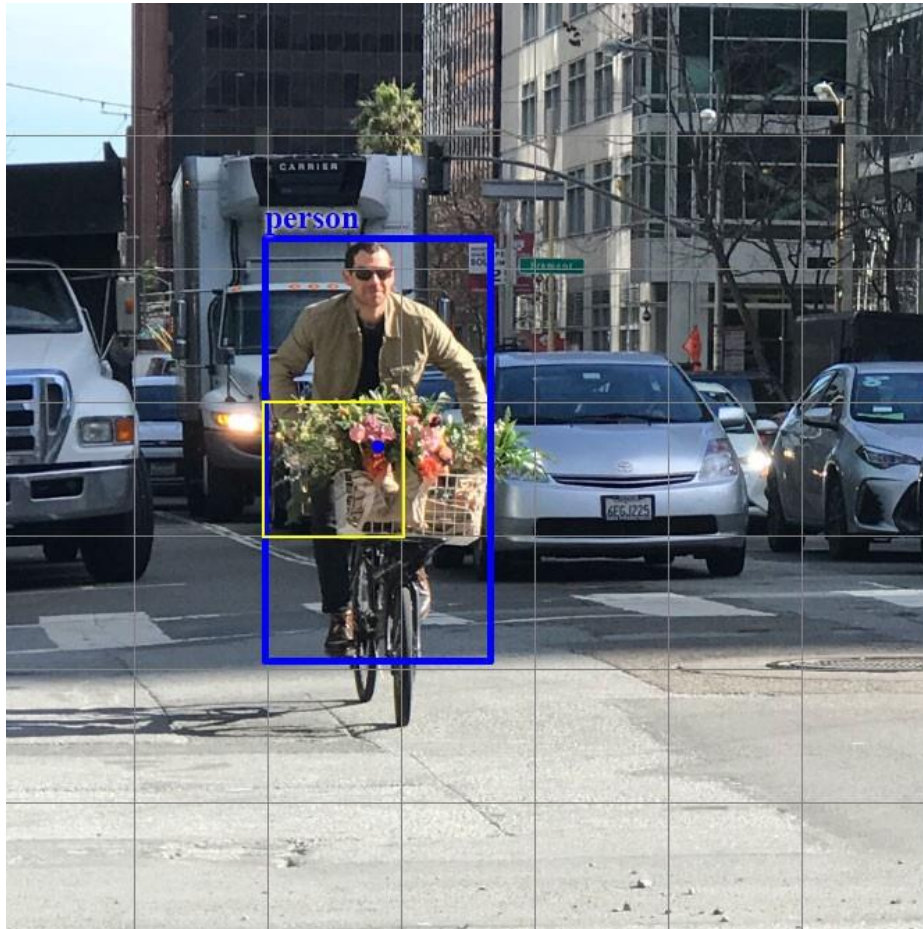
You Only Look Once

- ▷ Redmon et al., 2015
- ▷ Full-YOLO – veća točnost i preciznost
- ▷ Tiny-YOLO – bolje performanse
- ▷ Jedan prolaz kroz mrežu
- ▷ Lokalizacija i klasifikacija u jednoj mreži

Podjela slike na ćelije

- ▷ **CxC** ćelija
- ▷ Za svaku ćeliju algoritam predviđa:
 - **B** okvira i pouzdanost za svaki od njih
 - Samo 1 objekt
 - **P** vjerojatnosti – po jednu za svaku klasu **K**
- ▷ U implementaciji – 13x13 ćelija, 5 okvira predviđenih po ćeliji

Podjela slike na ćelije



Arhitektura mreže

Ime sloja	Filteri	Korak	Dimenzije na ulazu	Dimenzije na izlazu
Conv1	3 x 3 x 32	1 x 1	416 x 416 x 3	416 x 416 x 32
MaxPool1	2 x 2	2 x 2	416 x 416 x 32	208 x 208 x 32
Conv2	3 x 3 x 64	1 x 1	208 x 208 x 32	208 x 208 x 64
MaxPool2	2 x 2	2 x 2	208 x 208 x 64	104 x 104 x 64
Conv3	3 x 3 x 128	1 x 1	104 x 104 x 64	104 x 104 x 128
Conv4	1 x 1 x 64	1 x 1	104 x 104 x 128	104 x 104 x 64
Conv5	3 x 3 x 128	1 x 1	104 x 104 x 64	104 x 104 x 128
MaxPool2	2 x 2	2 x 2	104 x 104 x 128	52 x 52 x 128
Conv6	3 x 3 x 256	1 x 1	52 x 52 x 128	52 x 52 x 256
Conv7	1 x 1 x 128	1 x 1	52 x 52 x 256	52 x 52 x 128
Conv8	3 x 3 x 256	1 x 1	52 x 52 x 128	52 x 52 x 256
MaxPool3	2 x 2	2 x 2	52 x 52 x 256	26 x 26 x 256
Conv9	3 x 3 x 512	1 x 1	26 x 26 x 256	26 x 26 x 512
Conv10	1 x 1 x 256	1 x 1	26 x 26 x 512	26 x 26 x 256
Conv11	3 x 3 x 512	1 x 1	26 x 26 x 256	26 x 26 x 512
Conv12	1 x 1 x 256	1 x 1	26 x 26 x 512	26 x 26 x 256
Conv13	3 x 3 x 512	1 x 1	26 x 26 x 256	26 x 26 x 512
MaxPool4	2 x 2	2 x 2	26 x 26 x 512	13 x 13 x 512
Conv14	3 x 3 x 1024	1 x 1	13 x 13 x 512	13 x 13 x 1024
Conv15	1 x 1 x 512	1 x 1	13 x 13 x 1024	13 x 13 x 512
Conv16	3 x 3 x 1024	1 x 1	13 x 13 x 512	13 x 13 x 1024
Conv17	1 x 1 x 512	1 x 1	13 x 13 x 1024	13 x 13 x 512
Conv18	3 x 3 x 1024	1 x 1	13 x 13 x 512	13 x 13 x 1024
Conv19	3 x 3 x 1024	1 x 1	13 x 13 x 1024	13 x 13 x 1024
Conv20	3 x 3 x 1024	1 x 1	13 x 13 x 1024	13 x 13 x 1024
Route	from	Conv13		
Conv21	1 x 1 x 64	1 x 1	26 x 26 x 512	26 x 26 x 64
Space2Depth	reshape			
Concatenate	Route from Conv20	and	Space2Depth	
Conv22	3 x 3 x 1024	1 x 1	13 x 13 x 1280	13 x 13 x 1024
Conv23	1 x 1 x 425	1 x 1	13 x 13 x 1024	13 x 13 x 425

Ulaz i izlaz iz mreže

▷ Ulaz: $416 \times 416 \times 3$

▷ Izlaz: $13 \times 13 \times 425$

- Okviri: $(13 \times 13, 5, 4)$ raspoređenih kao $(x_{\text{top}}, y_{\text{top}}, x_{\text{bottom}}, y_{\text{bottom}})$
- Pouzdanost postojanja objekta u okviru: $(13 \times 13, 5, 1)$
- Vjerojatnost klasa za okvire kao vektor $(13 \times 13, 5, 80)$

Pogreška klasifikacije

$$\sum_{i=1}^{S \times S} 1_i^{obj} \sum_{c \in klase} (p_i(c) - \hat{p}_i(c))^2$$

gdje

- $1_i^{obj} = 1$ ako se objekt pojavljuje u čeliji i , inače 0
- $\hat{p}_i(c)$ označava uvjetnu vjerojatnost klase c u čeliji i

Pogreška lokalizacije

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] +$$
$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

gdje

- $1_{ij}^{obj} = 1$ ako se objekt nalazi u j -tom okviru i -te ćelije, inače 0
- λ_{coord} parametar za ugađanje težine za pogrešku u koordinatama okvira

Pogreška pouzdanosti

1. Objekt je detektiran u okviru

$$\sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)^2$$

gdje

- $1_{ij}^{obj} = 1$ ako se objekt nalazi u j -tom okviru i -te ćelije, inače 0
- \hat{C}_i jest iznos pouzdanosti za okvir j u ćeliji i

Pogreška pouzdanosti

2. Objekt nije detektiran u okviru

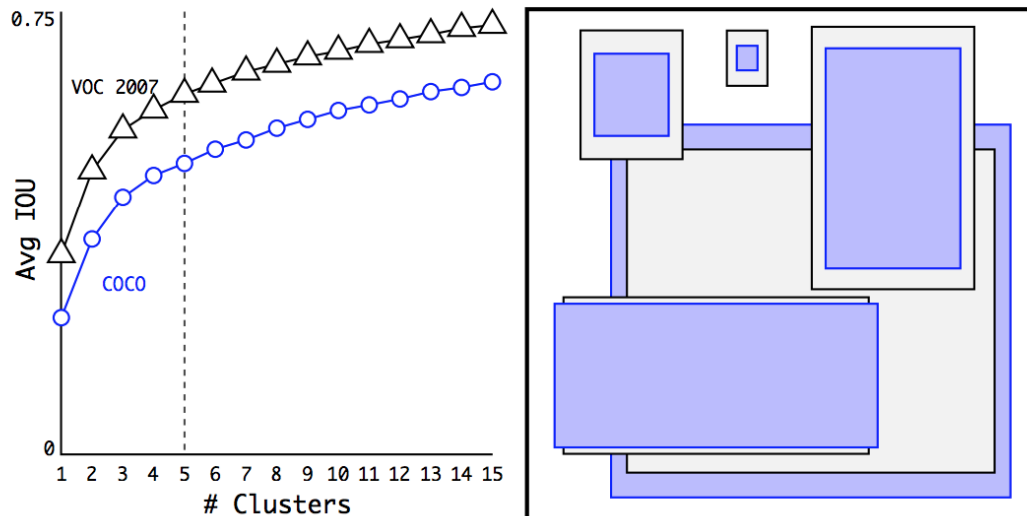
$$\lambda_{no-obj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{no-obj} (C_i - \hat{C}_i)^2$$

gdje

- 1_{ij}^{no-obj} jest komplement gornjeg izraza 1_{ij}^{obj}
- \hat{C}_i jest iznos pouzdanosti za okvir j u čeliji i
- λ_{no-obj} je faktor kojim se smanjuje iznos gubitka pri detekciji pozadine

Anchor boxes

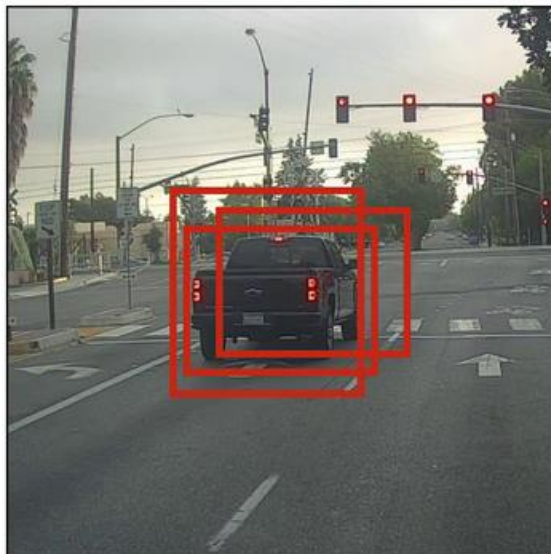
- ▷ **n** najčesčih okvira
- ▷ K-means clustering, $k=5$
- ▷ iz skupova COCO i VOC2007



Predviđanje

- ▷ Non-Maximal Suppression za odabiranje okvira
- ▷ Koristi Intersection over Union

Before non-max suppression



Non-Max
Suppression



After non-max suppression



6.

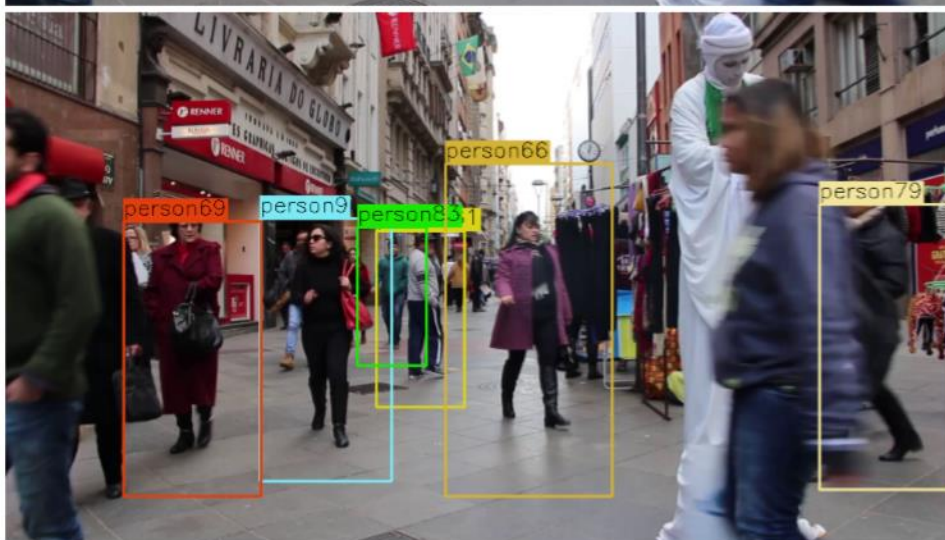
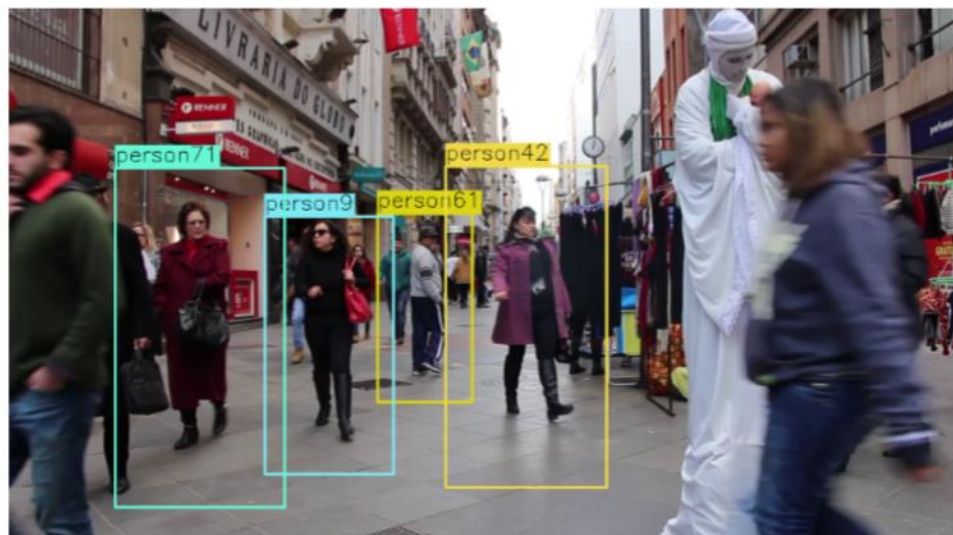
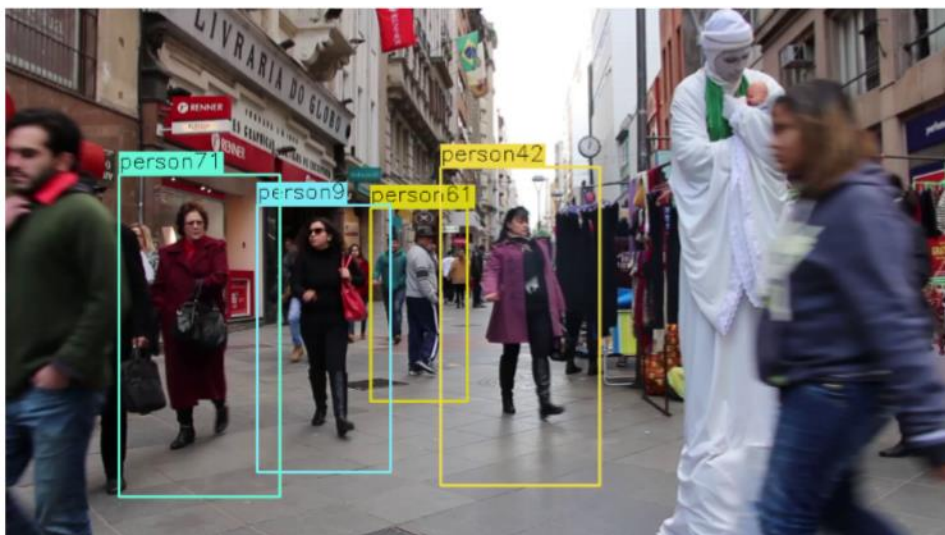
Rezultati

Viola Jones algoritam

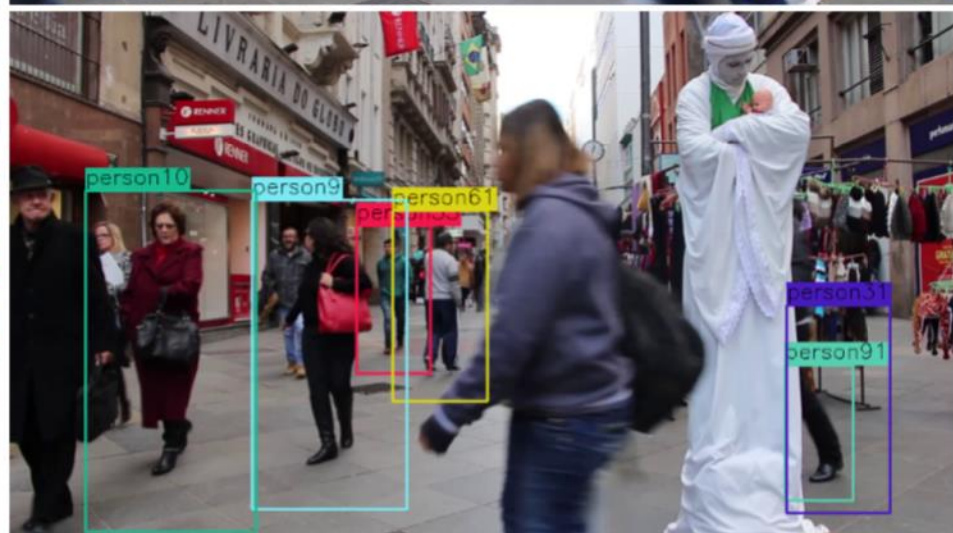
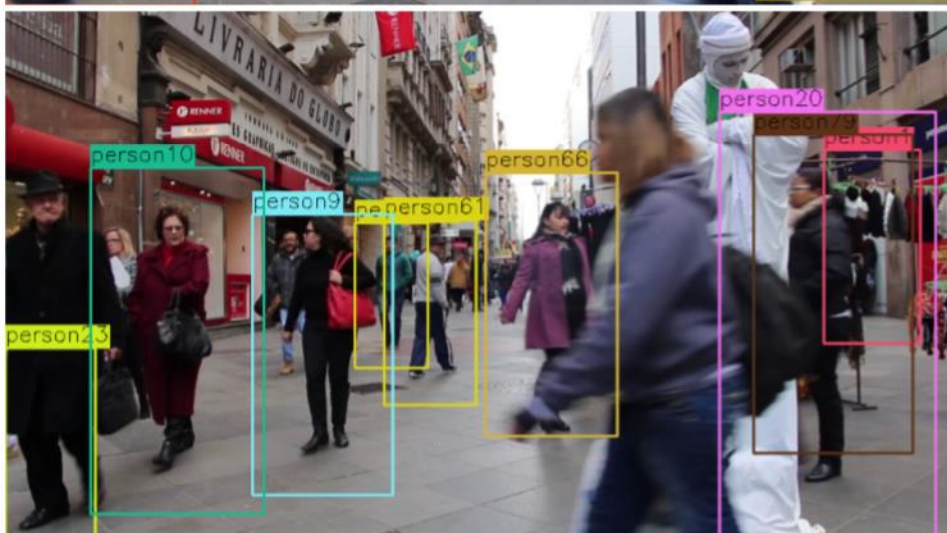
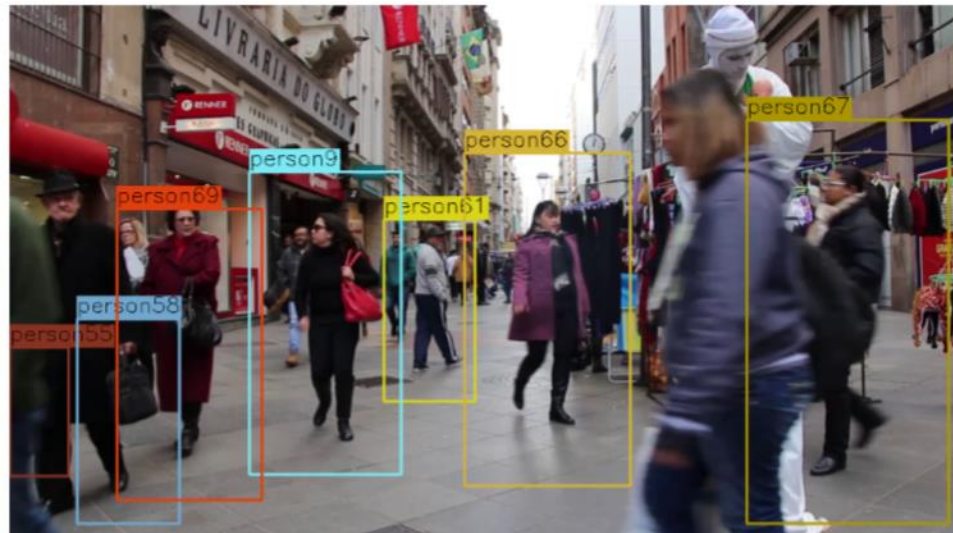
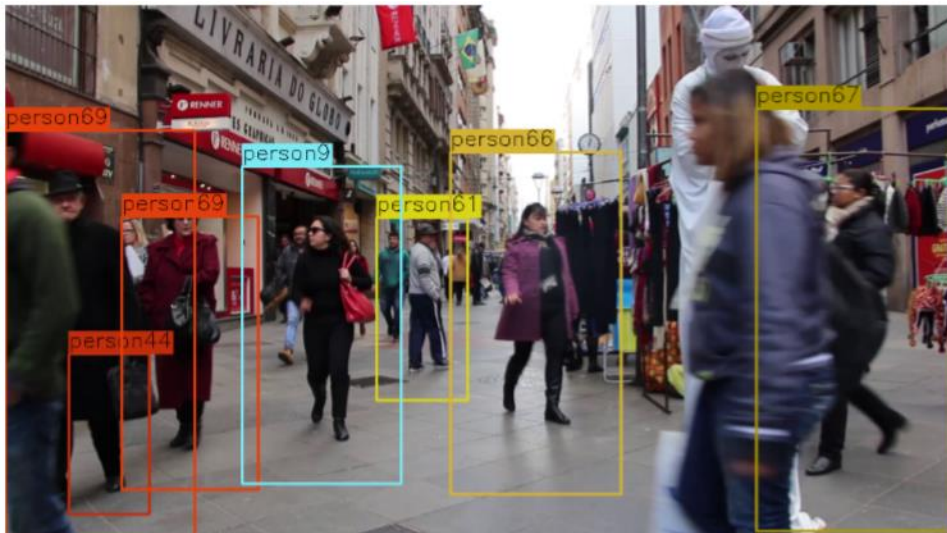
- ▷ Vrlo brz algoritam
- ▷ U većini slučajeva dovoljno točan
- ▷ Puzdan i jednostavan za implementaciju – *OpenCV*

Broj vidljivih osoba po slici	5.67
Broj detekcija po slici videa	4.91
True Positivea po slici	3.59
False Positivea po slici	1.32
Preciznost (precision)	73.11%

Detekcija



False positives



YOLO

- ▷ U izvornom obliku implementacija je u C-u
- ▷ Brzine 45 do 90 FPS
- ▷ Implementacija – Python + Keras
- ▷ Brzina ~ 60-75 FPS

Obrada slike	3.9219 s
Stvaranje modela	2.717 s
Predviđanje sa slike	1.201 s
Postprocesiranje	0.0003 s
Iscrtavanje okvira	0.0002 s

Detekcija

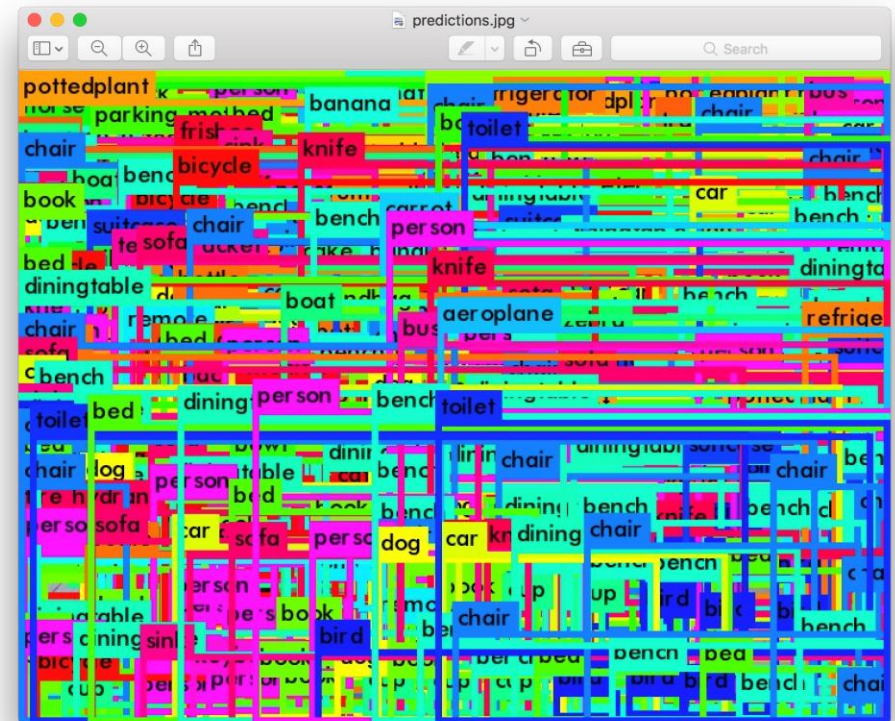


- ▷ Puno manje false positivea nego kod Viola Jones algoritma
- ▷ Puno sporiji na CPU

Broj vidljivih osoba po slici	5.67
Broj detekcija po slici videa	5.16
True Positivea po slici	4.53
False Positivea po slici	0.32
Preciznost (precision)	87.79%

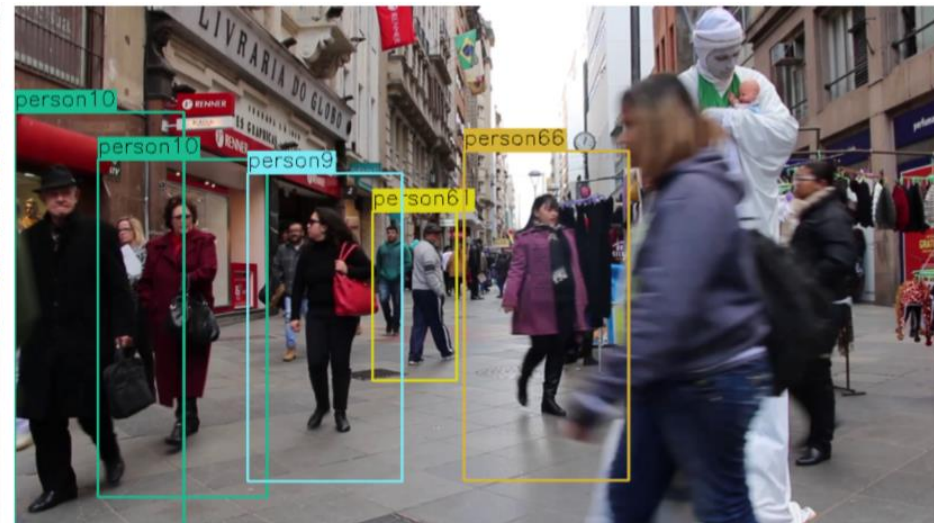
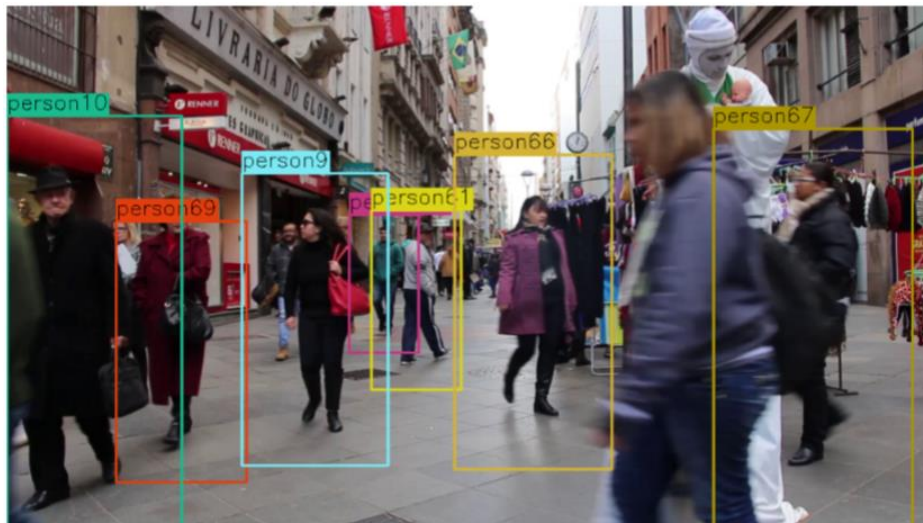
Dodane opcije za detekciju

- ▷ Opcija za preskakanje frameova
- ▷ Non-Maximal Suppression prag za crtanje okvira



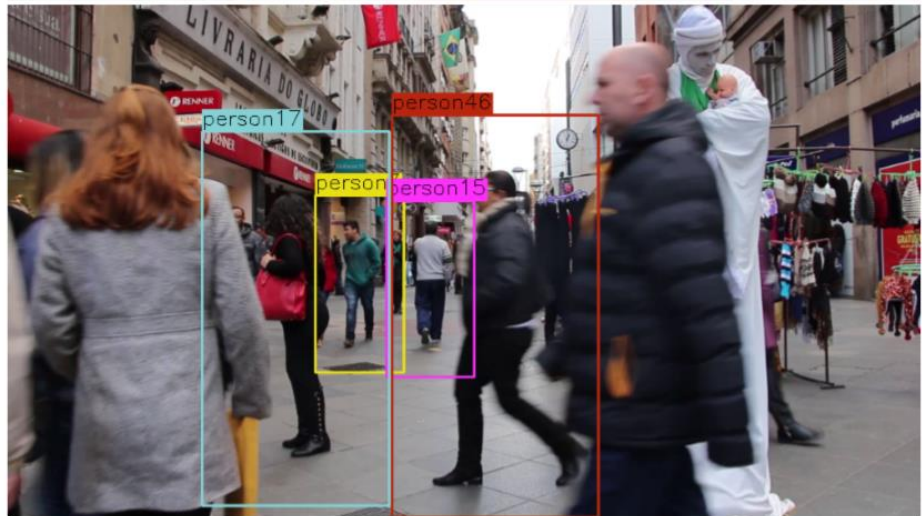
Praćenje objekata

- ▷ Intersection over Union
- ▷ Između 2 vremenska trenutka
- ▷ Usporedba okvira



Problem kretanja uz kameru

- ▷ Postavljen prag na veličinu okvira



7.

Zaključak

- ▷ Oba algoritma zadovoljavajuća
- ▷ Viola Jones
 - Brži
 - Više false positivea
- ▷ YOLO
 - Sporiji
 - Točniji
 - Za uspješno praćenje u stvarnom vremenu, potrebna je GPU
- ▷ Odabir ovisi o danom problemu i resursima

Pitanja



Hvala na pažnji