CPSC 540: Machine Learning
Fully-Convolutional Networks

Mark Schmidt

University of British Columbia

Winter 2019
Last Time: “Pipeline” Approach to Object Localization

- Early approach (region CNN):
  1. Propose a bunch of potential boxes.
  2. Compute features of box using a CNN.
  3. Classify each box based on an SVM.
  4. Refine each box using linear regression.

R-CNN: Regions with CNN features

Improved on state of the art, but not very elegant with its 4 steps.
Region Convolutional Neural Networks: “End to End” Approach

- Modern approaches **try to do the whole task with one neural network**.
  - The network extracts features, proposes boxes, and classifies boxes.

This is called an **end-to-end** model.
End-to-End Computer Vision Models

- Key ideas behind end-to-end systems:
  1. Write each step as a differentiable operator.
  2. Train all steps using backpropagation and stochastic gradient.

- There now exist end-to-end models for all the standard vision tasks.
  - Depth estimation, pose estimation, optical flow, tracking, 3D geometry, and so on.
  - A bit hard to track the progress at the moment.
  - A survey of $\approx 200$ papers from 2016:
    - http://www.themtank.org/a-year-in-computer-vision

- Let’s focus on the task of pixel labeling...
Straightforward CNN Extensions to Pixel Labeling

- Approach 1: apply an existing CNN to classify pixel given neighbourhood.
  - Misses long range dependencies in the image.
  - It’s slow: for 200 by 200 image, need to do forward propagation 40000 times.

- Approach 2: add per-pixel labels to final layer of an existing CNN.
  - Fully-connected layers lose spatial information.
  - Relies on having fixed-size images.
Fully-Convolutional Neural Networks

- Classic CNN architecture:

https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/image_segmentation.html
Fully-Convolutional Neural Networks

- **Fully-convolutional neural networks** (FCNs): CNNs with no fully-connected layers.
  - All layers maintain spatial information.

- Final layer upsamples to original image size.
  - With a learned “transposed convolution”.

- Parameter tying within convolutions allows images of different sizes.

https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/image_segmentation.html
Transposed Convolution Layer

- The upsampling layer is also called a **transposed convolution** or “deconvolution”.
  - Implemented as another convolution.

![Convolution vs Transposed Convolution](https://github.com/vdumoulin/conv_arithmetic)

- **Reasons for the names:**
  - “Tranposed” because sparsity pattern is transpose of a downsampling convolution.
  - “Deconvolution” is not related to the “deconvolution” in signal processing.
Fully-Convolutional Neural Networks

- FCNs quickly achieved state-of-the-art results on many tasks.

- FCN **end-to-end** solution is very elegant compared to previous “pipelines”:
  - No super-pixels, object proposals, merging results from multiple classifiers, and so on.
Variations on FCNs

- The transposed convolution at the last layer can lose a lot of resolution.
- One option is adding “skip” connections from earlier higher-resolution layers.

https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf
Variations on FCNs

Another approach to preserving resolution is deconvolutional networks:

https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/image_segmentation.html
Another way to address this is combining FCNs and CRFs.

DeepLab uses a fully-connected pairwise CRF on output layer. Though most recent version removed CRF.
Image Colourization

- An end-to-end image colorization network:

- Trained to reproduce colour of existing images after removing colour.

Image Colourization

- **Image colorization results:**

  [Images of colorized photos]
  
  - Colorado National Park, 1941
  - Textile Mill, June 1937
  - Berry Field, June 1909
  - Hamilton, 1936

- **Gallery:**
  [Link to gallery]
  http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/extra.html

- **Video:**
  [Link to video]
  https://www.youtube.com/watch?v=ys5nM04Q0iY
R-CNNs and Fully-Convolutional Networks

R-CNNs for Pixel Labeling

- An alternative approach: learn to apply binary mask to R-CNN results:

Where does data come from?

- Unfortunately, getting densely-labeled data is often hard.

- For pixel labeling and depth estimation, we explored getting data from GTA V:

- Easy to collect data at night, in fog, or in dangerous situations.
Where does data come from?

- Recent works use that you **don’t need full labeling**.  
  - Unobserved children in DAG don’t induce dependencies.
  - Although you would do better if you have an accurate dense labeling.

- Test object segmentation based on “single pixel” labels from training data:  
  - And some tricks to separate objects and remove false positives.

Show video...
Summary

- **End to end models**: use a neural network to do all steps.
  - Computer vision can now actually work!

- **Fully-convolutional networks**:
  - Elegant way to apply convolutional networks for dense labeling problems.