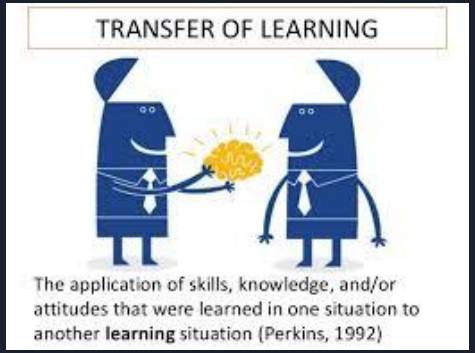
Segmentare Semantica

Curs 6

Transferul Parametrilor Invatati (1) (Knowledge Transfer)



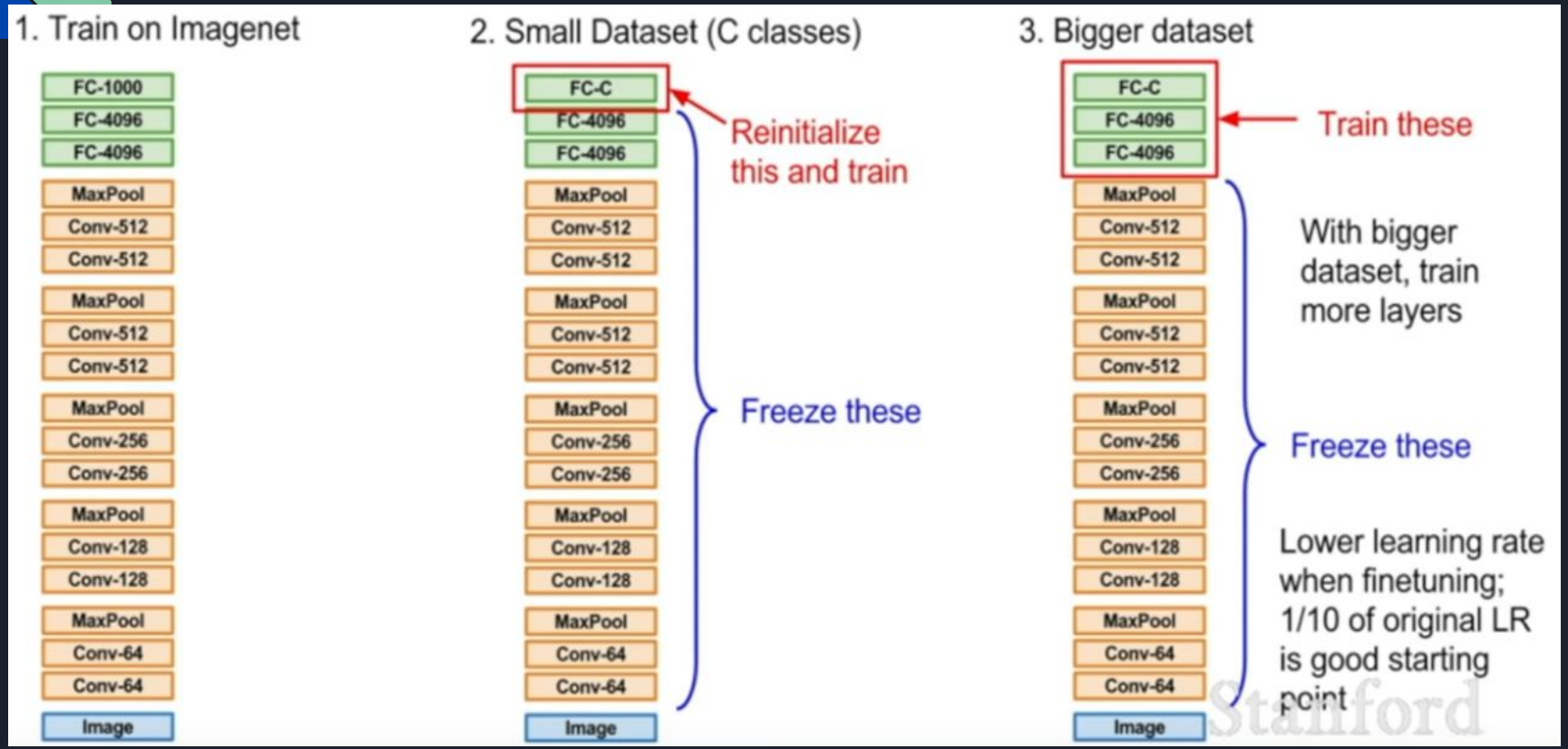
Problema:

- 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:

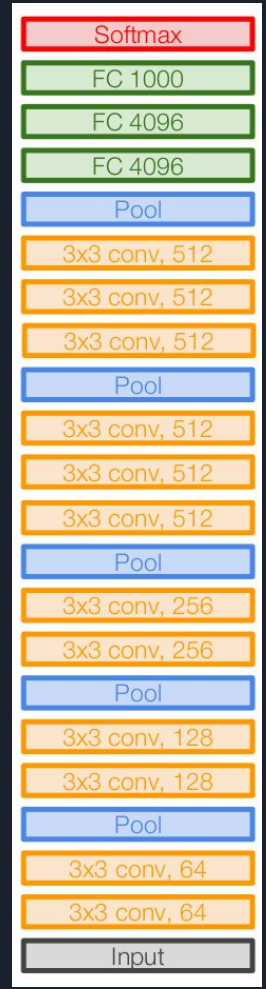
- In cazul in care avem un dataset mic, dar cu obiective de antrenare / invatare similare cu cele ale unor CNN'uri antrenare pe alte dataseturi, mari, putem sa preluam din "knowledge" ul acelor retele neurale prin "Transfer Learning"

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.

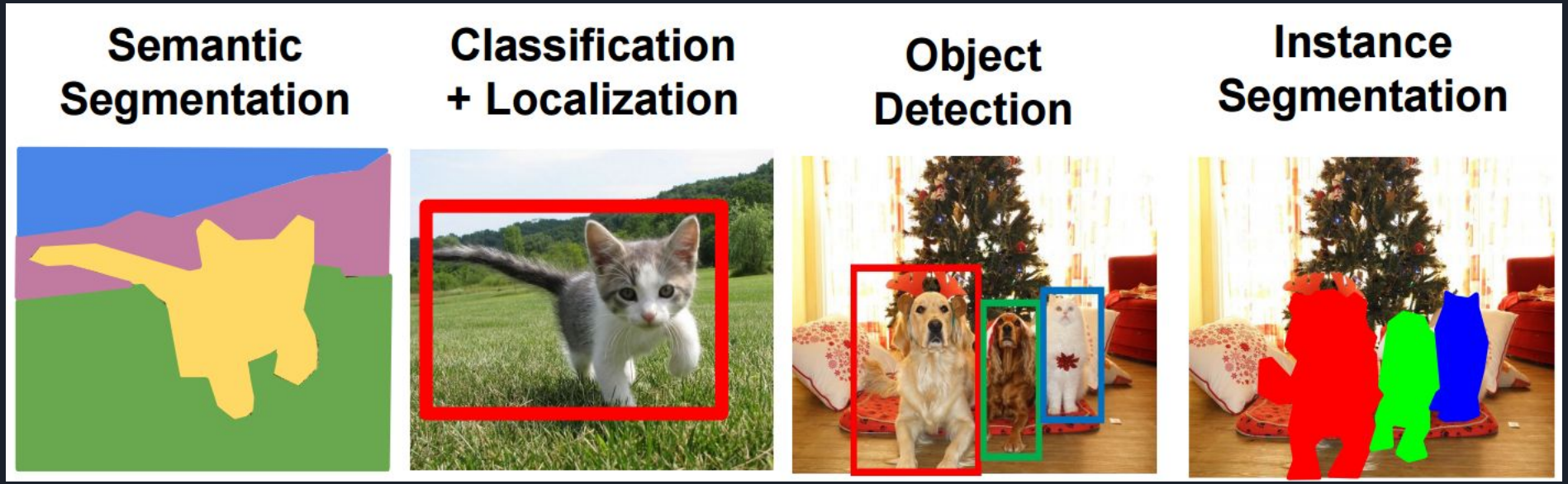


Recapitulare Task Clasificare

- Imagine de Input CNN Clasificare 1 obiect/poza
  - Answer DA/NU
- 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.
  - Posibil sa folosim Global Average Pooling in loc de flatten
  - Dar vom avea un layer FC la final

- 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)

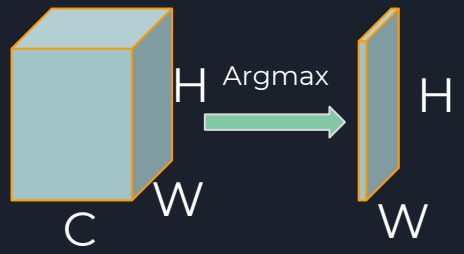
Imagine Input -> CNN -> Masca Segmentare

- 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 ?

Intuitie pentru Semantic Segmentation

- Avem o retea de clasificare binara: 1/0 per poza, sau clasificare pe C clase
- Dorim sa extindem 1/0 per poza la 1/0 per pixel (segmentare binara)
- Segmentare: pixelwise classification
- Convoluțiile păstrează ordinea spatiala
  - Ordinea este pastrata, dar avem height/width mai mic
  - Putem face upsampling pentru a recastiga rezolutia din input
  - Din natura convolutiilor, vine firesc sa folosim doar convolutii pentru dense classification: segmentation
- In dreapta:
  - C numarul de clase: avem volum HxWxC
  - Per pixel i, j = (1:H, 1:W) alegem:
    - Id-ul clasei C cu cel mai bun scor
  - Softmax-ul este pixelwise in dimensiunea C



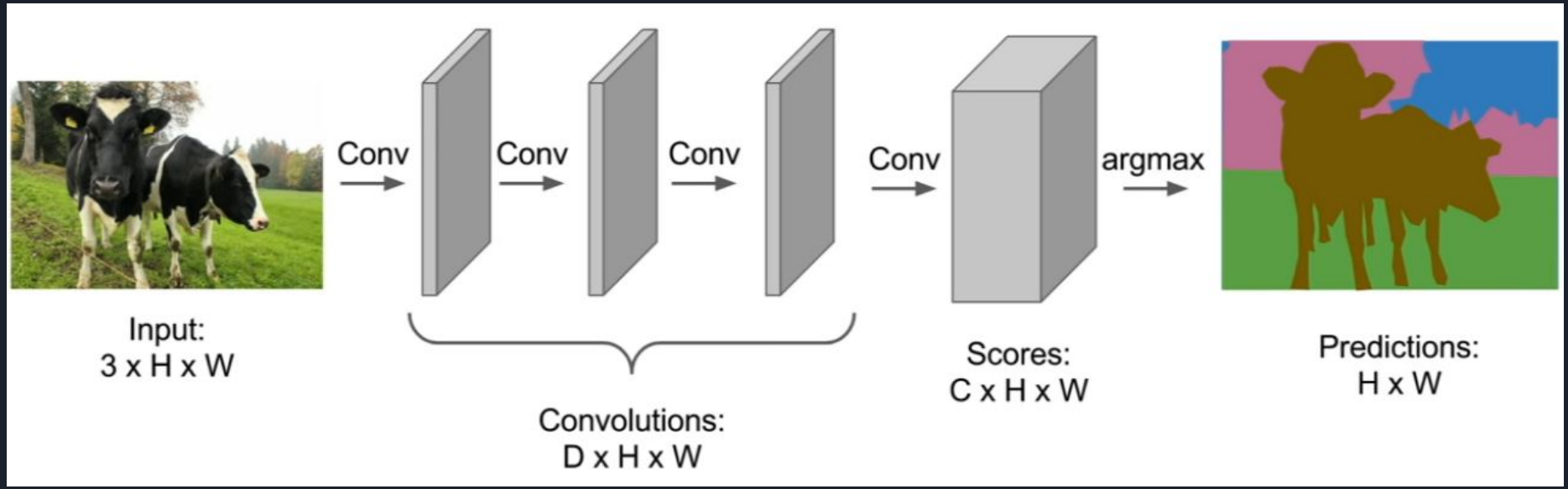
Segmentare Semantica (2)

Abordarea Naiva / Vanilla. Observatie : Doar Layere Convolutionale :

Fully Convolutional Neural Network (FCN)

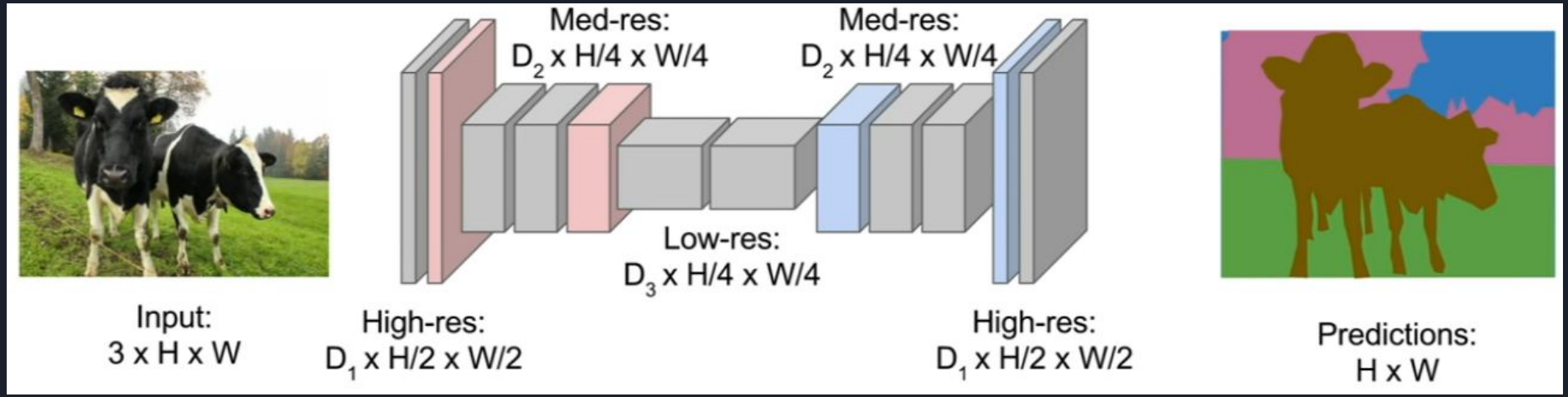
Care este problema in cazul acesta?

Cum dezvolti dataset cu GT pentru Segmentare semantica?

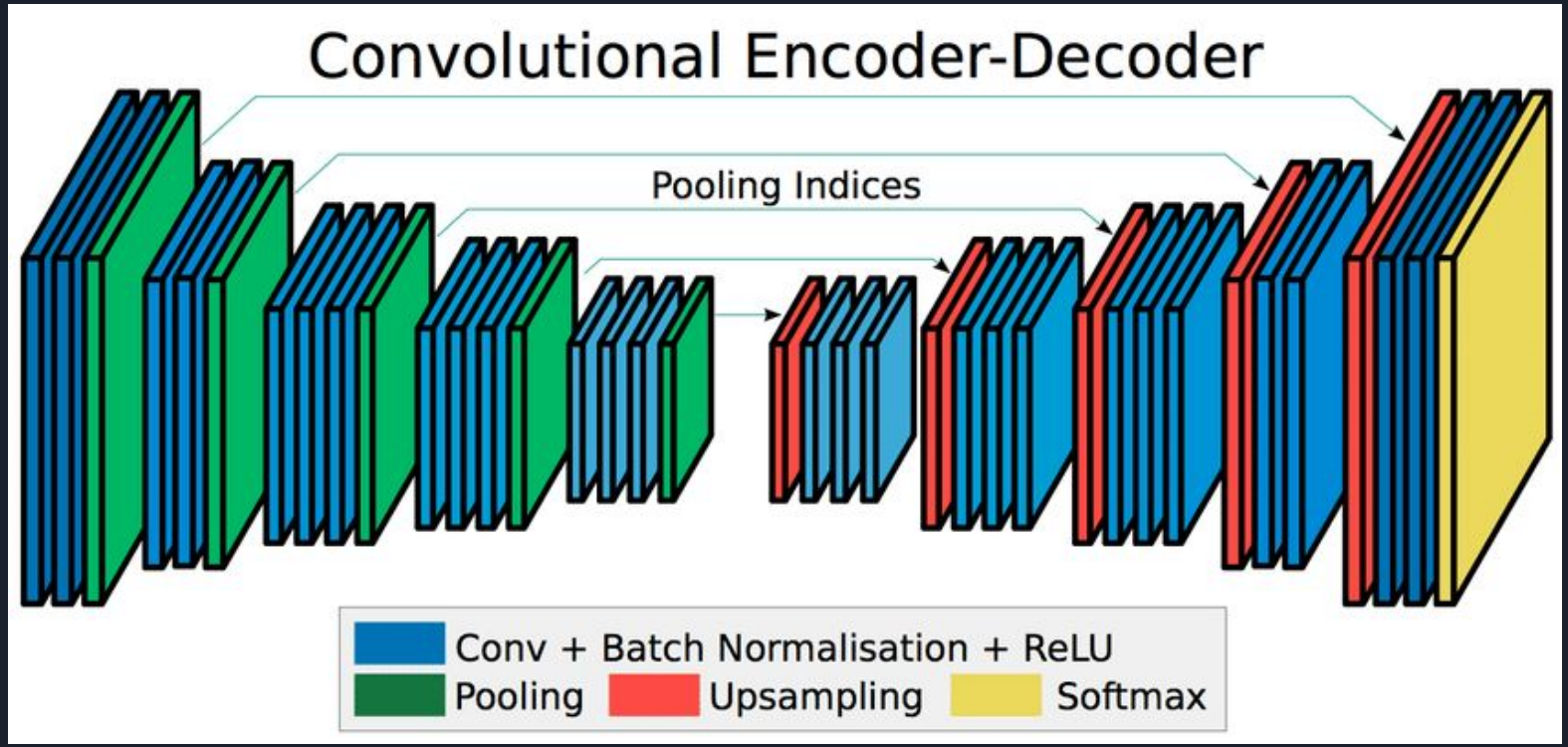


Segmentare Semantica (3)

- 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 (Hourglass - Encoder/Decoder):
- Input Image -> Downsampling (pooling / stride) > Upsampling < Output Image Mask



Encoder/Decoder Hourglass (3)



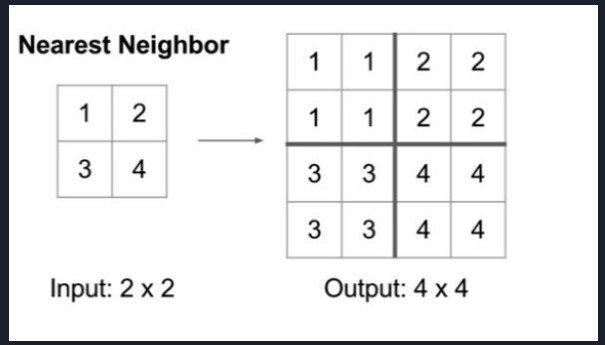
Encoder Part

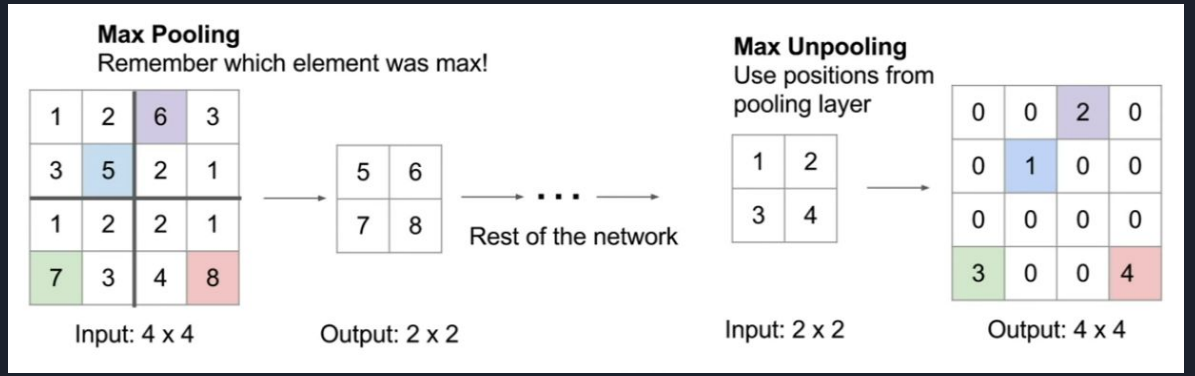
Decoder Part

Upsampling

Upsampling :

- Nearest Neighbor
- Bilinear Upsampling
- Transposed Convolution
- Max Unpooling

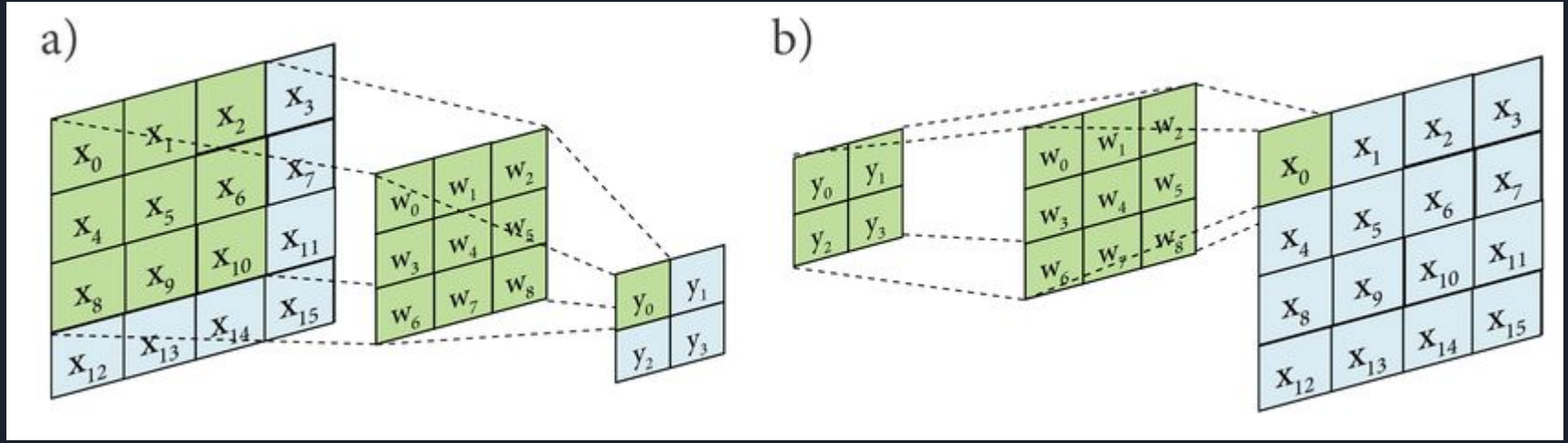




Convolutia Transpusa

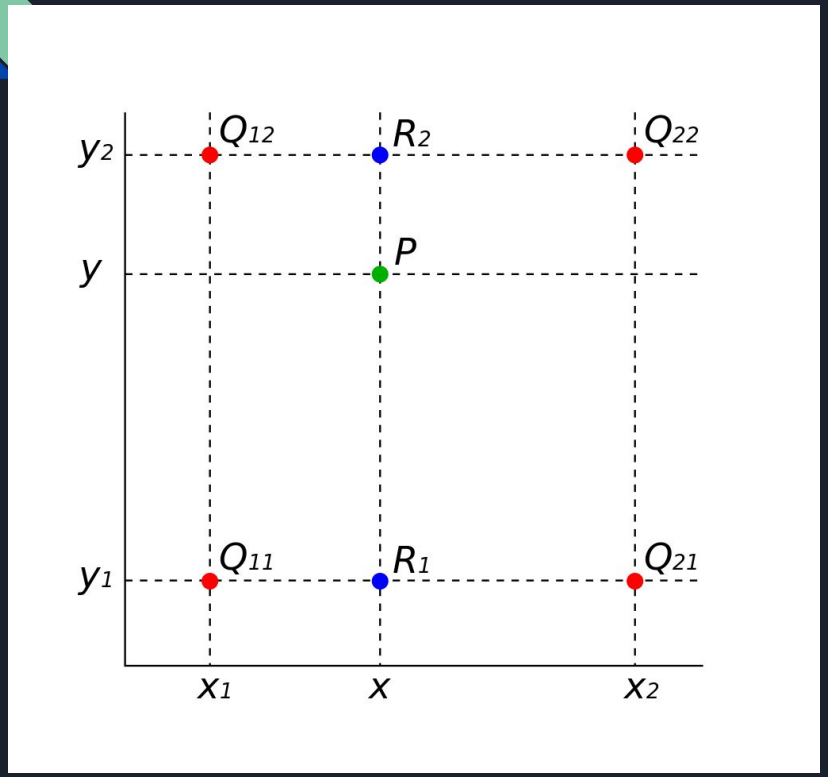
Upsampling cu parametri invatabili : Convolutie Transpusa

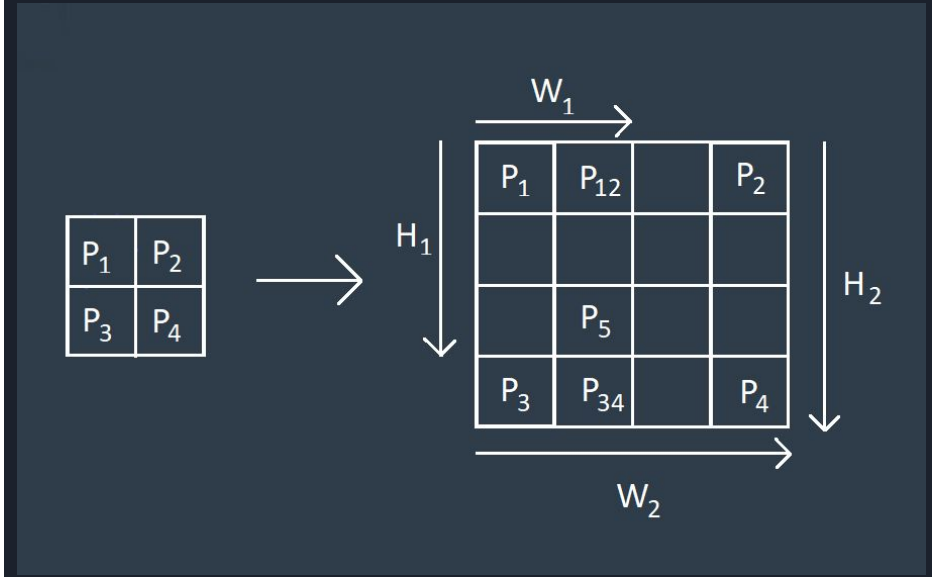
- In forward pass avem convolutie normala. In backward pass avem convolutie transpusa
- Face upsampling la feature-map invatand prin parametrii unei convolutii felul in care trebuie sa distribuie, ponderat, valorile pentru urmatorul feature-map
- Valorile filtrului W sunt inmultite cu valoarea de input 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 Conv



http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html
https://arxiv.org/pdf/1603.07285.pdf
https://github.com/vdumoulin/conv_arithmetic

Bilinear Interpolation



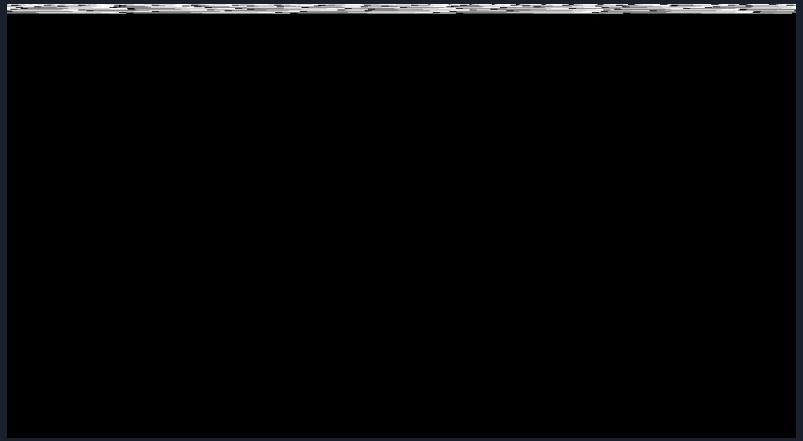


Intuitie pentru convolutia transpusa: invatam modul de influenta al vecinilor

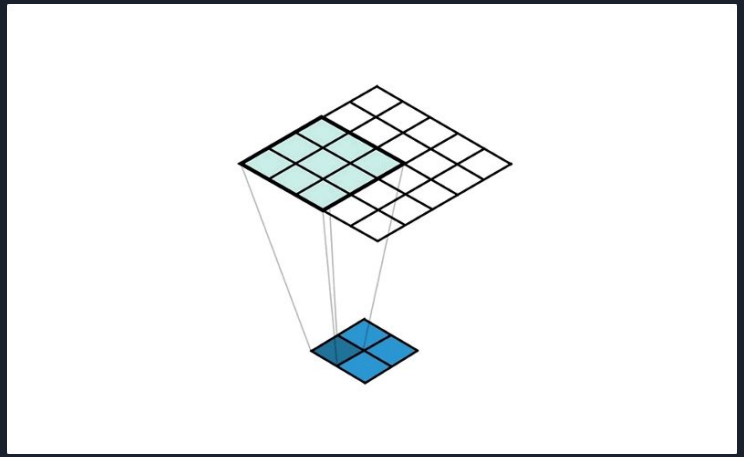
Convolutia Transpusa

- Intuitiv, cu cat o valoare (pixel) e mai aproape de centru in output, cu atat acumuleaza **“fractiuni”** din mai multe valori (pixeli) de input
- Stride-ul se aplica acum in **output**: stride mai mare -> dimensiune mai mare
  - Stride-ul mare produce un output mai mic decat input/s in convolutia directa
- Inversam input/output: input-ul devine output si vice versa:
  - $\mathbf{Input^T = Output_C}$
  - $\mathbf{Output^T = Input_C}$
  - Se doreste reconstituirea input-ului care a fost folosit in convolutia directa pentru generara output-ului
- Dimensiuni output: (I - 1) * S + K

Stride 1

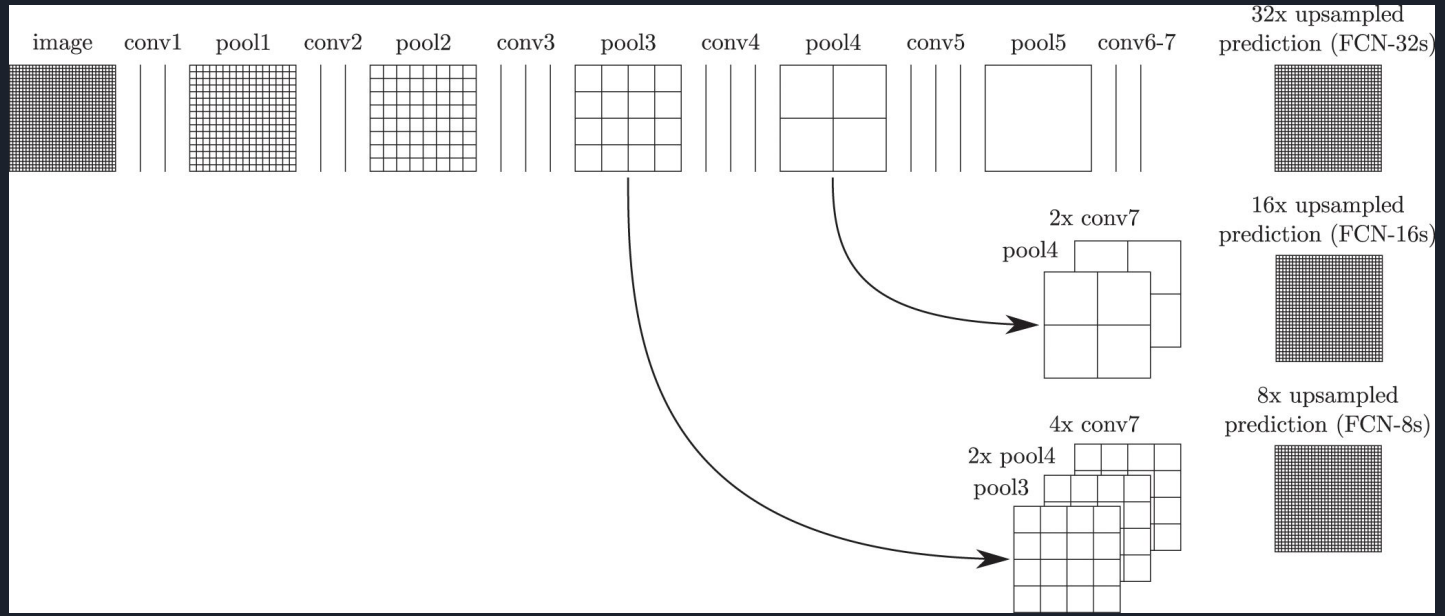


Stride 2

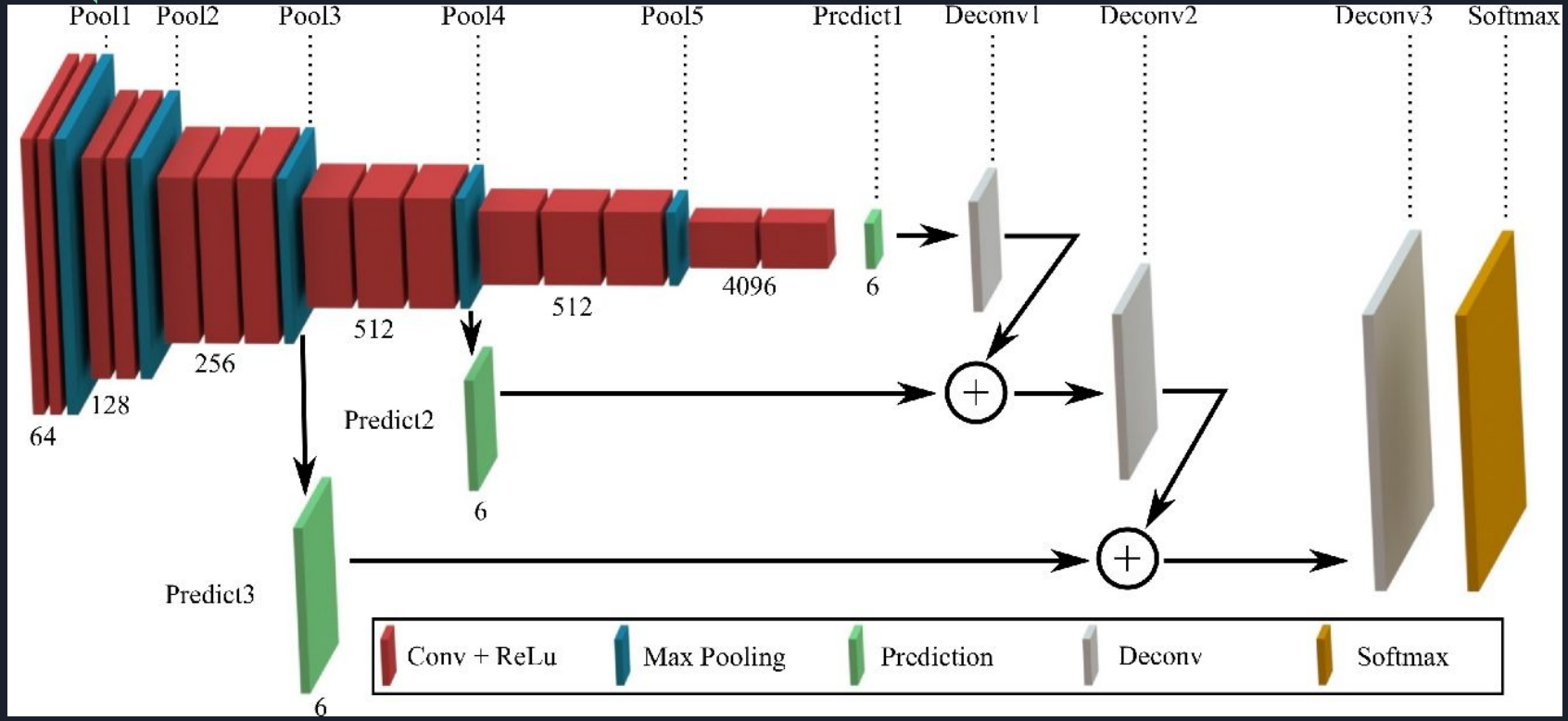


Fully Convolutional Networks - E Shelhamer

- Skip connections
- Upsample from stride 16 to stride 8:
- Use 1x1 bottlenecks to compress to number of classes
- Add with encoder counterpart from stride 8 (1x1 compressed to number of classes)



Fully Convolutional Networks



Resurse

- Aritmetica Convolutiilor
  - http://deeplearning.net/software/theano/tutorial/conv_arithmetic.html
  - https://arxiv.org/pdf/1603.07285.pdf
  - https://github.com/vdumoulin/conv_arithmetic
  - https://towardsdatascience.com/up-sampling-with-transposed-convolution-9ae4f2df52d0#d907
  - https://medium.com/apache-mxnet/transposed-convolutions-explained-with-ms-excel-52d13030c7e8
- Segmentare Semantica folosind FCN
  - https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

Q & A