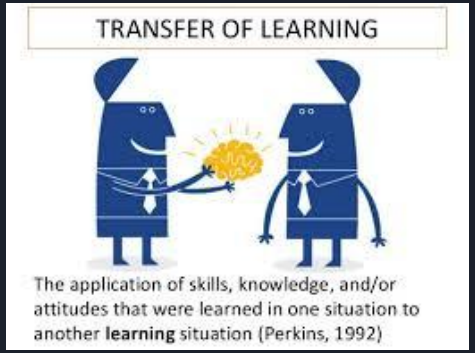
Detectie Segmentare

Curs 6

Transferul Parametrilor Invatati (1)



Problema :

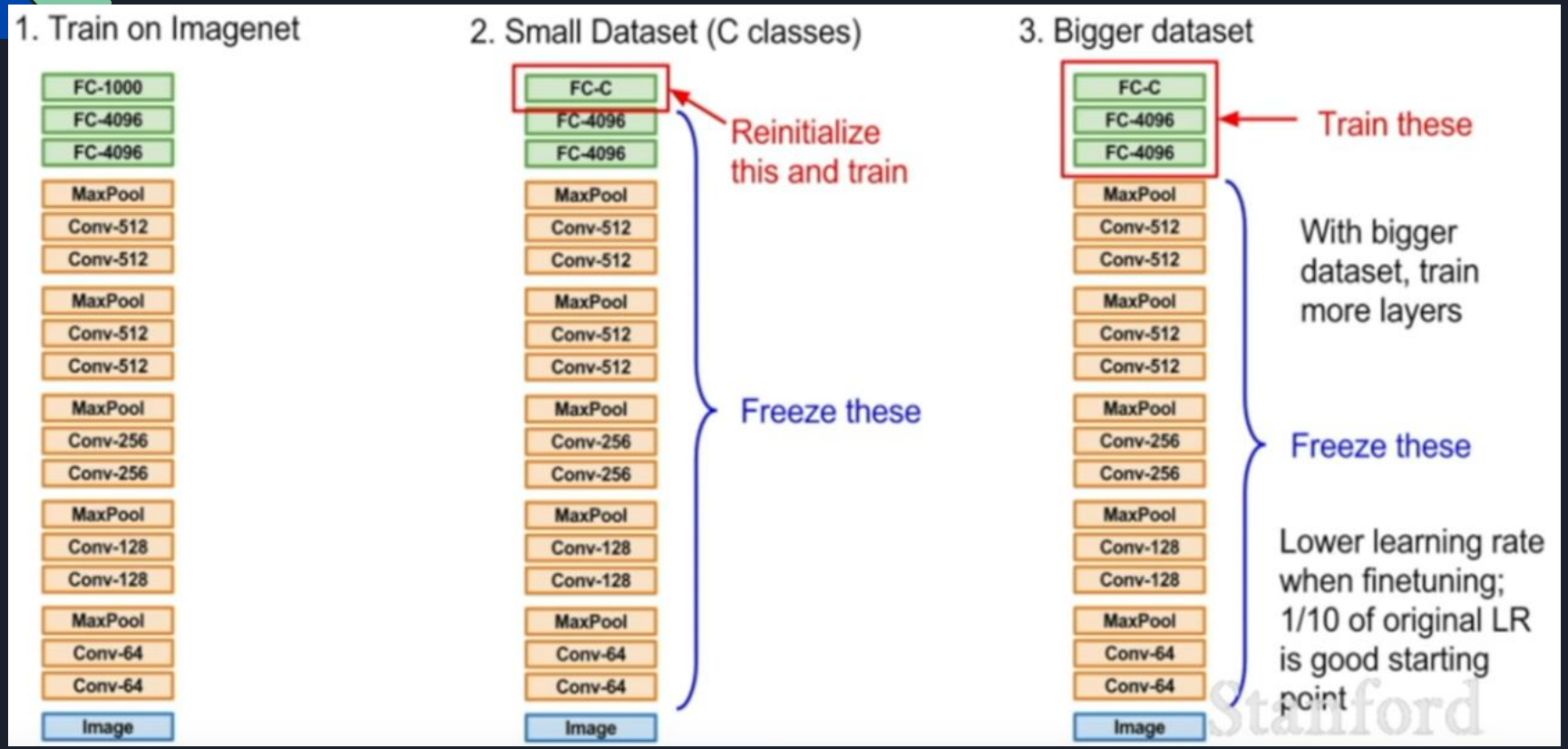
- Pentru o arhitectura complexa, cu un numar mare de parametrii, este necesar un dataset mare, pentru a nu face overfitting (invatare mot-a-mot a datasetului)

Solutie :

- In cazul in care avem un dataset mic, dar cu obiective de antrenare / invatare similare cu cele ale unor CNN'uri antrenare pe alte dataseturi, mari, putem sa preluam din "knowledge" ul acelor retele neurale prin "Transfer Learning"

"You need a lot of data if you want to train/use CNNs" - mith busted

Transferul Parametrilor Invatati (2)



Transferul Parametrilor Invatati (3)

	Dataset Similar	Dataset Diferit
Dataset Mic / Fara diversitate	Schimb clasificatorul liniar de pe ultimul layer al unei retele pre-invatate	Greu de rezolvat. Reinitializeaza / Finetune parti mai mari din retea. Experimenteaza / Augmenteaza
Dataset Bogat / Complex	Finetune ultimele cateva layere. Cate? Cum va functiona reteaua pe datasetul initial?	Finetune peste un numar mare de layere. Permite retelei sa invete noile caracteristici ale Datasetului.

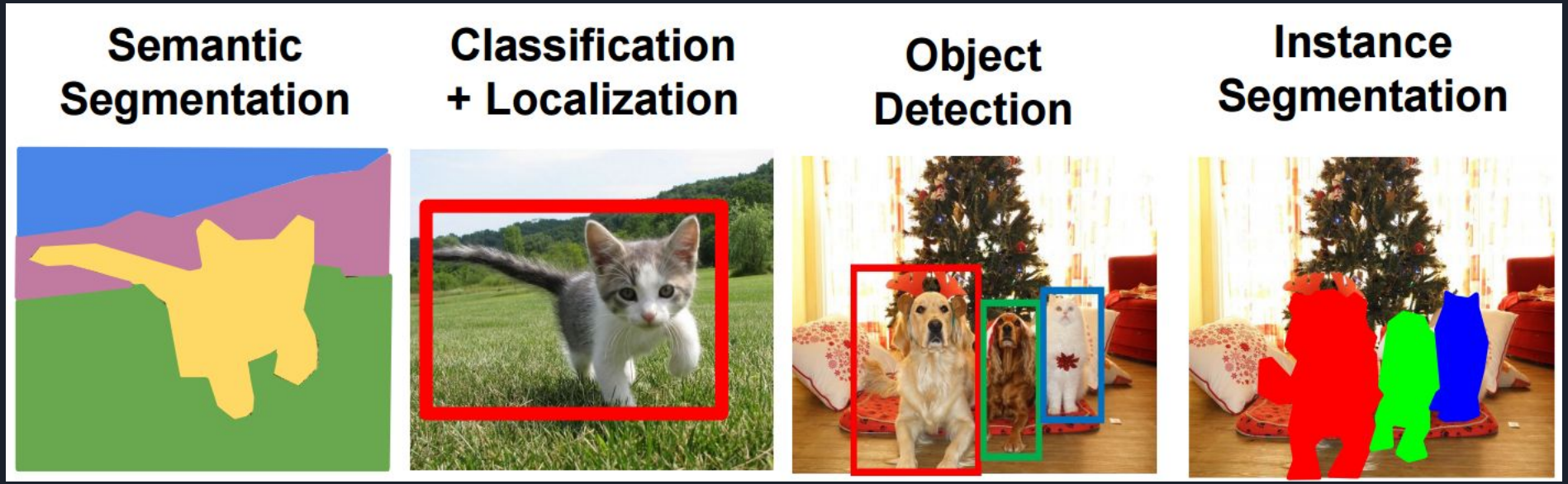
Recapitulare Task Clasificare

- Imagine de Input ---- CNN ----- Clasificare 1 obiect.
- Obiectul poate sa fie oriunde in poza, in orice pozitie, dar preferabil in foreground, de dimensiune mare

- Softmax / CrossEntropy ca functie de loss.
- One-Hot-Encoding ca reprezentare a datelor
- 1 sau mai multe layere FC la sfarsitul retelei.

Ce se intampla daca sunt mai multe obiecte in poza?
Ce alta informatie ne intereseaza, legat de instanta fiecarui obiect?
Ce se intampla daca obiectele se suprapun?

Urmatoarele Taskuri in Computer Vision



Segmentare Semantica (1)

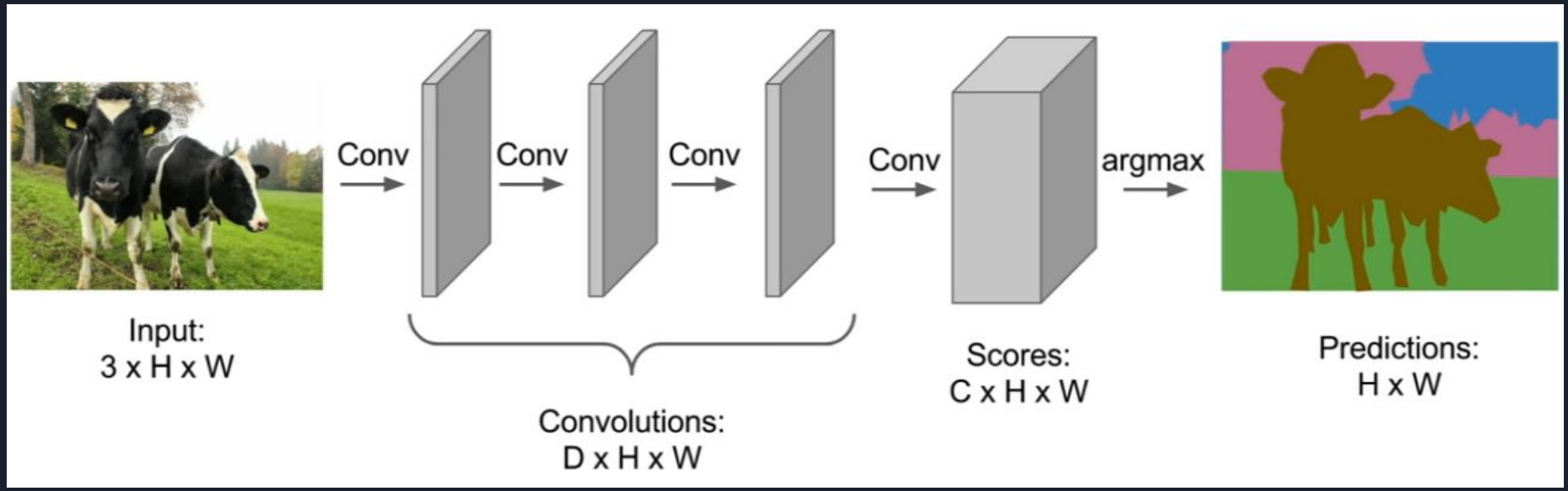
Imagine Input ------ CNN ---- Masca Segmentare

- Nu diferentiaza intre instantele de obiecte
- Conteaza doar clasificarea fiecarui pixel in clasa din care face parte
- CrossEntropy loss peste fiecare pixel

Cum arata Ground Truth-ul ?

Segmentare Semantica (2)

Abordarea Naiva / Vanilla. Observatie : Doar Layere Convolutionale : FCN
Care este problema in cazul acesta?
Cum dezvolti dataset cu GT pentru Segmentare semantica?

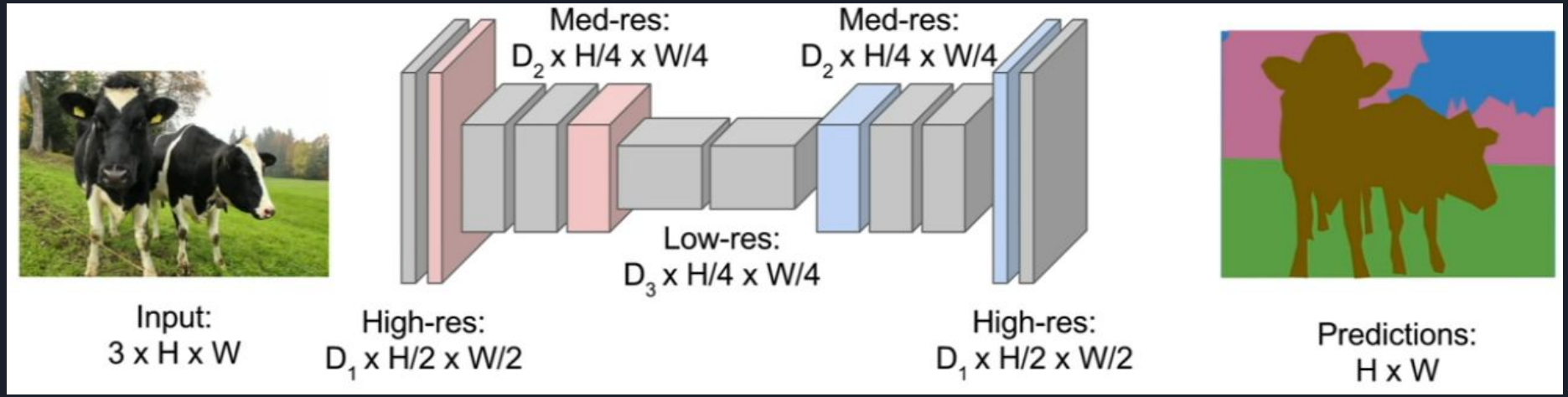


Segmentare Semantica (3)

- In practica, este ineficient sa pastram layere convolutionale (cu acelasi width / height si depth 64 / 128 / 256) pe toata adancimea arhitecturii. (numar de parametrii foarte mare, consum de memorie foarte mare) =>

Arhitectura Clepsidra :

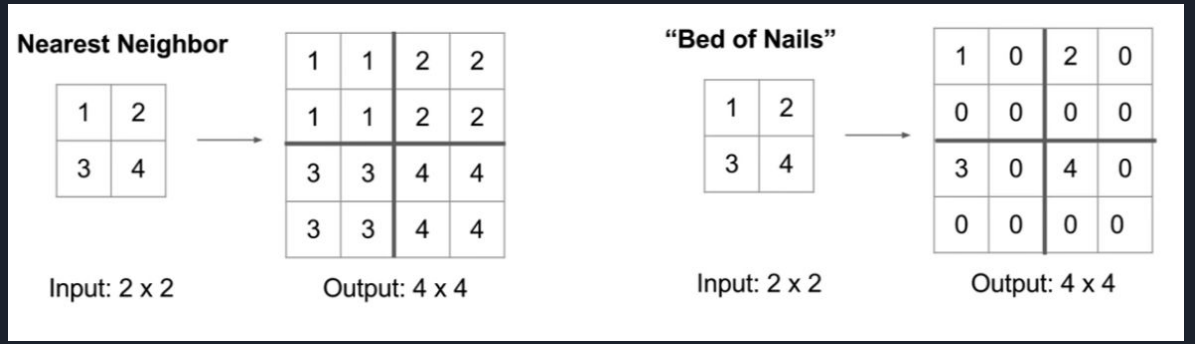
- Input Image ----- Downsampling (pooling / stride) ------ Upsampling ------ Output Image Mask



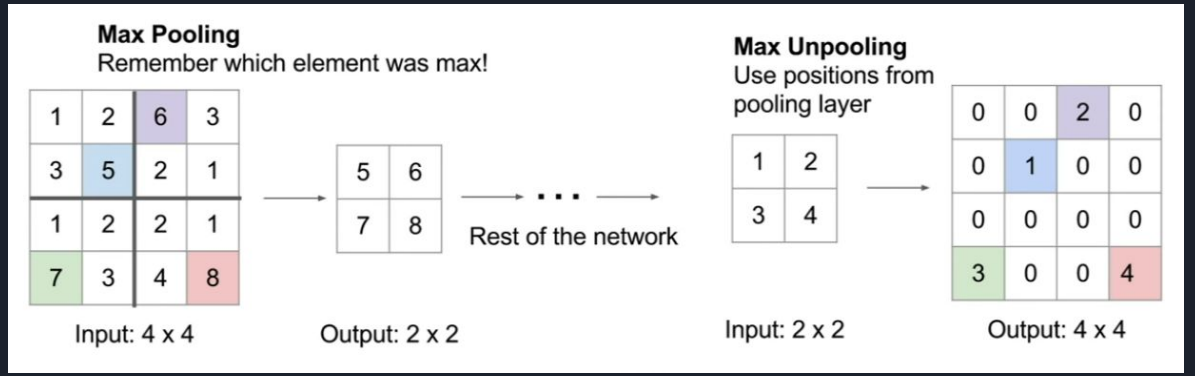
Segmentare Semantica (4)

Upsampling :
(UnPooling)

- Nearest Neighbor
- Bed of Nails



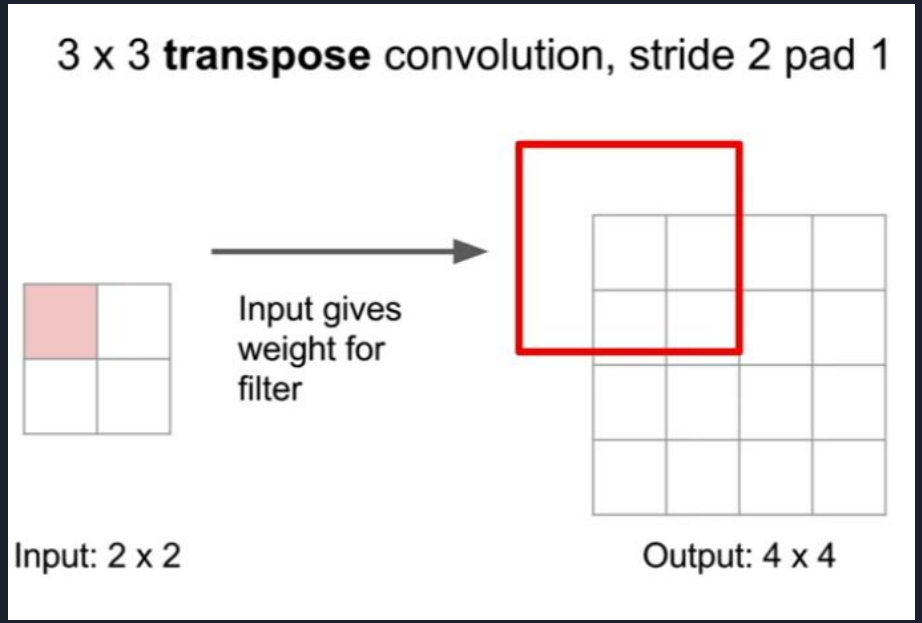
- Max Unpooling
(arhitecturi simetrice)
Pastreaza din informatia spatiala pierduta prin MaxPooling



Segmentare Semantica (5)

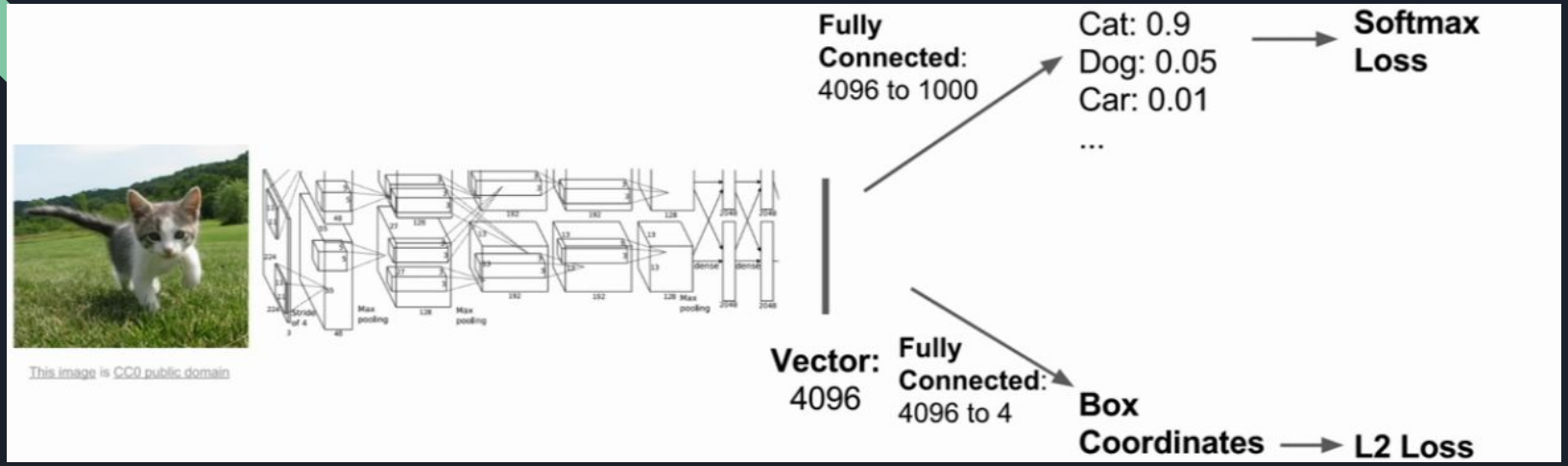
Upsampling cu parametrii invatabili : Convolutie Transpusa

- Face upsampling la feature-map invatand prin parametrii unei convolutii felul in care trebuie sa distribuie, ponderat, valorile pentru urmatorul feature-map



- Valorile filtrului W sunt inmultite cu valoarea de input din feature map => In output vor exista copii ponderate ale W
- Acolo unde aceste copii se suprapun, valorile W*pondere se aduna.
- Se mai numeste si “Deconvolution”, ”Fractionally Strided Convolution”, ”Backward Strided Convolution”

Clasificare + Localizare



- Adauga inca un task final retelei (multitask loss) : localizare.
- Trateaza localizarea ca pe o problema de regresie (outputul corespunde unor valori intregi, continue, nu unor indici de clasa, categorici / reprezentanti ai unei categorii) =>
  - Nu se mai foloseste Softmax. (L1, L2, SmoothL1 etc)
  - GT nu se mai vectorizeaza one-hot.
  - Coordonatele boxului / valori intregi : (x,y,w,h) --------- L2 Loss -------- Coordonatele boxului GT (x' , y' , w' , h')

$$S = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

- Se aplica o pondere fiecarui loss. Aceasta pondere este hyper-parametru, si schimba in mod direct "cat de afectata este reteaua" de importanta fiecarui task !

Detectie de obiecte (1)

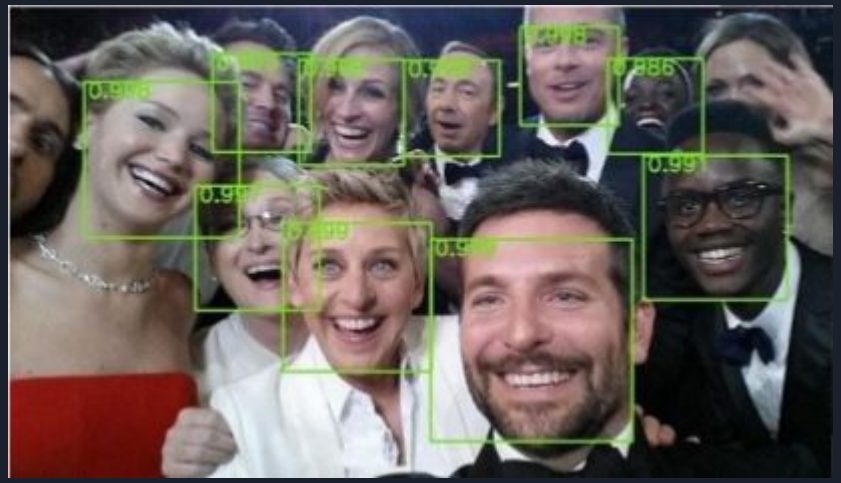
Detectia tuturor obiectelor din poza (localizare + clasificare) care corespund unui set fixat / cunoscut de clase.

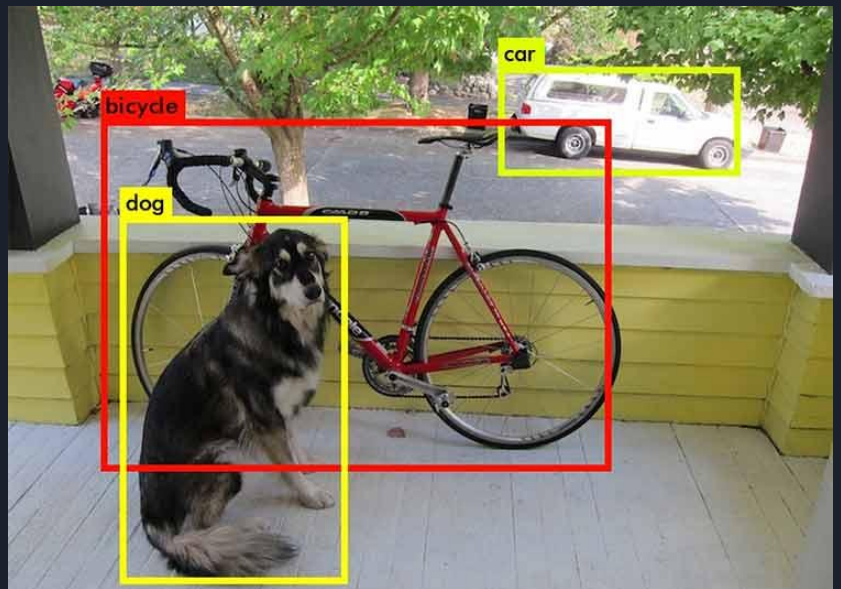
- Numar variat de obiecte pentru fiecare imagine in parte.
- Diferite pozitii / dimensiuni / grade de ocluzia sau truncare pentru fiecare obiect in parte

Q:

- Cum apare un singur box pentru fiecare obiect?
- Sliding window approach?
- Ce scor de clasa o sa am pentru zonele care nu au obiecte din setul cunoscut de clase?

(hint : add “background” class)

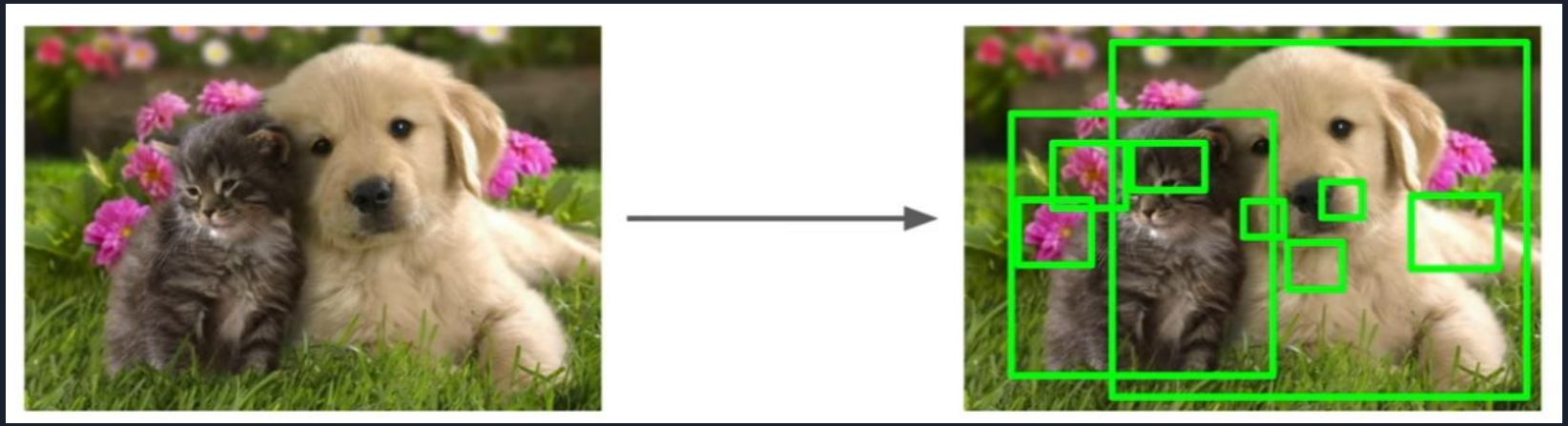




Detectie de obiecte (2)

Propunere de regiuni : (Region-Proposal / Regions Of Interest / ROIs)

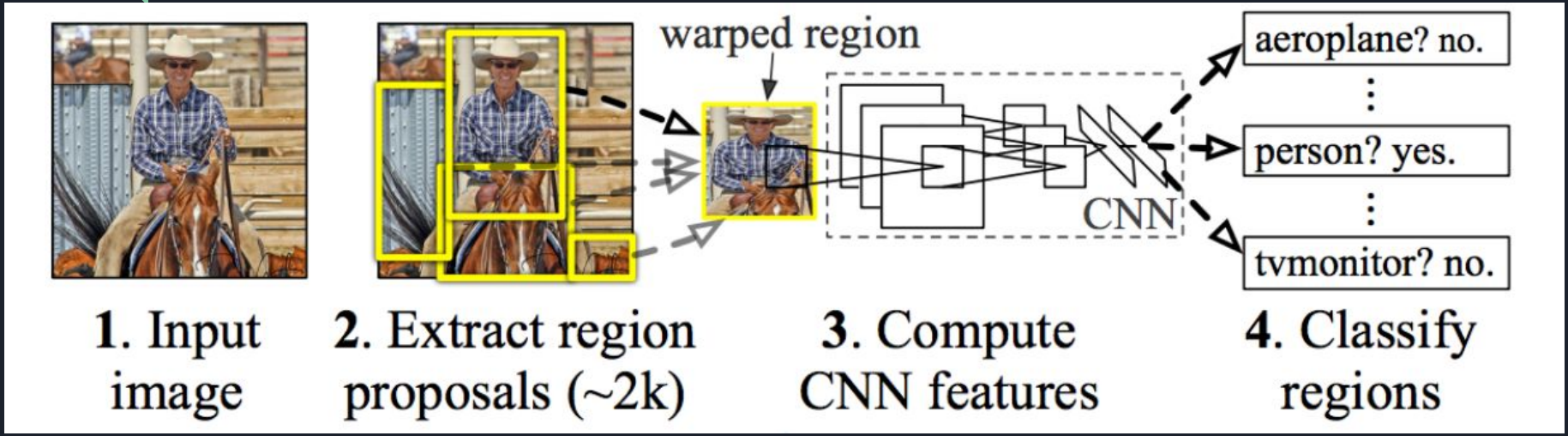
- Naiv : cauta zone “blobby” (cu masa de pixeli asemanatori si/sau edgeuri inchise), care au sansa mare sa contina un obiect.
- Output este un numar fix de propuneri unde poate sa fie un obiect
- Exemplu : Selective Search (2000 propuneri / 2 secunde CPU)



- Idee : Aplica Selective Search pe imagine, apoi CNN Clasificare pe fiecare crop (o varianta mai performanta computational decat sa incerci toate combinatiile de regiuni si scale posibile)

R-CNN 2014

https://github.com/rbgirshick/rcnn
https://arxiv.org/pdf/1311.2524.pdf

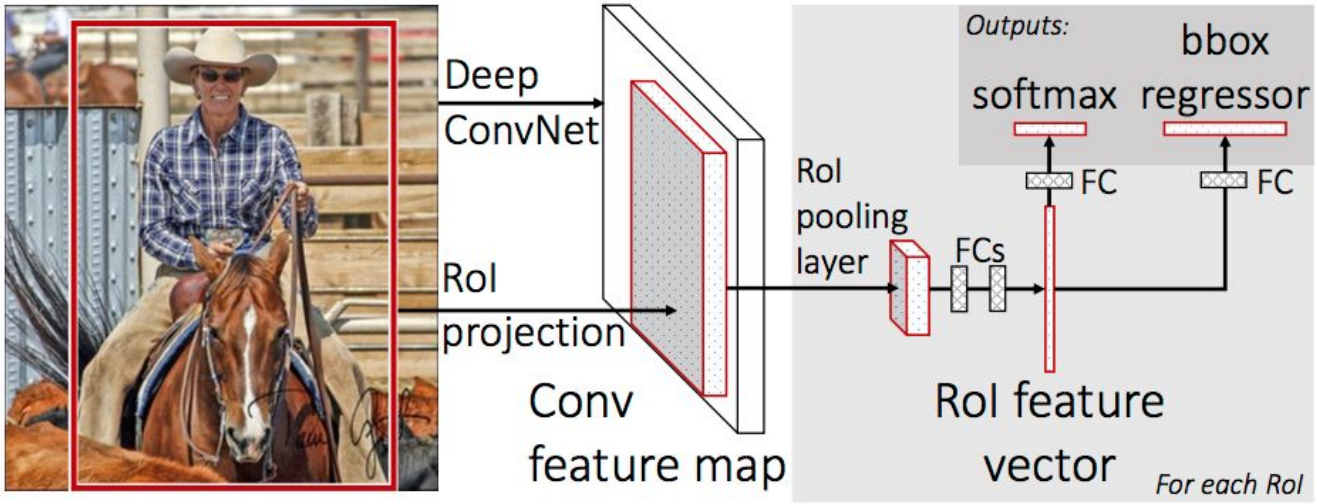


OBS : Fiecare crop este re-dimensionat pentru Input CNN

Timp de inferenta per imagine? Idei pentru optimizare?
Cum arata GT? Cum se calculeaza loss'ul pe fiecare crop? Hint : IoU

Fast R-CNN 2015

https://github.com/rbgirshick/fast-rcnn
https://arxiv.org/pdf/1504.08083.pdf



1. Input Image
2. CNN ---- feature Maps
3. Region Proposal pe ultimul layer de activari din CNN (shared computation)
4. Warp / reshape crops din feature map (ROI pooling layer)
5. “Second Stage” :

- Clasificare per crop (FC layers + Softmax)
- Localizare per crop (FC layers + Smooth L1 Loss) : localizarea regreseaza delta (x,y,w,h), adica ajusteaza pozitia cropului propus de Selective Search, in functie de GT, pentru o mai buna mapare cu fiecare obiect in parte.
- Test Time : 2.3 secunde (cu tot cu Selective Search) / 0.32 sec (forward + backpropagation)

Faster-RCNN 2016

https://github.com/rbgirshick/py-faster-rcnn
https://arxiv.org/pdf/1506.01497.pdf

Problema : Selective Search dureaza prea mult !

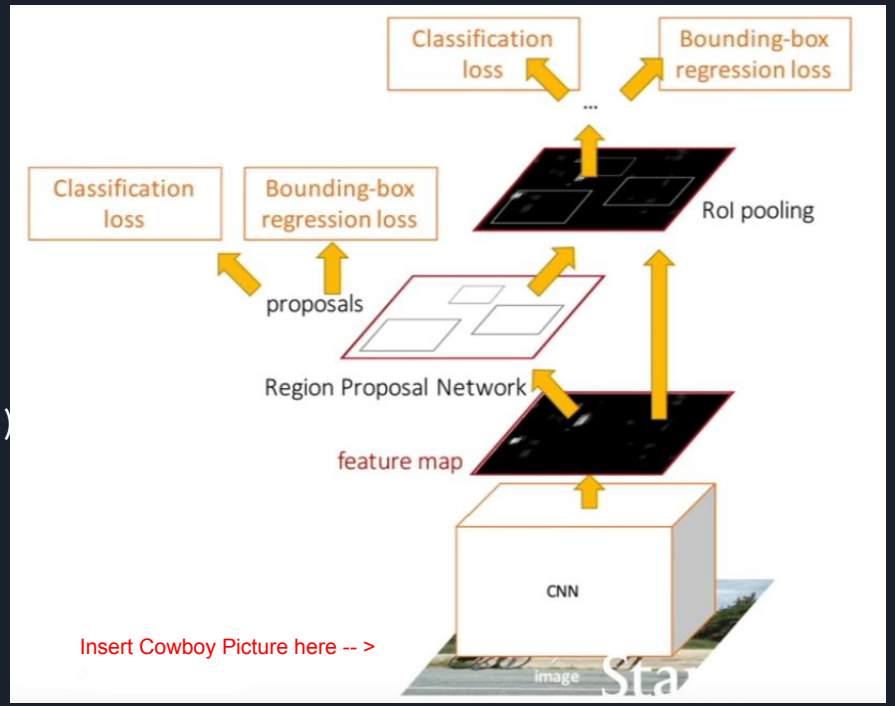
Solutie : Inlocuim Selective Search cu RPN (retea neurala care propune regiuni (localizare) si le clasifica in “obiect” / “notobiect”)

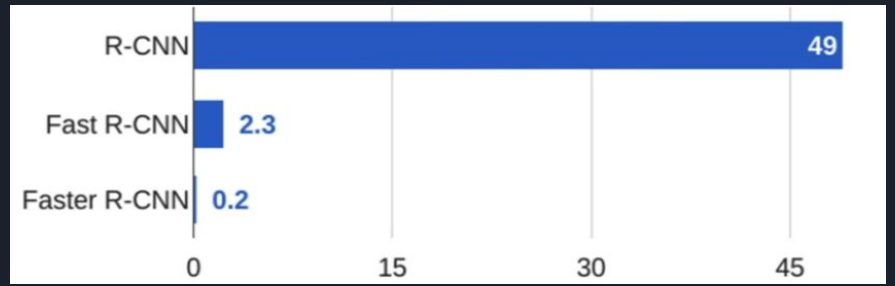
4 Loss’uri :

- First Stage : RPN regression coordonate crops
- First Stage : RPN classification crops (object / not object)
- Second Stage : Clasificare obiect, per crop
- Second Stage : Regresie delta / ajustare coordonate, per crop

Tutorial / Tips / Tricks / Go Crazy / Become a ML Researcher :
https://github.com/tensorflow/models/tree/master/research/object_detection

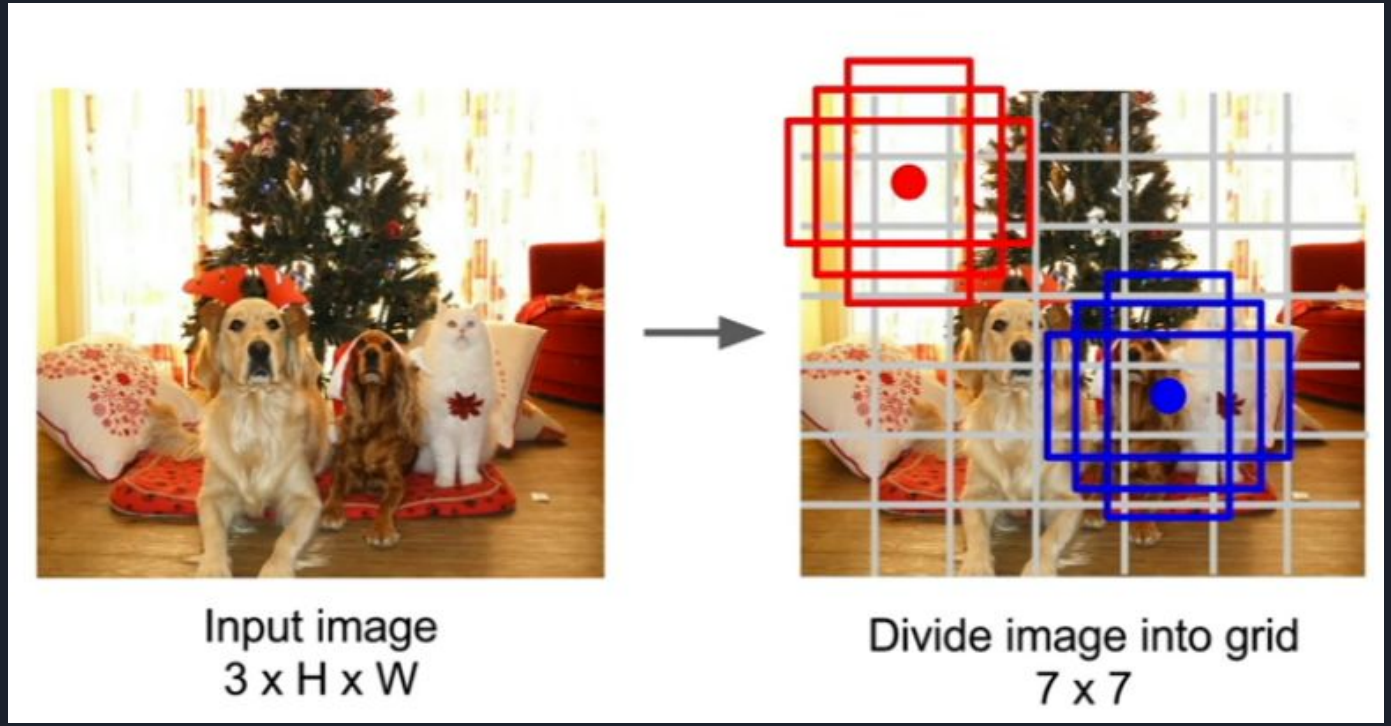
https://arxiv.org/pdf/1611.10012.pdf





Detectie Fara Propuneri de Regiuni

YOLO : https://arxiv.org/pdf/1506.02640.pdf
SSD : https://arxiv.org/pdf/1512.02325.pdf



Q & A