Învățare Automată în Arta Vizuală

Curs 8: Detecția Obiectelor

Localizarea Obiectelor vs. Detecție

Clasificare

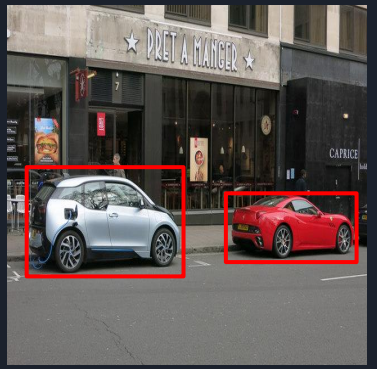
‘mașină’

Clasificare cu Localizare

‘mașină’

1 obiect (clasă) / imagine

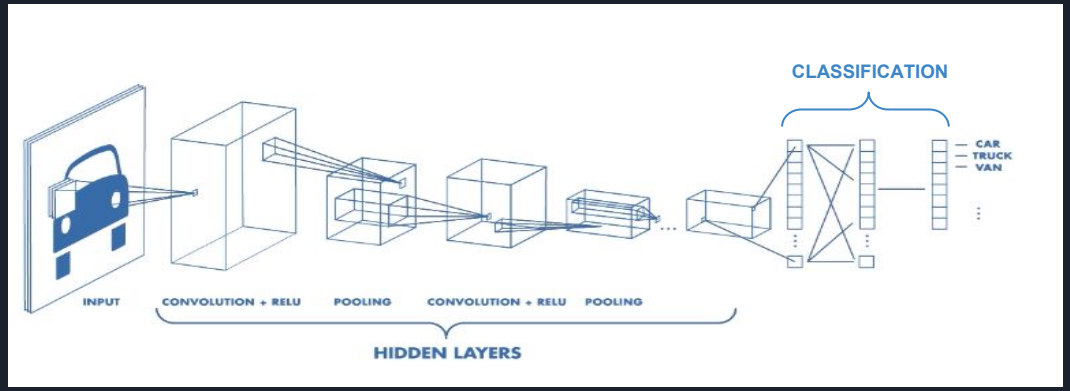
Detecție



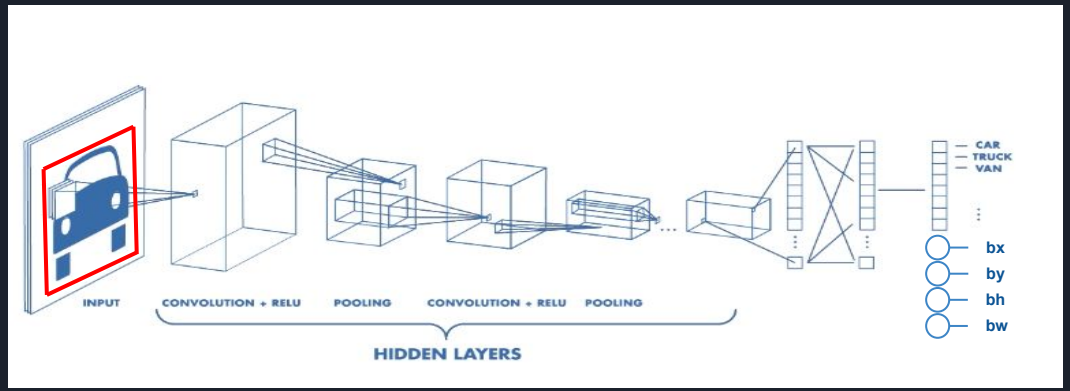
multiple obiecte (clase) / imagine

Clasificare cu Localizare

- Clasificare:



- Clasificare cu Localizare



Definirea etichetei (label) y

- 4 clase: ***pieton***, ***mașină***, ***bicicletă***, ***fundal*** **(background - niciuna din clasele de mai sus)**
- trebuie să obținem ***bx, by, bh, bw*** și etichetele claselor (1-4)
- y = $\begin{bmatrix} pc \\ bx \\ by \\ bh \\ bw \\ c1 \\ c2 \\ c3 \end{bmatrix}$, pc - obiect/fundal

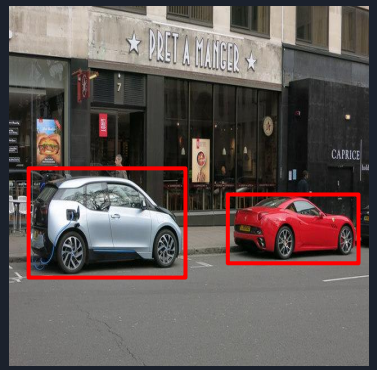
$\begin{bmatrix} 1 \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \\ 0 \end{bmatrix}$

$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$

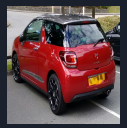
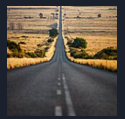
- exemplu funcție loss:
  - $y_1 = 1$ (pc = 1): $\mathbf{L(\hat{y}, y) = Sum((\hat{y}_i - y_i)^2)}$
  - $y_1 = 0$ (pc = 0): $\mathbf{L(\hat{y}, y) = (\hat{y}_1 - y_1)^2}$

Detecția Obiectelor - Fereastră Glisantă (Sliding Window)

- Exemplu - Detecția Mașinilor



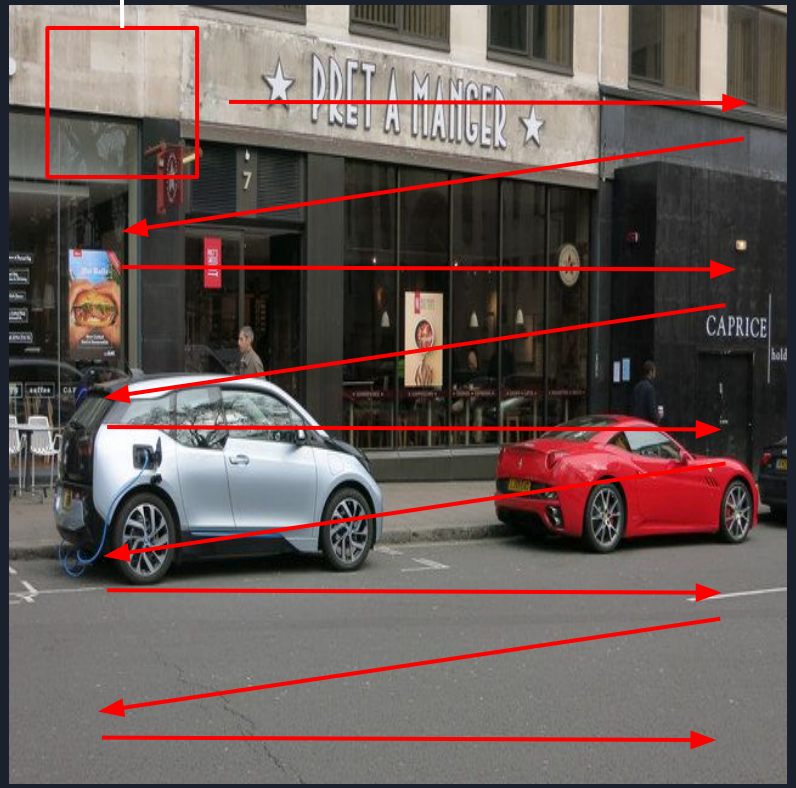
Set de antrenare:

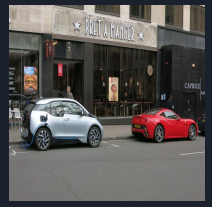
X	y
	1
	1
	1
	0

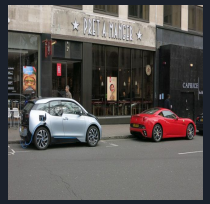
 → ConvNet → y

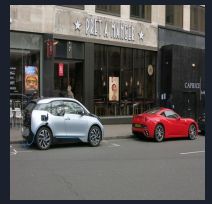
Detecția Obiectelor - Fereastră Glisantă (Sliding Window)

ConvNet → 1 / 0

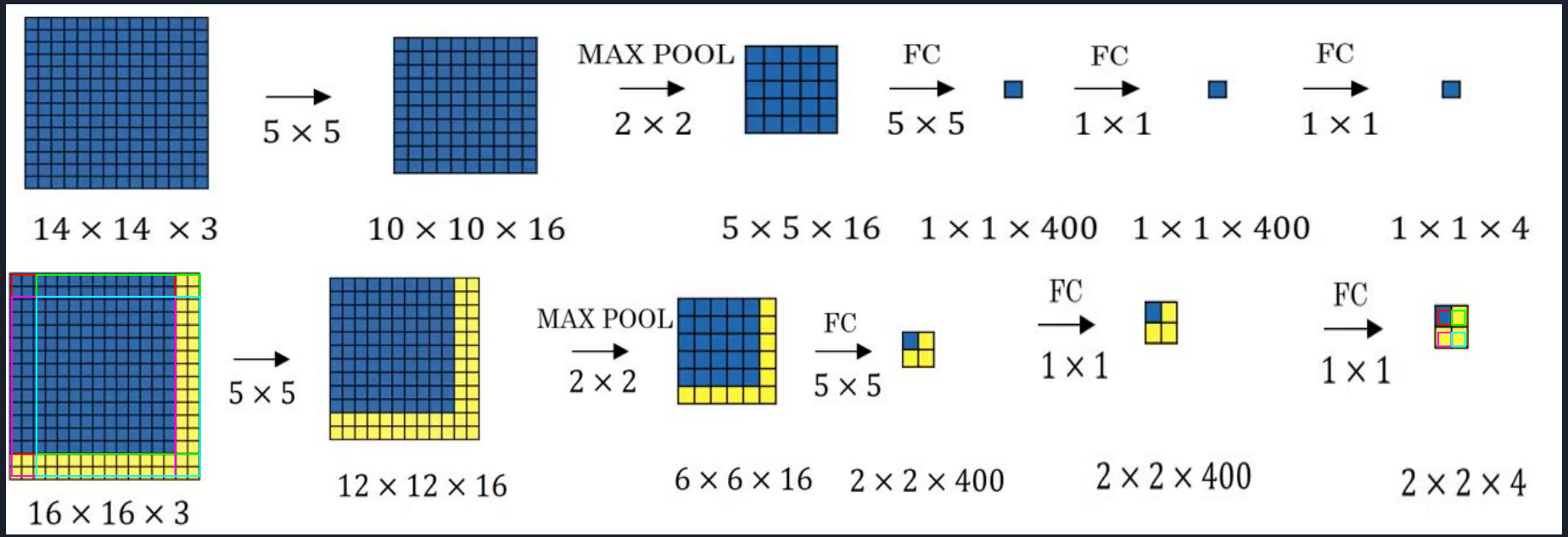








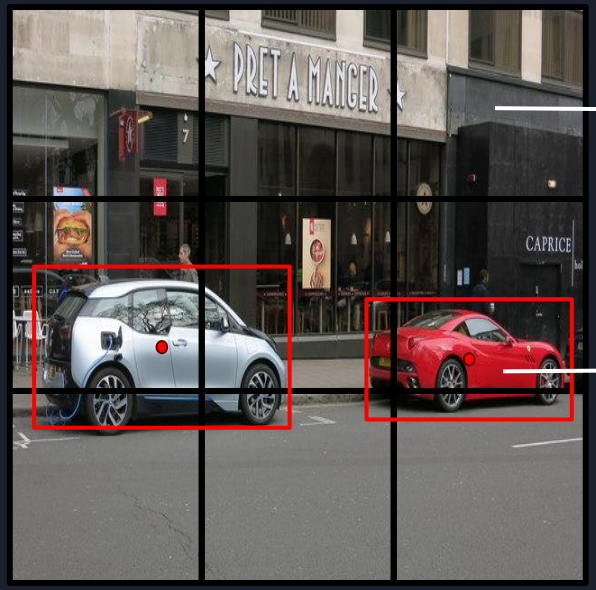
Detecția Obiectelor - Fereastră Glisantă - Implementare Convoluțională



Detecția Obiectelor - Fereastră Glisantă - Implementare Convoluțională

- Probleme:
  - chenarele (bounding boxes) prezise nu sunt precise
  - chenarele au aspecte diferite, nu sunt necesar pătrate

Detecția obiectelor - Algoritmul YOLO (You Only Look Once)

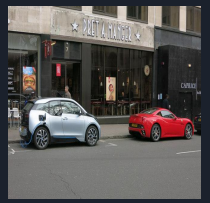


- pentru fiecare celulă, y = $\begin{bmatrix} pc \\ bx \\ by \\ bh \\ bw \\ c1 \\ c2 \\ c3 \end{bmatrix}$

$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$

$\begin{bmatrix} 1 \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \\ 0 \end{bmatrix}$

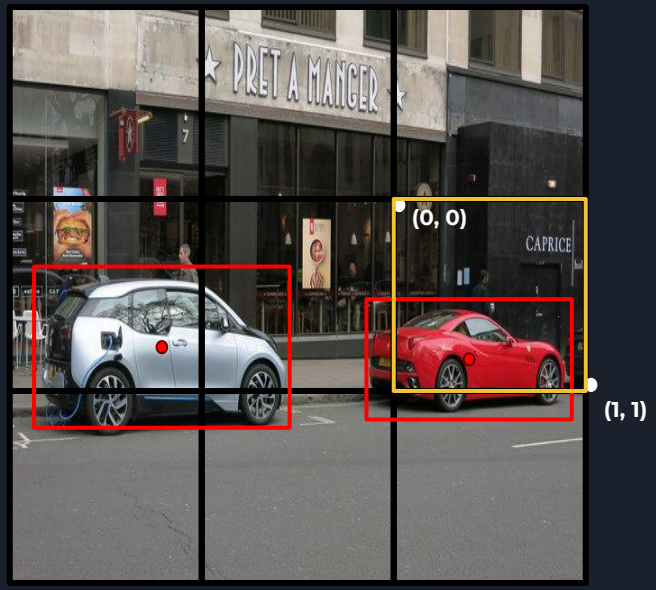
- rezultat - **volum 3x3x8**



ConvNet

3x3x8

Detecția obiectelor - Algoritmul YOLO (You Only Look Once)



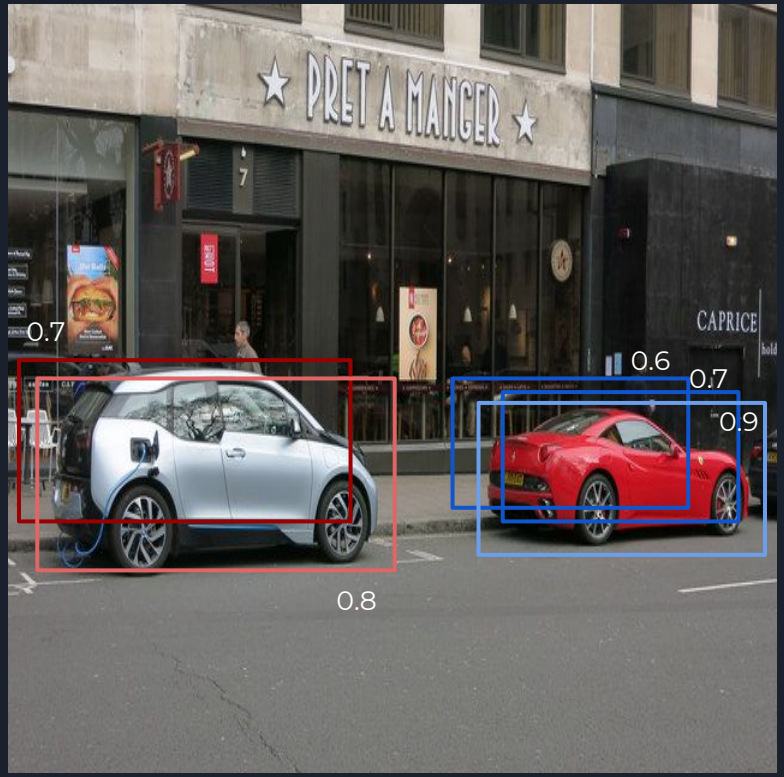
- definirea chenarului obiectului în interiorul unei celule:

$$y = \begin{bmatrix} 1 \\ bx \\ by \\ bh \\ bw \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.4 \\ 0.8 \\ 0.5 \\ 1.1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Detecția obiectelor - Intersection over Union (IoU)

- Evaluarea localizării obiectelor
  - IoU = $\frac{\text{size of}}{\text{size of}}$ 
  - Correct if IoU >= 0.5

Detecția Obiectelor - Non-Maximum Supression



- alege detecția cu scorul cel mai mare
- elimină detecțiile care se intersectează cu detecția aleasă la pasul anterior (IoU >= 0.5)
- repetă procesul pentru toate detecțiile din imagine

Detecția Obiectelor - Ancore

- până acum, am presupus că fiecare celulă poate conține cel mult un obiect
- rezultat de forma ***3x3x8*** (pentru exemplul cu 3 clase)
- dar dacă există mai multe obiecte în aceeași celulă?
- definim ancore de forme diferite și, pentru fiecare celulă, asociem predicții multiple, câte una pentru fiecare ancoră
- fiecare obiect este asignat ancorei cu cel mai mare IoU; rezultatul este acum de forma **3x3x(8 * #ancore)**

Detecția Obiectelor - Propunerea de Regiuni

- **R-CNN** (Regions with CNN features) - Propune regiuni (approx. 2000); clasifică regiunile propuse, pe rând => clasă (label) + chenar (bounding box)
- **Fast R-CNN** - Propune regiuni; folosește implementarea convoluțională a algoritmului de `Fereastră Glisantă` pentru a clasifica toate regiunile
- **Faster R-CNN** - Folosește CNN pentru a propune regiuni