Review from deepsystems.io

DeepLab

Semantic Image Segmentation with Deep Convolutional Nets

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille

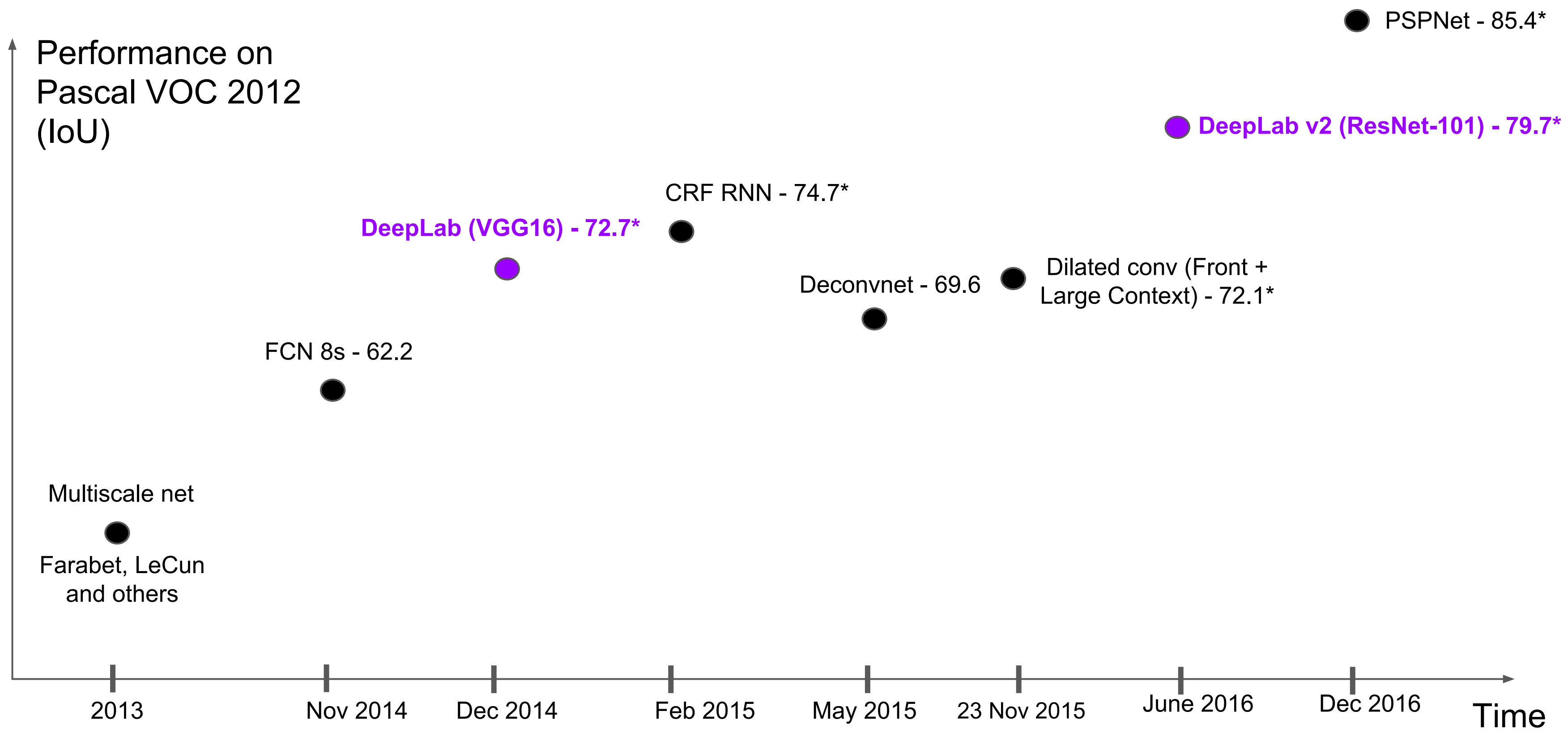
Semantic Segmentation Task

input image

per-pixel
class labels

deepsystems.io

The Big Picture

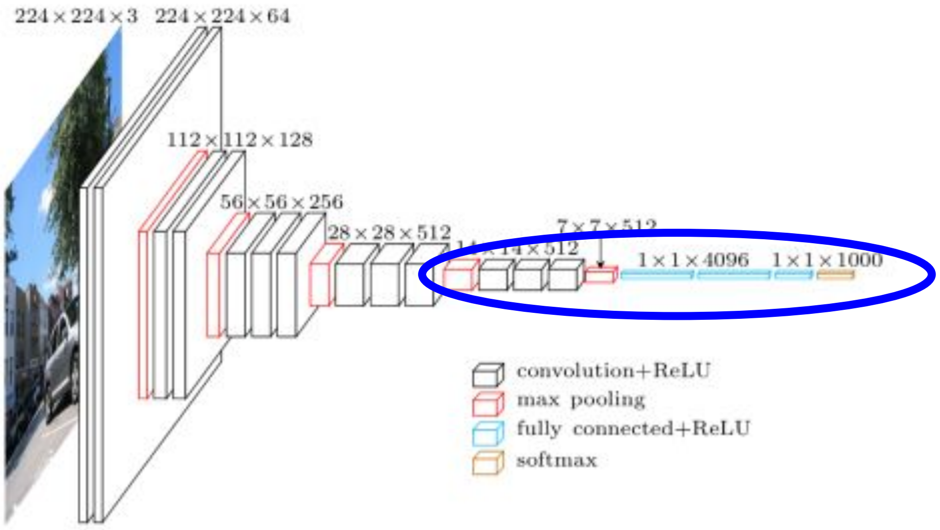


* MS COCO dataset was also used for training

Outline

- Recap: Convolutional networks for semantic segmentation - pros and cons
  - Focus on representing spatial info
- Trick #1: **dilated convolutions**
  - Widen receptive field effectively
  - Avoid spatial resolution coarsening
- Trick #2: **conditional random field** for segmentation post-processing
  - Extra smoothing for better local consistency
  - Align segment boundaries with sharp changes in the image

Recap: Deep Convolutional Nets for Classification



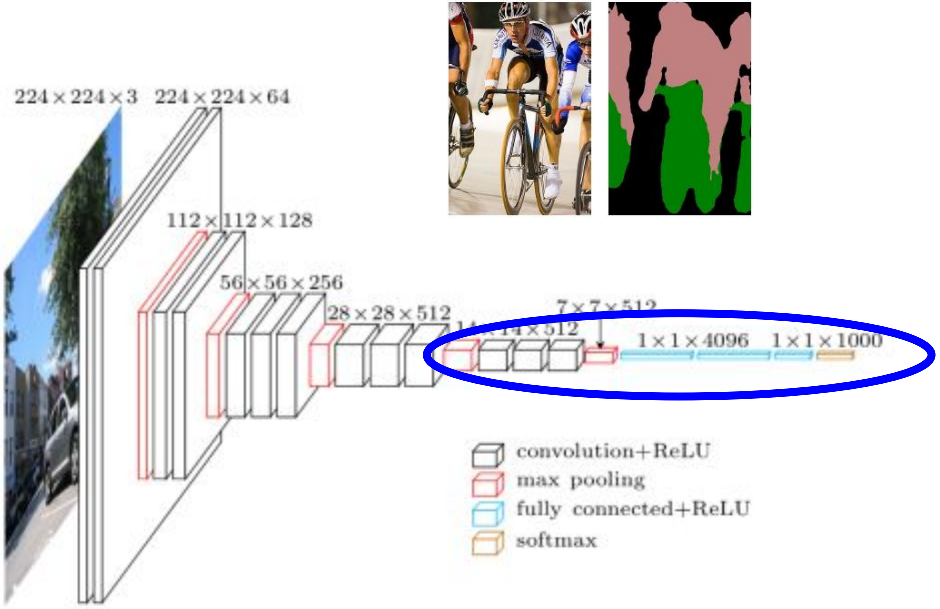
Only need to reason about the image as a whole!

Success factors:

- Wide receptive field → global information
- Spatial invariance

Image credit: Davi Frossard, https://www.cs.toronto.edu/~frossard/post/vgg16/

Recap: Deep Convolutional Nets for Segmentation?



~~Only need to reason about the image as a whole~~

Need to reason about individual pixels!

Success factors?

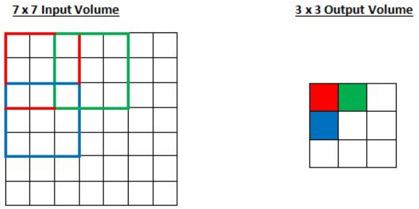
- Wide receptive field → still great!
- Spatial invariance → now bad
  - Need to preserve spatial info!

Image credit: Davi Frossard, https://www.cs.toronto.edu/~frossard/post/vgg16/

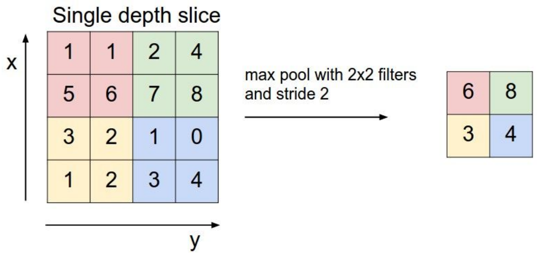
deepsystems.io

Recap: Segmentation-Specific Challenges

- Want both **wide receptive field** and **high spatial resolution**
- Standard way to grow receptive field - add **strided** convolution and pooling layers:



convolution, stride 2



max pooling, stride 2

- Strided layers decrease spatial resolution

Image credit: [Adit Deshpande](), [Andrej Karpathy]()

deepsystems.io

Recap: Possible Compromise - Learn to Upsample

- “Deconvolutional networks”
  - Convolution/pooling blocks to very coarse resolution (e.g. 1/32 of original)
  - Classify coarse cells
  - Upsample to original resolution
    - Maybe learn the upsampling parameters
- Can be made to work very well

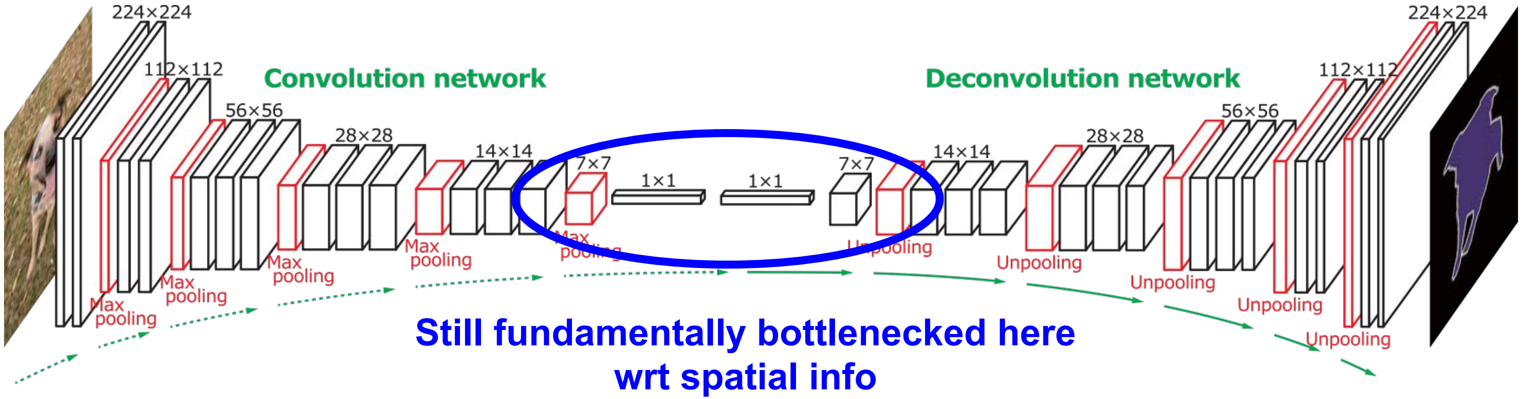


Image credit: http://cvlab.postech.ac.kr/research/deconvnet/

Outline

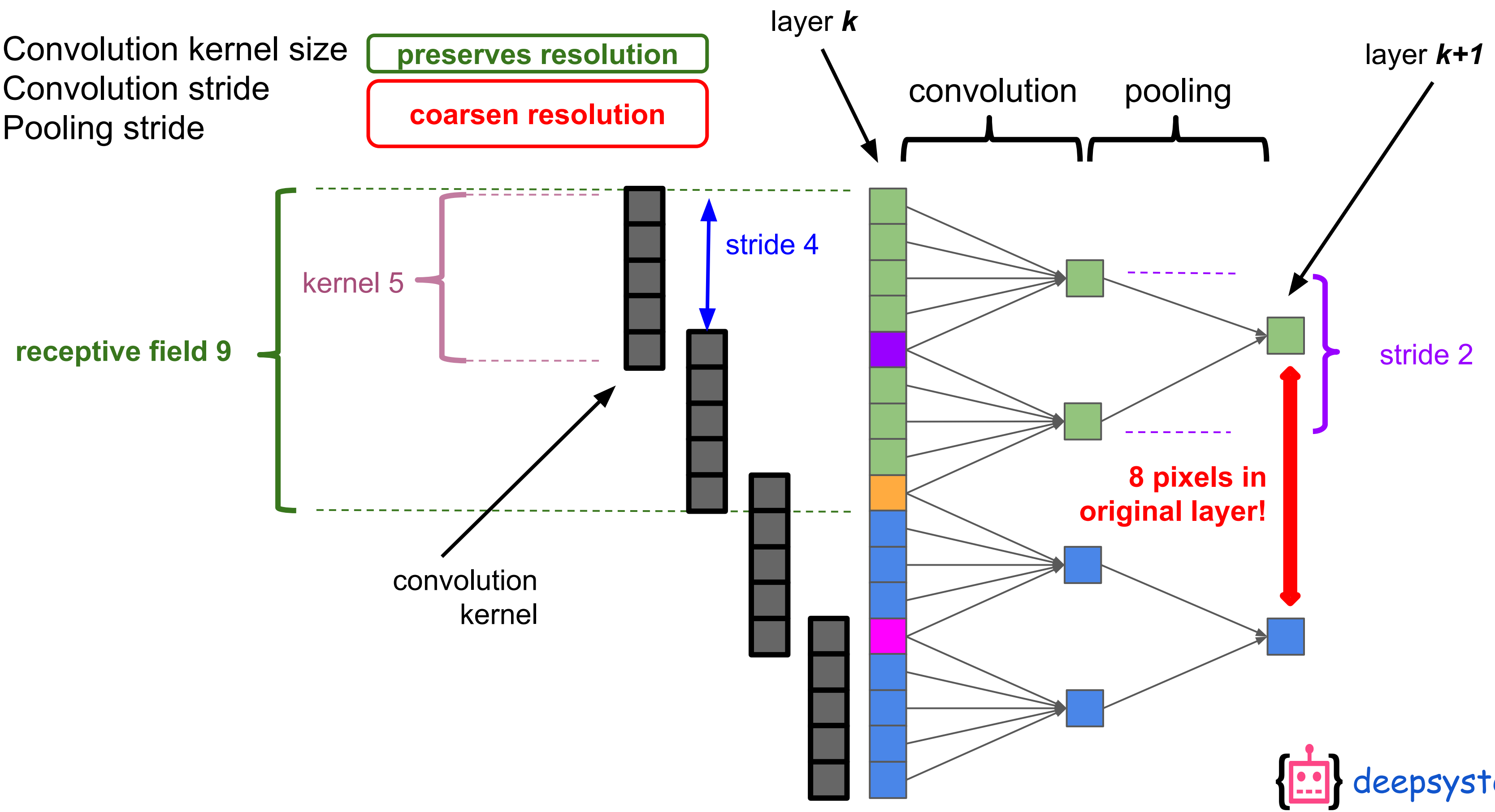
- Recap: Convolutional networks for semantic segmentation - pros and cons
  - Focus on representing spatial info

- Trick #1: **dilated convolutions**
  - Widen receptive field effectively
  - Avoid spatial resolution coarsening

- Trick #2: **conditional random field** for segmentation post-processing
  - Extra smoothing for better local consistency
  - Align segment boundaries with sharp changes in the image

Receptive Field Factors

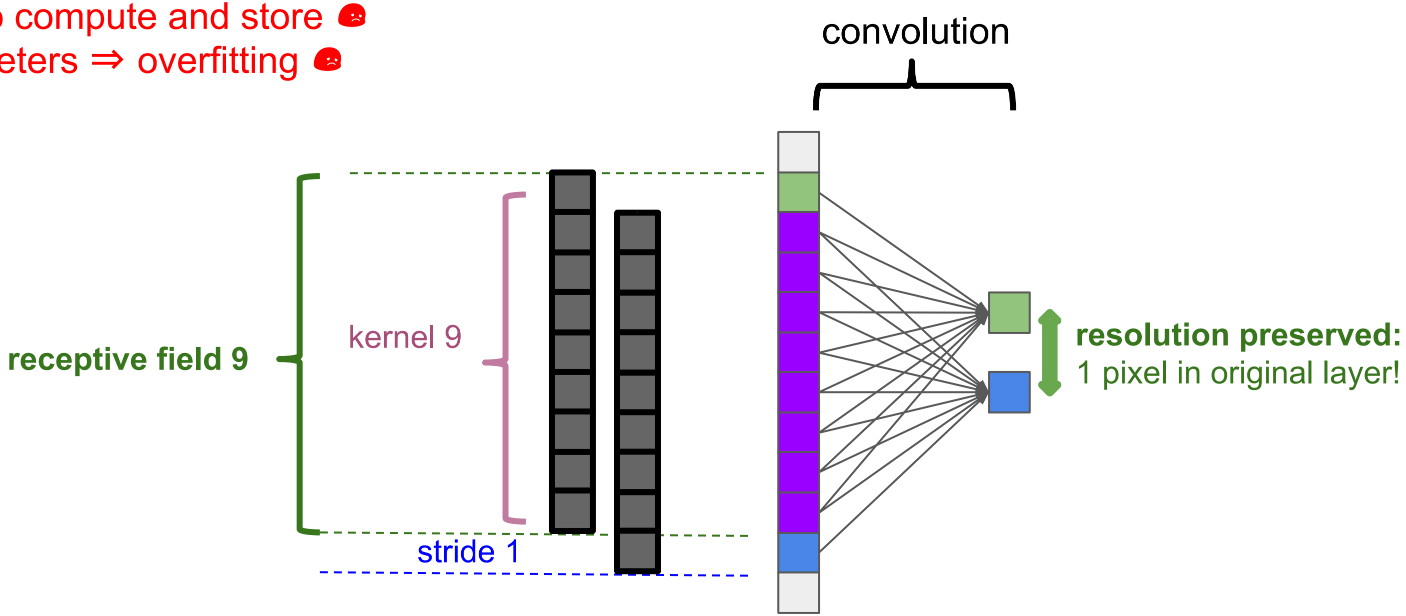
- Convolution kernel size
- Convolution stride
- Pooling stride



deepsystems.io

Preserving resolution - convolution-only net

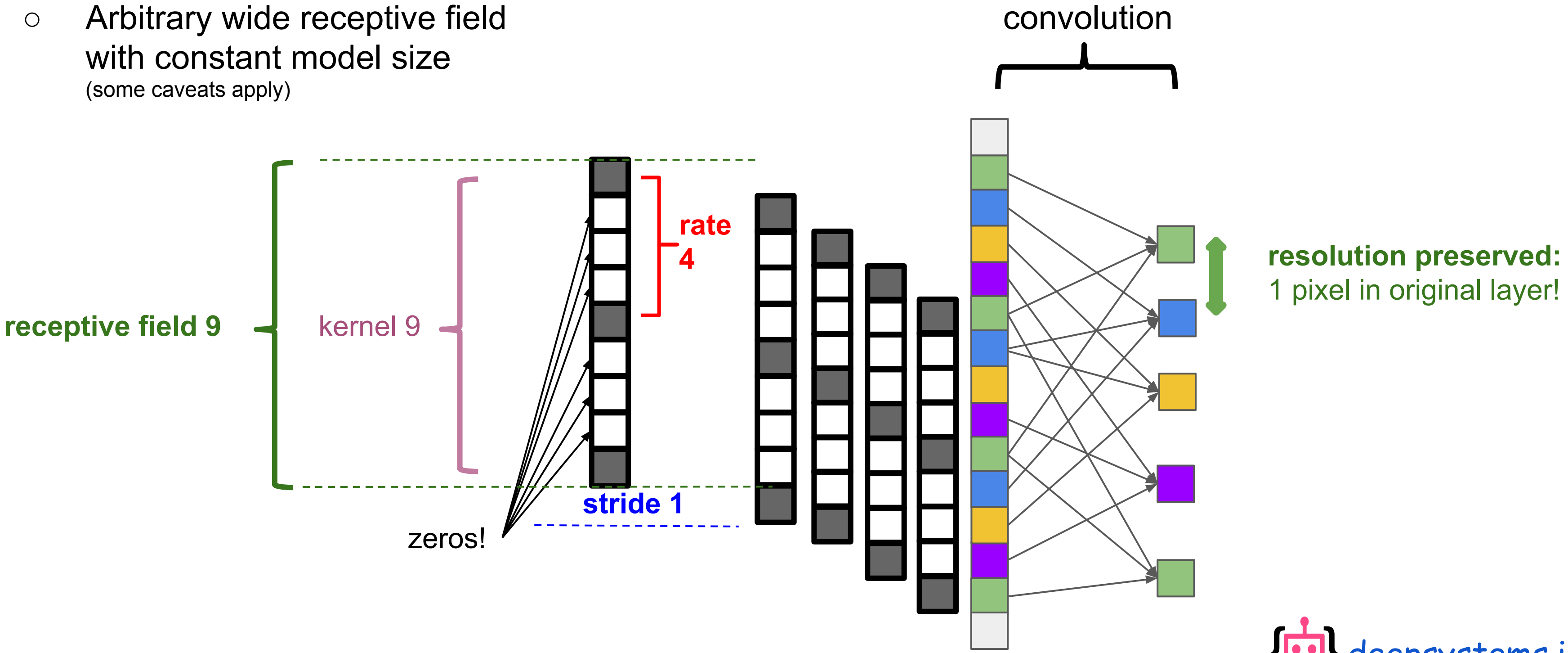
- Remove pooling
- Convolution stride = 1
- **Large convolution kernel** to increase receptive field
    - expensive to compute and store 😞
    - more parameters ⇒ overfitting 😞



deepsystems.io

Trick #1 - Dilated convolution

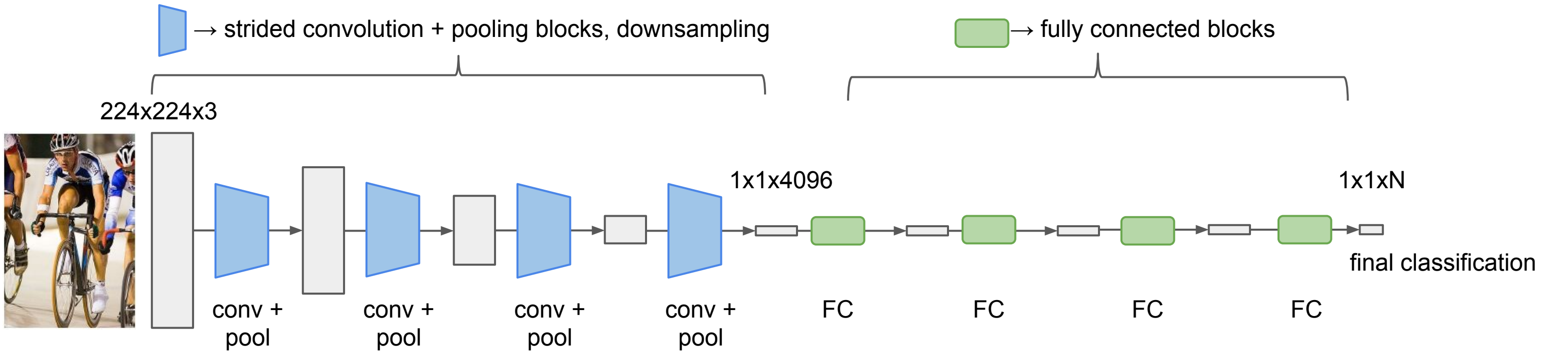
- Large, but **sparse** convolution kernel
- Kernel **rate** - step between nonzeros
  - Computation complexity ~ number of nonzeros
  - Arbitrary wide receptive field with constant model size
    (some caveats apply)



deepsystems.io

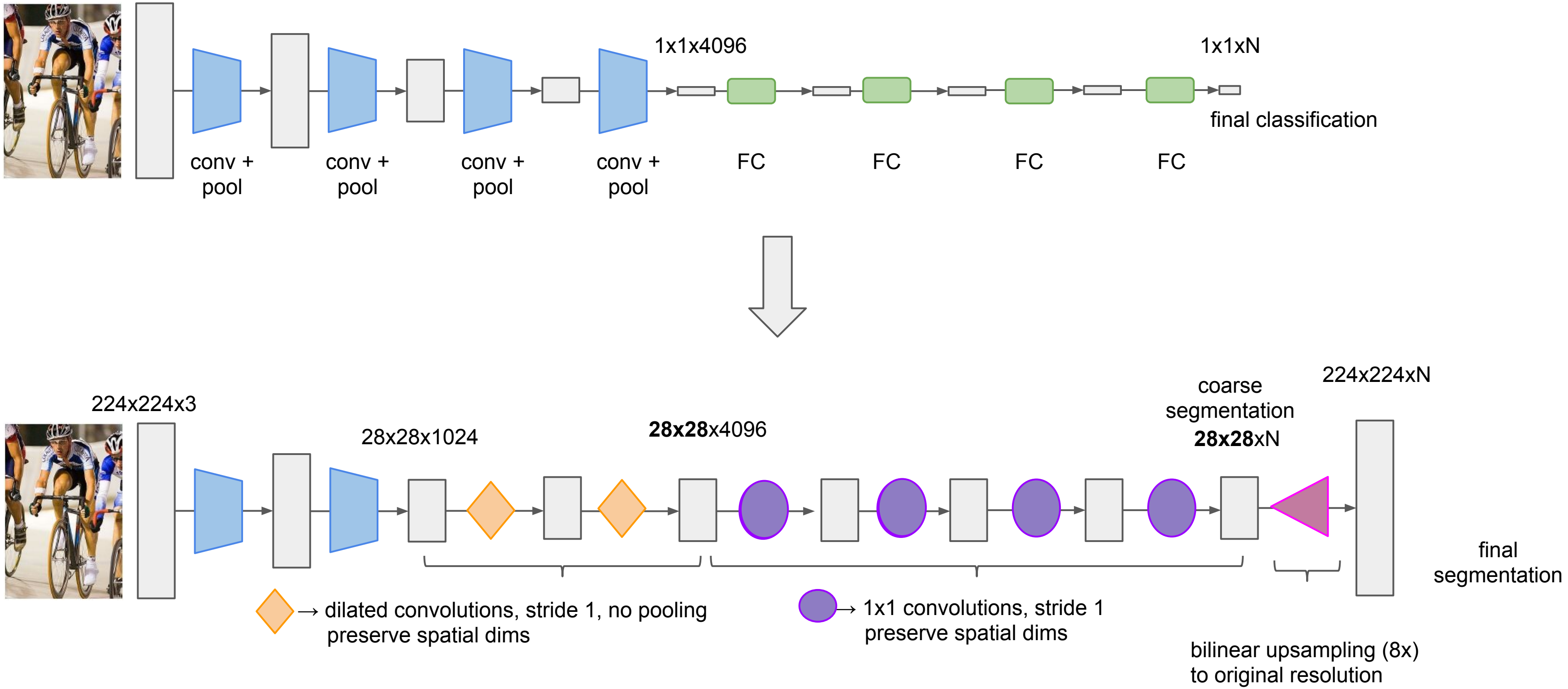
Classification → Segmentation with Dilated Convolutions

- Start with a convnet for classification (e.g. VGG-16)



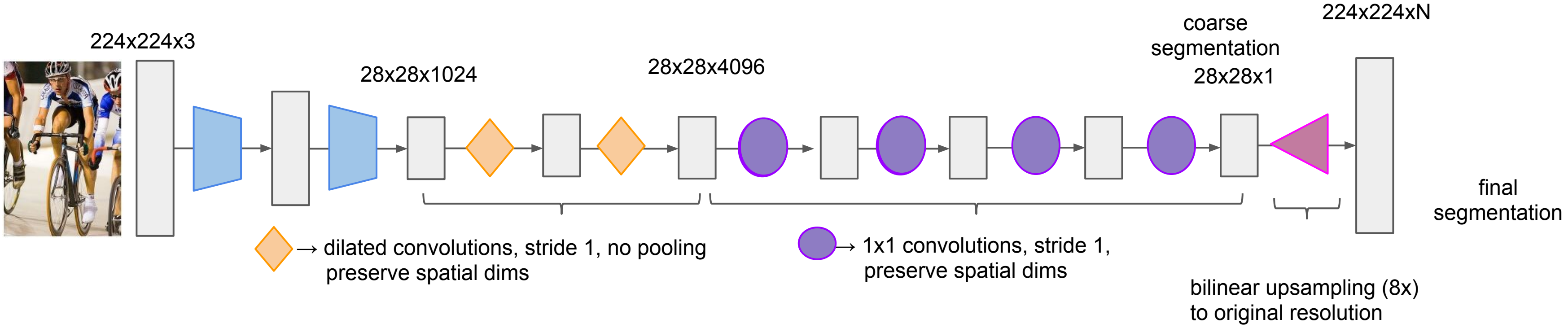
Classification → Segmentation with Dilated Convolutions

- Start with a convnet for classification (e.g. VGG-16)



deepsystems.io

Classification → Segmentation with Dilated Convolutions



- Fully connected stages → *replace* by convolutions with 1x1 spatial kernel
- Last several convolution+pooling blocks → *replace* by dilated convolutions, stride 1, no pooling
  - Preserves spatial dimensions
- Bilinear upsampling in the end to original resolution
  - Relatively small upsampling factor, not much need for learned upsampling schemes

deepsystems.io

Outline

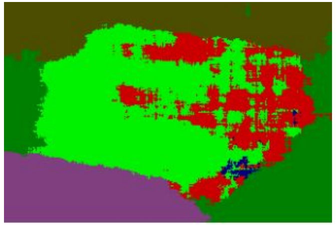
- Recap: Convolutional networks for semantic segmentation - pros and cons
    - Focus on representing spatial info
- Trick #1: **dilated convolutions**
    - Widen receptive field effectively
    - Avoid spatial resolution coarsening
- Trick #2: **conditional random field** for segmentation post-processing
    - Extra smoothing for better local consistency
    - Align segment boundaries with sharp changes in the image

Why Postprocessing?

- Convnet does not (explicitly) encode common sense segmentation properties:
  - Nearby pixels are likely to have the same class (smoothness)
  - Segment boundaries typically correspond to sharp color changes in image
- Postprocessing: tweak convnet segmentation to explicitly enforce **smoothness** and segment **edge alignment with** the **underlying image**.

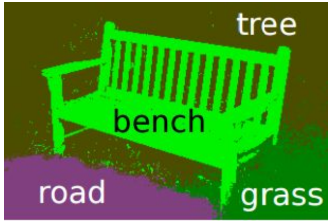
original

+



convnet segmentation
(bad quality exaggerated)

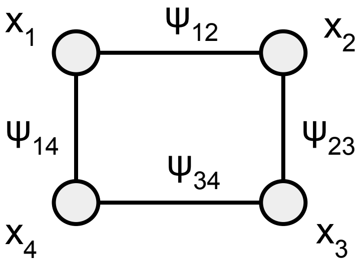
postprocessing



final

Recap: Conditional Random Fields

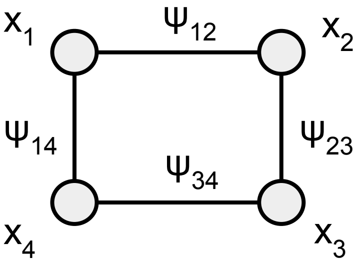
- Represent (unnormalized) high-dimensional probability distribution as a product of low-dimensional potentials:



$$P(x_1x_2x_3x_4) \propto \prod_i \psi_i(x_i) \prod_{ij \in G} \psi_{ij}(x_ix_j)$$

- Potentials exist only over variables *directly connected* by graph edges
- Edges indicate *direct* dependencies
- Exponentially fewer parameters: $D^4 \Rightarrow 4D + 4D^2$ in above example
- Exact inference (e.g. $P(x_1)$, argmax $P(x_1)$) still intractable
  - Approximate iterative methods often work well in practice

CRF Potentials: Potts Model



$$P(x_1x_2x_3x_4) \propto \underbrace{\prod_i \psi_i(x_i)}_{\text{single-pixel class beliefs from convnet}} \; \underbrace{\prod_{ij \in G} \psi_{ij}(x_ix_j)}_{\text{smoothness + segment edge alignment with in-image edges}}$$

Difference between pixels colors.
Pixels with similar colors influence each other more.
Encodes segment edge alignment with in-image edges.

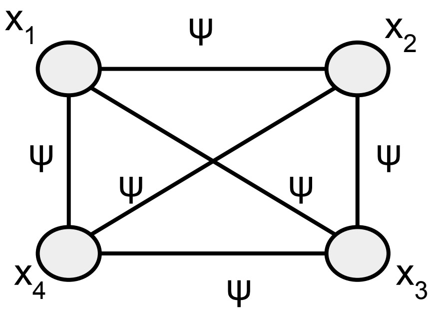
Potts model: $\psi_{ij}(x_ix_j) = \exp[\, - \underbrace{I(x_i \neq x_j)}\, \bullet \, \exp(\underbrace{-a^2|p_i - p_j|^2} \overbrace{- b^2|C_i - C_j|^2}) \,]$

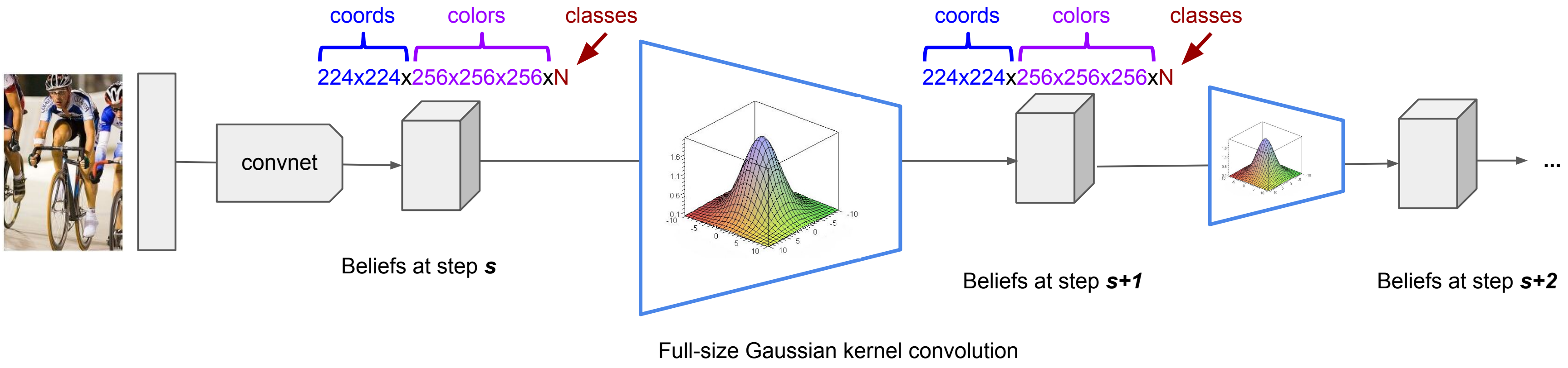
Indicator:
1 if $x_i \neq x_j$
0 if $x_i = x_j$

Geometrical distance between two pixels in the image.
Nearby pixels influence each other more.
Encodes labels spatial smoothness.

Inference (super high level)

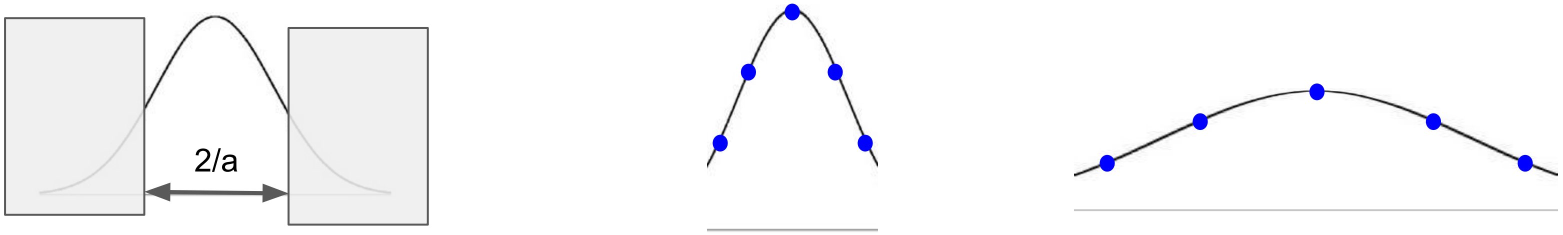
- Fully connected graph: an edge for every pair of pixels
- Potts model: $\psi(x_i x_j) = \exp[\, -I(x_i \neq x_j) \bullet \underbrace{\exp(\, -a^2|p_i - p_j|^2 - b^2|C_i - C_j|^2)}_{\text{Looks like a Gaussian...}} \,]$
- Approximate inference update is a convolution with Gaussian kernel in (coordinates x colors x beliefs) space [see paper]
    - Kernel size == full image size!





Inference (super high level)

- Approximate inference update is a convolution with Gaussian kernel in (coordinates x colors x beliefs) space [see paper]
- Exact update is still too expensive: each pixel depends on the whole image
    - O(|pixels|^2) complexity
- But **Gaussian kernel** admits an efficient approximation!
    - Ignore pixels more than ~1/a in distance away
        - 95% of Gaussian probability mass is within 2 standard deviations
    - Subsample the rest at rate C/a (enough by the sampling theorem)

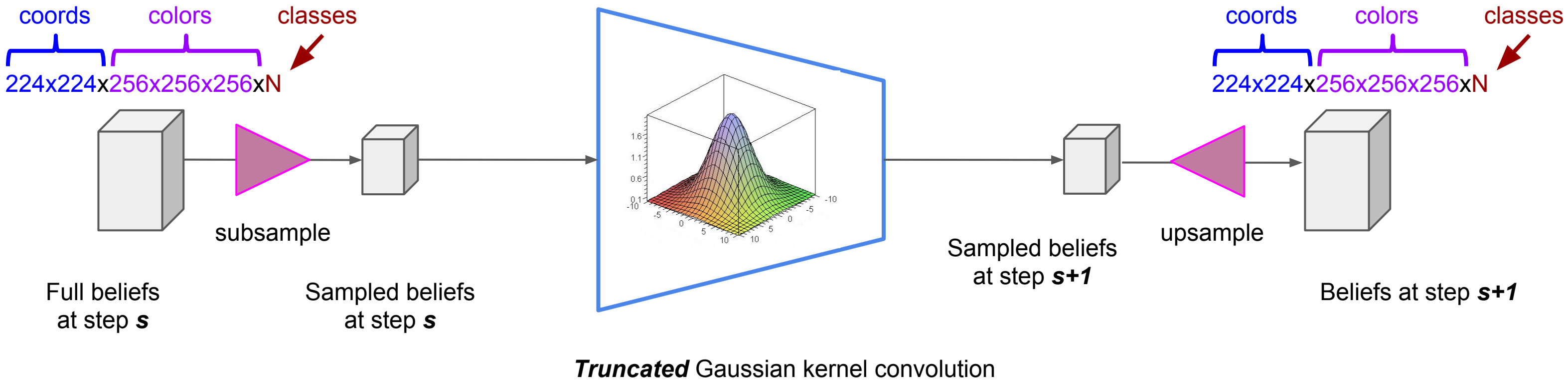


- *O(1)* per pixel update complexity *regardless of Potts model parameters **a, b***
- O(|pixels|) whole image update complexity

Inference (super high level)

- Potts model: $\psi(x_i x_j) = \exp[\, -I(x_i \neq x_j) \bullet \exp(\, -a^2|p_i - p_j|^2 - b^2|C_i - C_j|^2) \,]$

- Constant-time per-pixel inference with subsampling and truncated Gaussian convolution



Conclusions

- Typical convnets discard most spatial information about the image
  - Great for classification, but problematic for segmentation

- **Dilated convolutions** help achieve a wide receptive field
  - without coarsening spatial resolution
  - efficiently both computationally and statistically

- A template for adapting any classification convnet for segmentation
  - Fully connected layers → 1x1 convolutions
  - Last several conv+pool blocks → dilated convolutions with stride 1
  - Examples used: VGG-16, ResNet-101

- **Conditional random field** for segmentation post-processing
  - Extra smoothing for better local consistency
  - Align segment boundaries with sharp changes in the image
  - Efficient approximate inference as convolutions in [coordinates x color] space

Links

- Paper: https://arxiv.org/abs/1606.00915
- Code: https://bitbucket.org/aquariusjay/deeplab-public-ver2

Thank you!

deepsystems.io

inbox@deepsystems.ru

Our team is looking for business partners to make exciting deep learning solutions.