Neural Networks and Deep Learning

Convolutional Networks II - Applications

Nicholas Dronen

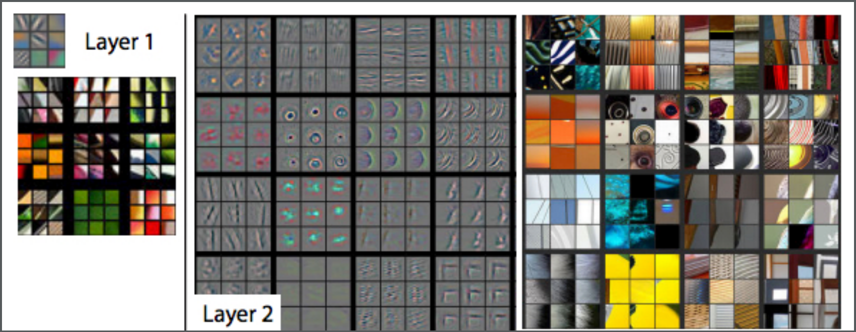
Department of Computer Science
dronen@colorado.edu

February 13, 2019

University of Colorado Boulder

Zeiler and Fergus, 2013

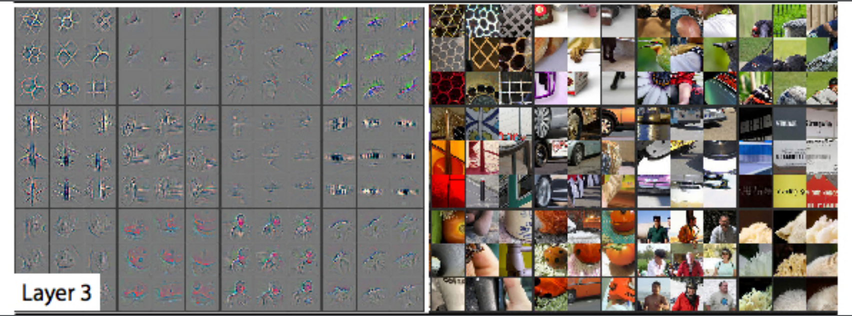
Visualizing and Understanding Convolutional Networks



Visualization of features in a fully trained model, top 9 activations in a random subset of feature maps across the validation data, projected down to pixel space using deconvolutional network approach.

Zeiler and Fergus, 2013

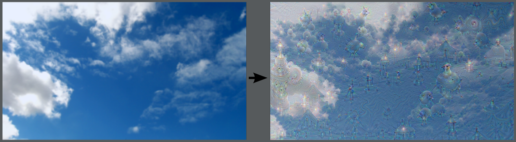
Visualizing and Understanding Convolutional Networks



Visualization of features in a fully trained model.

Mordvintsev, Olah, and Tyka (2015)

Deep Dream



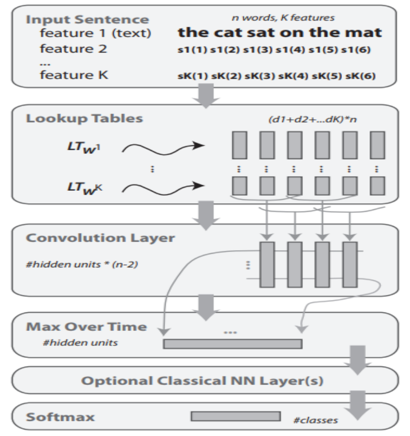
Results produced by a CNN trained on animals.

Feed a random input to a ConvNet, choose which class(es) the network should believe the input contains, and backpropagate the error *to the input*. Eventually, under constraints, the input takes on surreal qualities.

Collobert and Weston, 2008 (!)

A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning

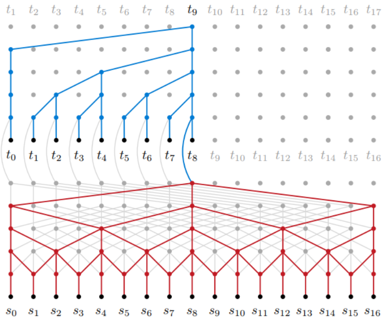
- First layer contains a matrix of word *embeddings* (authors call this a look-up table)
- Multi-task learning of POS tagging, chunking, NER, language modeling, and semantic role labeling.



ByteNet, 2016

Neural Machine Translation in Linear Time

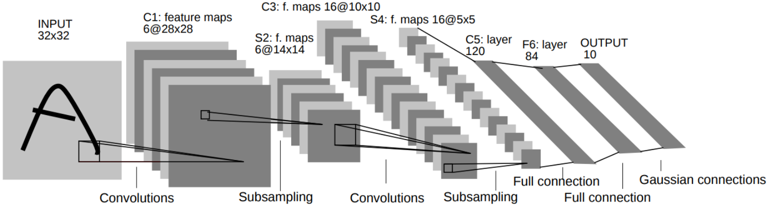
- An encoder-decoder ConvNet with dilated convolutions.
- To address the differing lengths of the source and the target, the decoder is dynamically unfolded over the representation of the encoder.



ByteNet architecture. The target decoder (blue) is stacked on top of the source encoder (red).

LeNet (1998)

Gradient-Based Learning Applied to Document Recognition

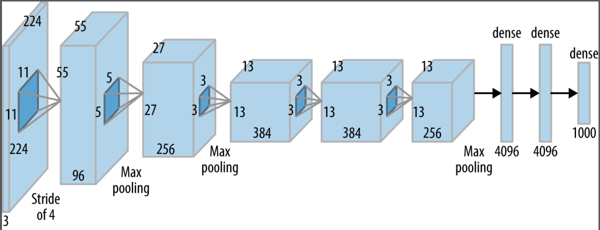


LeNet-5 architecture

- An early ConvNet – note the two convolution layers and three fully-connected hidden layers.
- How many output filters at each convolutional layer?
- What is the effective receptive field of the fully-connected layers?

AlexNet (2012)

ImageNet Classification with Deep Convolutional Neural Networks



Architecture of the AlexNet

- Won ILSVRC-2012 w/15.3% top-5 error rate (2nd place, 26.2%)
- Five convolutional layers, three fully-connected layers
- ReLU, local response normalization (deprecated)
- What are the filter sizes in each convolutional layer?
- Dropout to regularize fully-connected layers.
- Early use of GPU for training (see Krizhevsky's cuda-convnet2)

VGGNet (2014)

Very Deep Convolutional Networks for Large-Scale Image Recognition

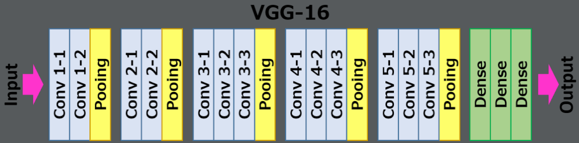


Image source: https://neurohive.io/en/popular-networks/vgg16/

- First and second places in the localisation and classification tasks, respectively, in ILSVRC-2014.
- Nineteen convolutional layers, three fully-connected layers
- Filters are 3×3 throughout network

SPP-net (2014)

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition

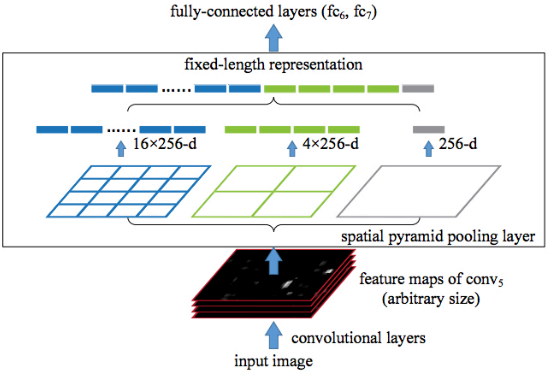
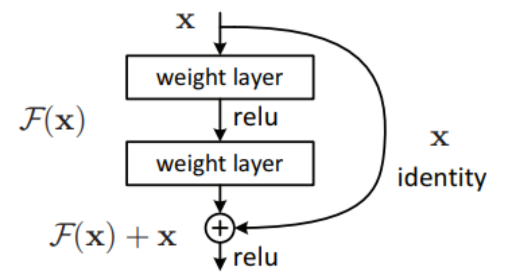


Figure 3: A network structure with a **spatial pyramid pooling layer**. Here 256 is the filter number of the $conv_5$ layer, and $conv_5$ is the last convolutional layer.

ResNet (2015)

Deep Residual Learning for Image Recognition

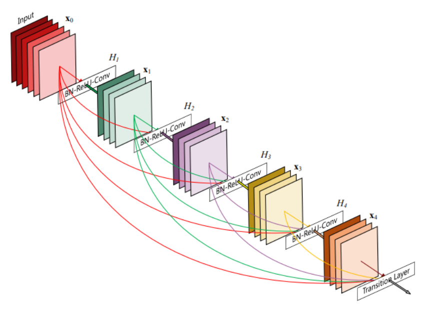


Residual learning: a building block

- First place on the ILSVRC 2015 classification task.
- Employs "skip connections" – like an identity function
- What does this do to feature maps from lower layers?
- Why might this help the model perform the task better?

DenseNet (2016)

Densely Connected Convolutional Networks

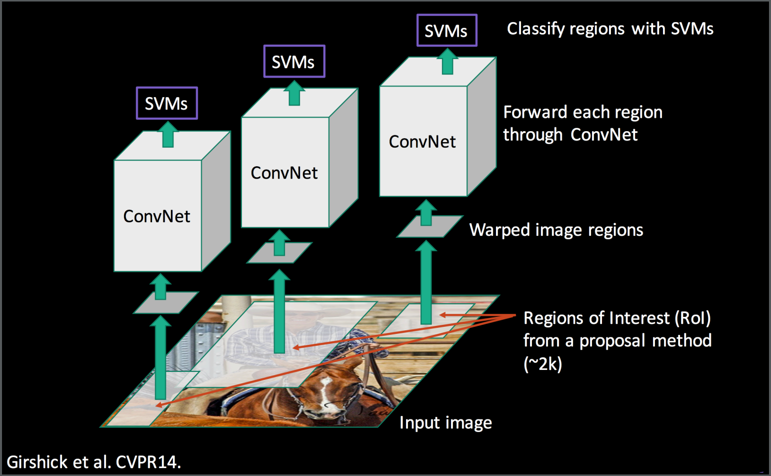


A 5-layer dense block, each layer takes all preceding feature-maps as input.

- Fewer parameters for the same level of accuracy
- All layers connected to all downstream layers
- Claim: gradient flow is improved. True?

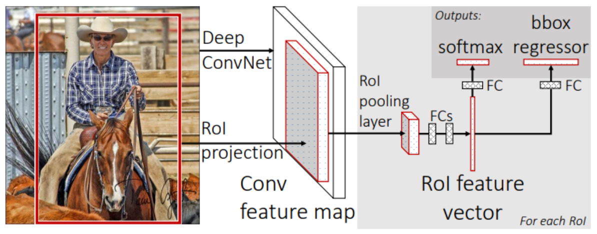
R-CNN (2014)

Rich feature hierarchies for accurate object detection and semantic segmentation



Fast R-CNN (2015, April)

Fast R-CNN



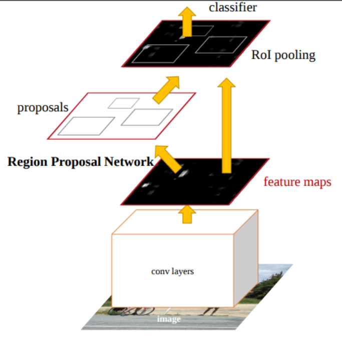
Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Region proposals come from a separate algorithm (e.g. selective search). Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers. The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets.

- R-CNN is slow ($\sim$ 50 seconds/image)
- Single-stage model learns jointly to classify object proposals and refine their spatial locations.
- Still uses an external region proposal algorithm.

Faster R-CNN (2015, June)

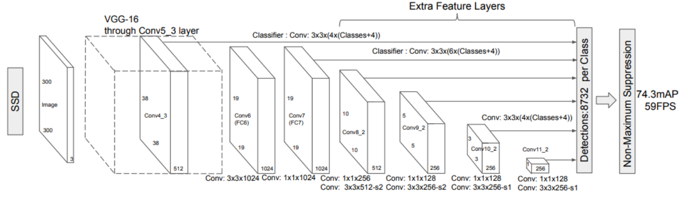
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

- Instead of using external region proposal algorithm, add a region proposal network (RPN)
- RPN consists of additional convolutional layers that regress region bounds as well as *objectness* scores at each grid location.



SSD (2015)

SSD: Single-Shot Multibox Detector

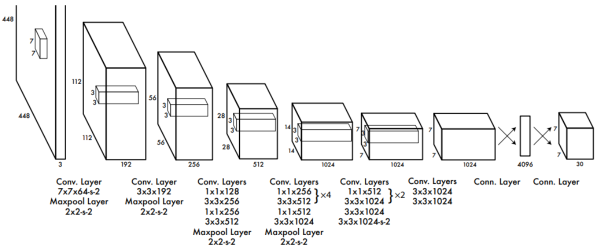


SSD Architecture.

- Object detection systems are variants of the following approach: hypothesize bounding boxes, resample pixels or features for each box, and apply a high quality classifier.
- SSD is simple relative to methods that require object proposals because it completely eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates all computation in a single network.

YOLO (2015)

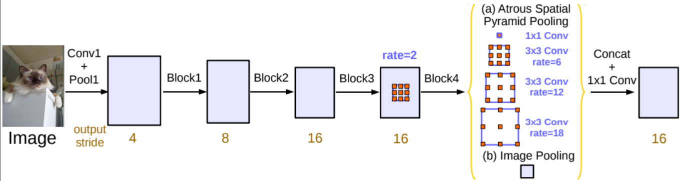
You Only Look Once: Unified, Real-Time Object Detection



SSD Architecture.

- Prior work on object detection re-purposes classifiers to perform detection. Here, object detection is framed as a regression problem to spatially separated bounding boxes and associated class probabilities.
- A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation.

DeepLab (2014-2017)

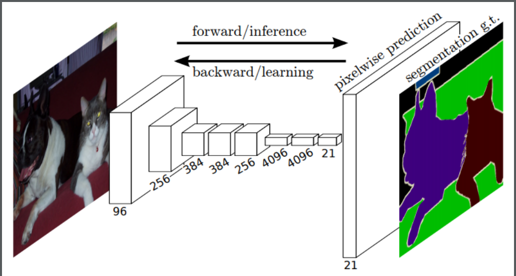


Parallel modules with atrous convolution (ASPP), augmented with image-level features

- v1: Atrous Convolutions to increase receptive field, Post-processing using Conditional Random Fields (CRFs) to overcome poor localization.
- v2: Atrous spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales.
- v3: Improves on ASPP and Improves over previous DeepLab versions without DenseCRF post-processing

Fully-Convolutional Network (FCN) (2015)

Fully Convolutional Networks for Semantic Segmentation



Fully-Convolutional Network (FCN) (2015)

Fully Convolutional Networks for Semantic Segmentation

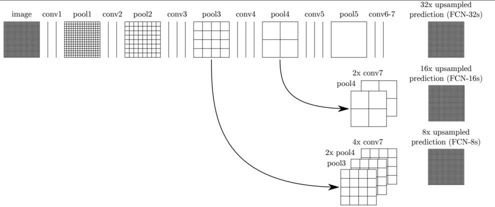
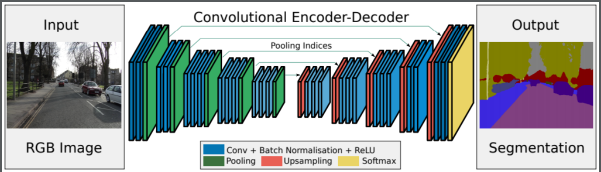


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Pooling and prediction layers are shown as grids that reveal relative spatial coarseness, while intermediate layers are shown as vertical lines. First row (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Second row (FCN-16s): Combining predictions from both the final layer and the pool4 layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Third row (FCN-8s): Additional predictions from pool3, at stride 8, provide further precision.

SegNet (2015)

SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation

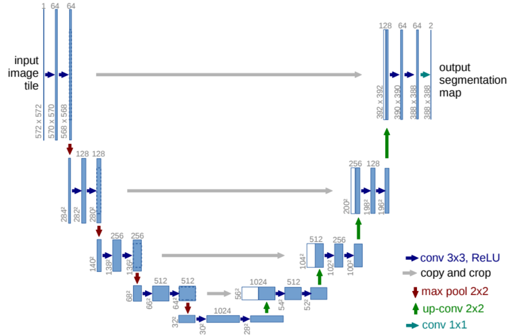


SegNet architecture.

- There are no fully connected layers and hence it is only convolutional. A decoder upsamples its input using the transferred pool indices from its encoder to produce a sparse feature map(s).
- It then performs convolution with a trainable filter bank to densify the feature map. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification

UNet (2015)

U-Net: Convolutional Networks for Biomedical Image Segmentation

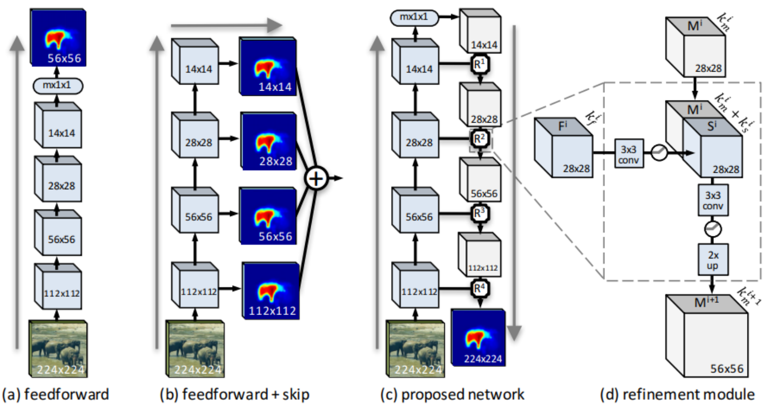


UNet architecture.

- Encoder-decoder architecture with lateral connections from each encoder module to the corresponding decoder module.

SharpMask (2016)

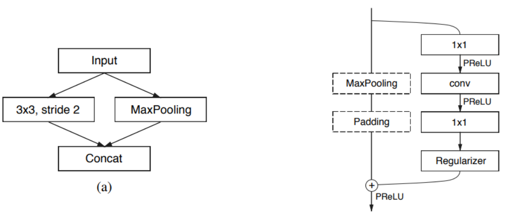
Learning to Refine Object Segments



(a) Feedforward nets predict masks using only upper-layer features, resulting in coarse pixel masks. (b) Common 'skip' architectures are equivalent to making independent predictions from each layer and averaging the results such an approach is not well suited for object instance segmentation. (c,d) Augmenting feedforward nets with a top-down refinement approach. The resulting bottom-up/top-down architecture is capable of efficiently generating high-fidelity object masks.

ENet (2016)

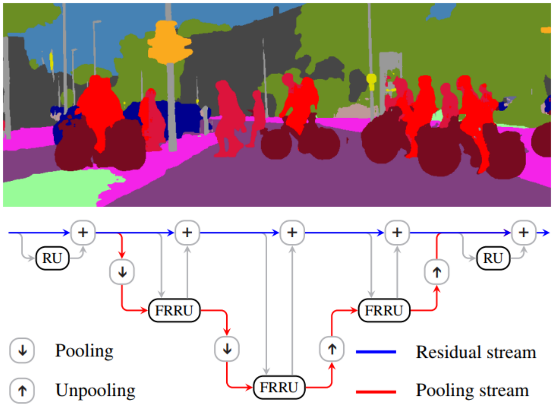
ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation



- ENet, is created specifically for tasks requiring low latency operation. ENet is up to 18×faster, requires 75× less FLOPs, has 79× less parameters, and provides similar or better accuracy to existing models.

Full-Resolution Residual Networks (FRRN) (2017)

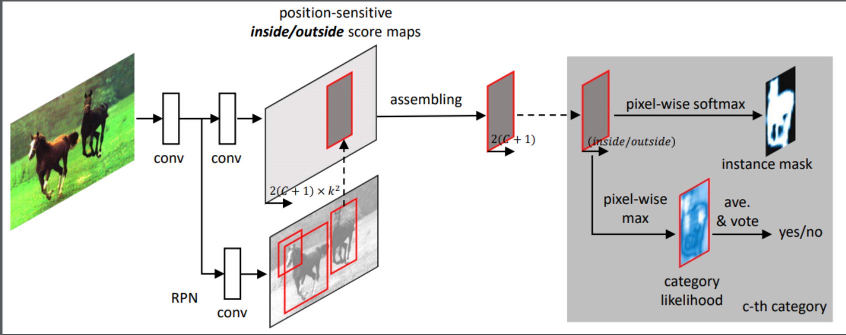
Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes



Abstract architecture of FRRN, the network has 2 processing streams. The residual stream (blue) stays at the full image resolution, the pooling stream (red) undergoes a sequence of pooling and unpooling operations. The 2 processing streams are coupled using full-resolution residual units.

Instance-Aware FCN (2017)

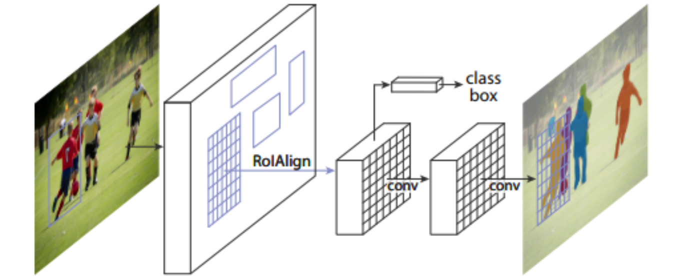
Fully Convolutional Instance-Aware Semantic Segmentation



Overall architecture of FCIS. A region proposal network (RPN) shares the convolutional feature maps with FCIS. The proposed region-of-interests (ROIs) are applied on the score maps for joint object segmentation and detection. The learnable weight layers are fully convolutional and computed on the whole image. The per-ROI computation cost is negligible.

Mask R-CNN (2017)

Mask R-CNN



The Mask R-CNN framework for instance segmentation.

- Extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition.

PoseNet

PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization

Figure 1: **PoseNet: Convolutional neural network monocular camera relocalization.** Relocalization results for an input image (top), the predicted camera pose of a visual reconstruction (middle), shown again overlaid in red on the original image (bottom). Our system relocalizes to within approximately $2m$ and 3° for large outdoor scenes spanning $50,000m^2$. For an online demonstration, please see our project webpage: `mi.eng.cam.ac.uk/projects/relocalisation/`