Cross-stitch Networks for Multi-task Learning

Ishan Misra* Abhinav Shrivastava* Abhinav Gupta Martial Hebert
Carnegie Mellon University

Overview

Goal

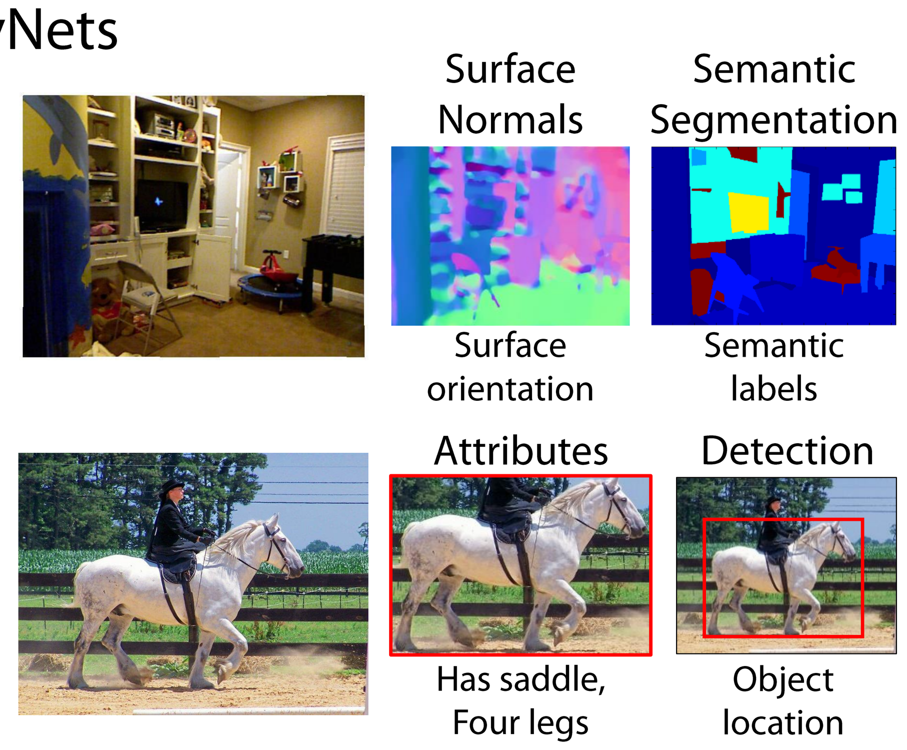
- Multi-task Learning with ConvNets
- No exhaustive search across architectures

Key Ideas

- Learn information sharing across tasks

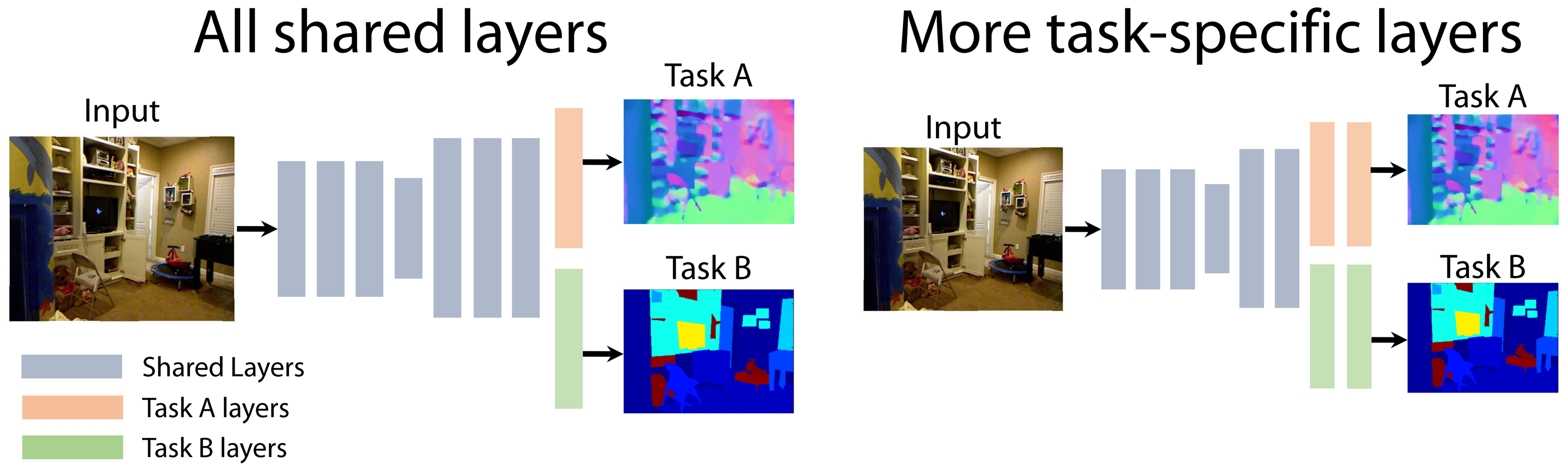
Outcome

- Explores exponentially more architectures in a principled manner
- Generalizes across tasks

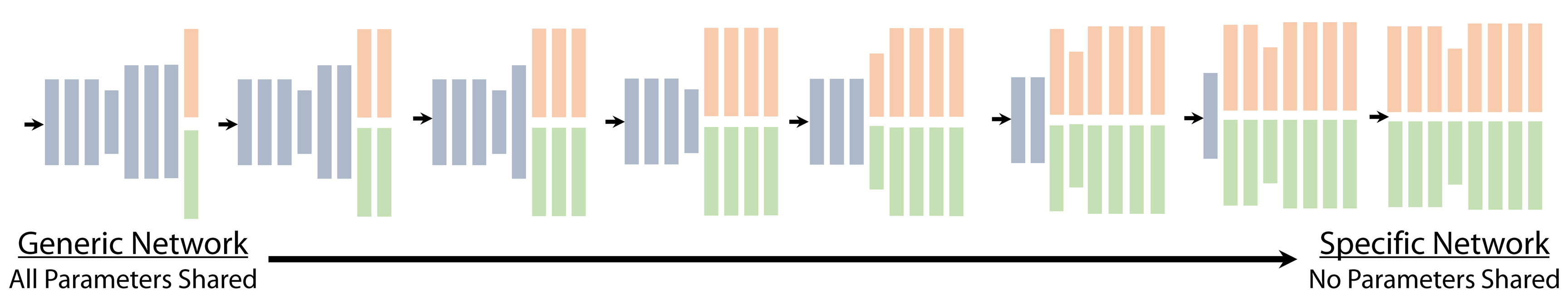


VOC08 [Everingham et al., 2008], [Farhadi et al, 2010], NYUv2 [Silberman et al., 2012], [Gupta et al., 2013]

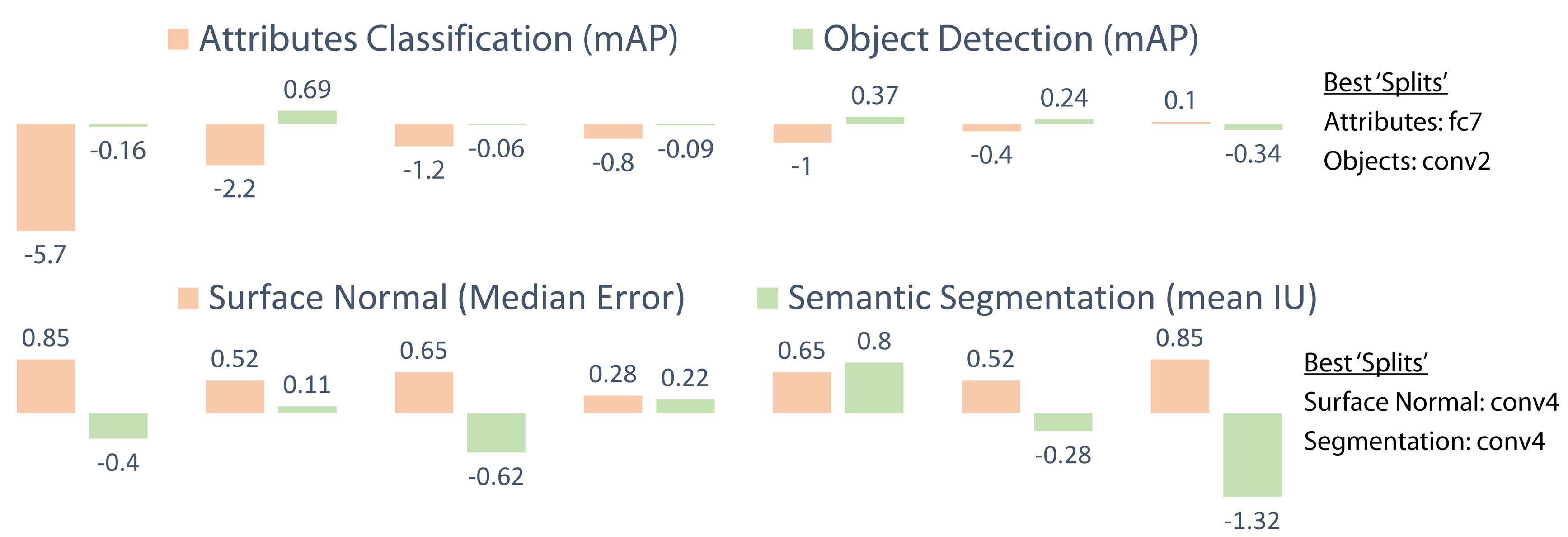
Standard Approach to Multi-task Learning



Spectrum of sharing: Exploring all 'split' Architectures



Relative performance wrt. Specific Network

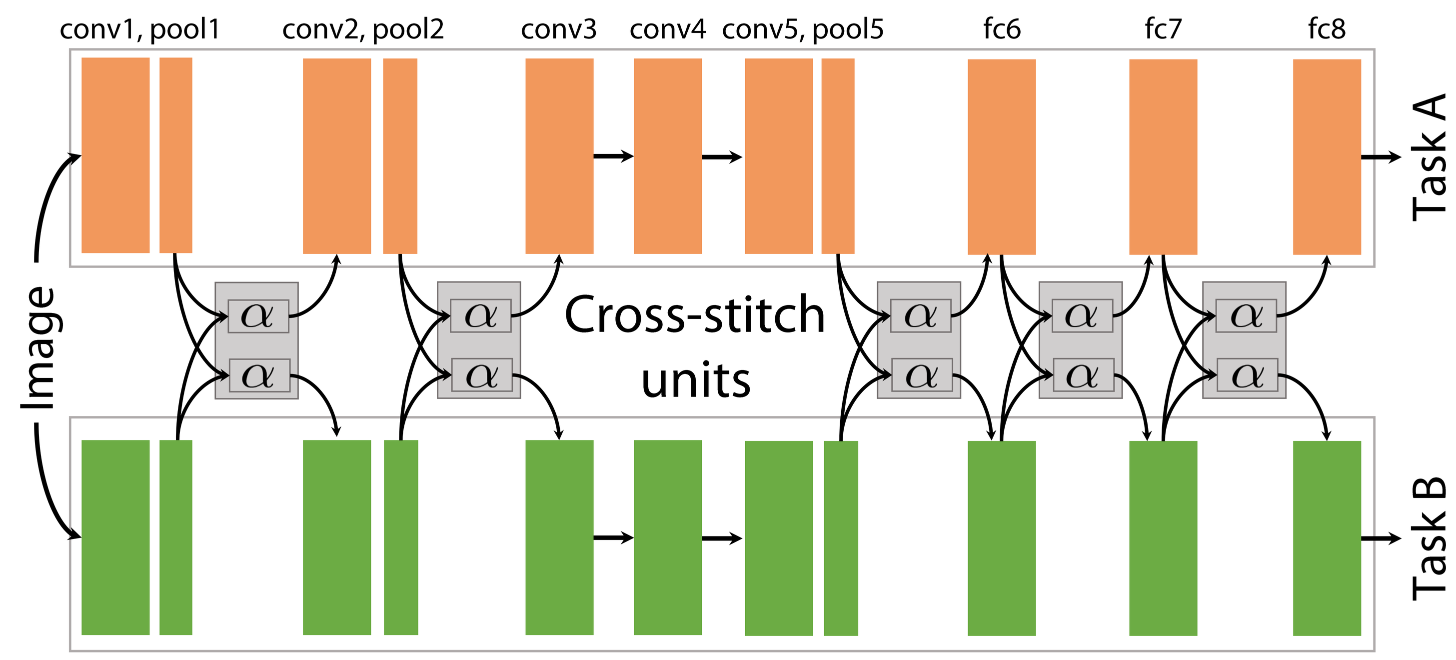


Problems and Limitations

- Practically Expensive
- No principled way of exploring architectures
- One architecture may not improve all tasks
- Does not generalize across tasks

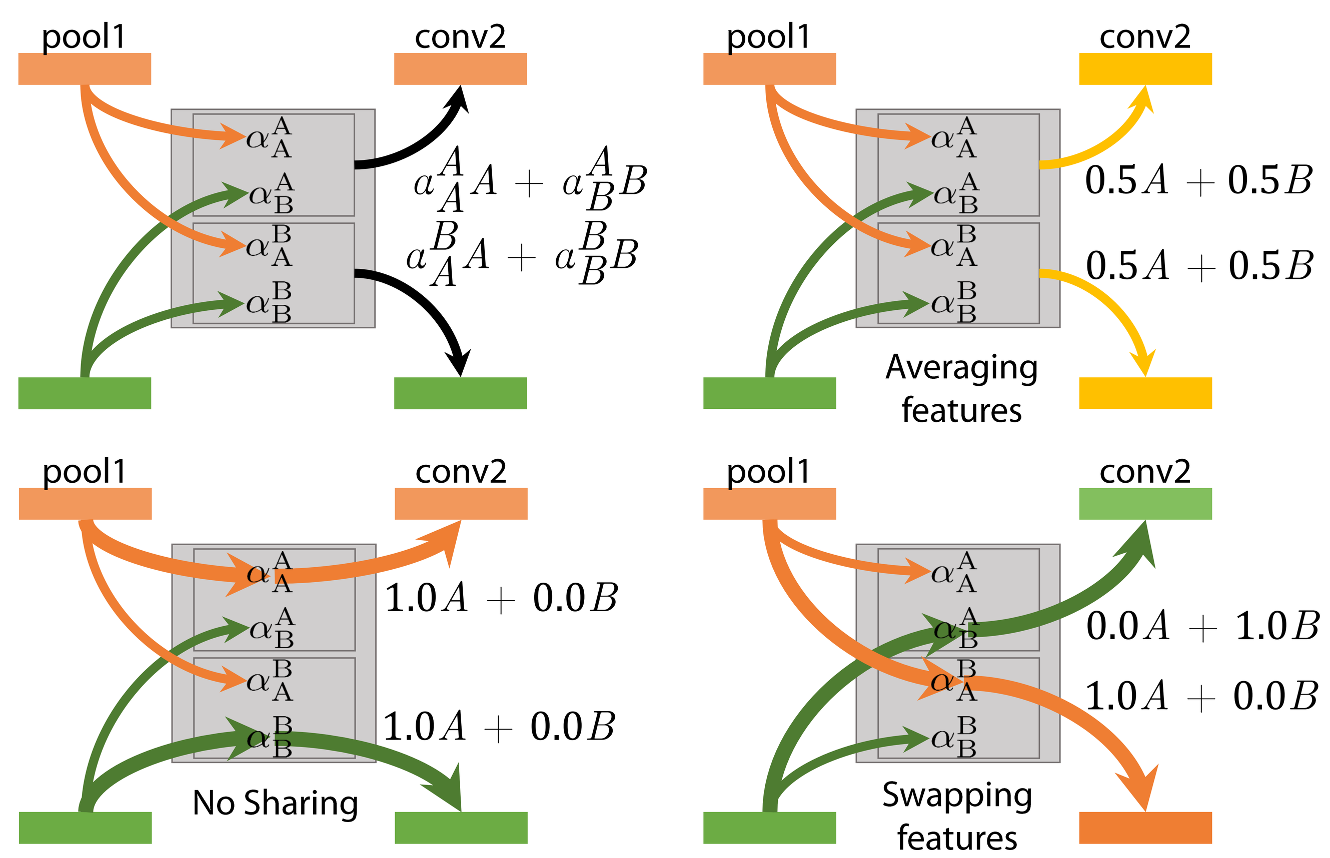
Approach

Cross-stitching



- Two networks trained independently
- Combine information across layers and networks
- Explores all split architectures and more
- Performs better than the best 'split' architecture

Cross-stitch unit: Learning to share



- Model information sharing as a linear combination
- Fuse information from multiple tasks
- Unit decides how much information to share

Results

Object Detection and Attributes on PASCAL VOC 2008

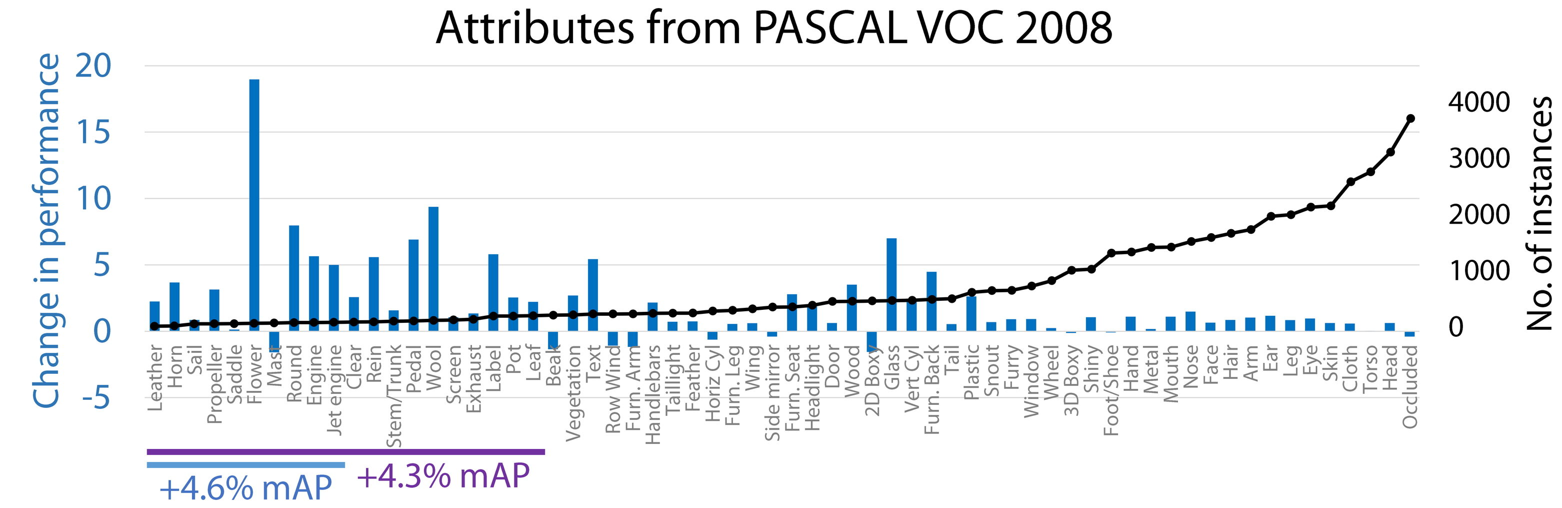
Method	Detection mAP	Attributes mAP
One-Task Network	44.9	60.9
MTL Baseline [Zhou et al., 2013]	42.7	54.1
Ensemble of two Networks	**46.1**	61.1
Split architecture (brute-force) [Attributes]	44.6	61.0
Split architecture (brute-force) [Detection]	44.8	59.7
Cross-stitch (Ours)	45.2	**63.0**

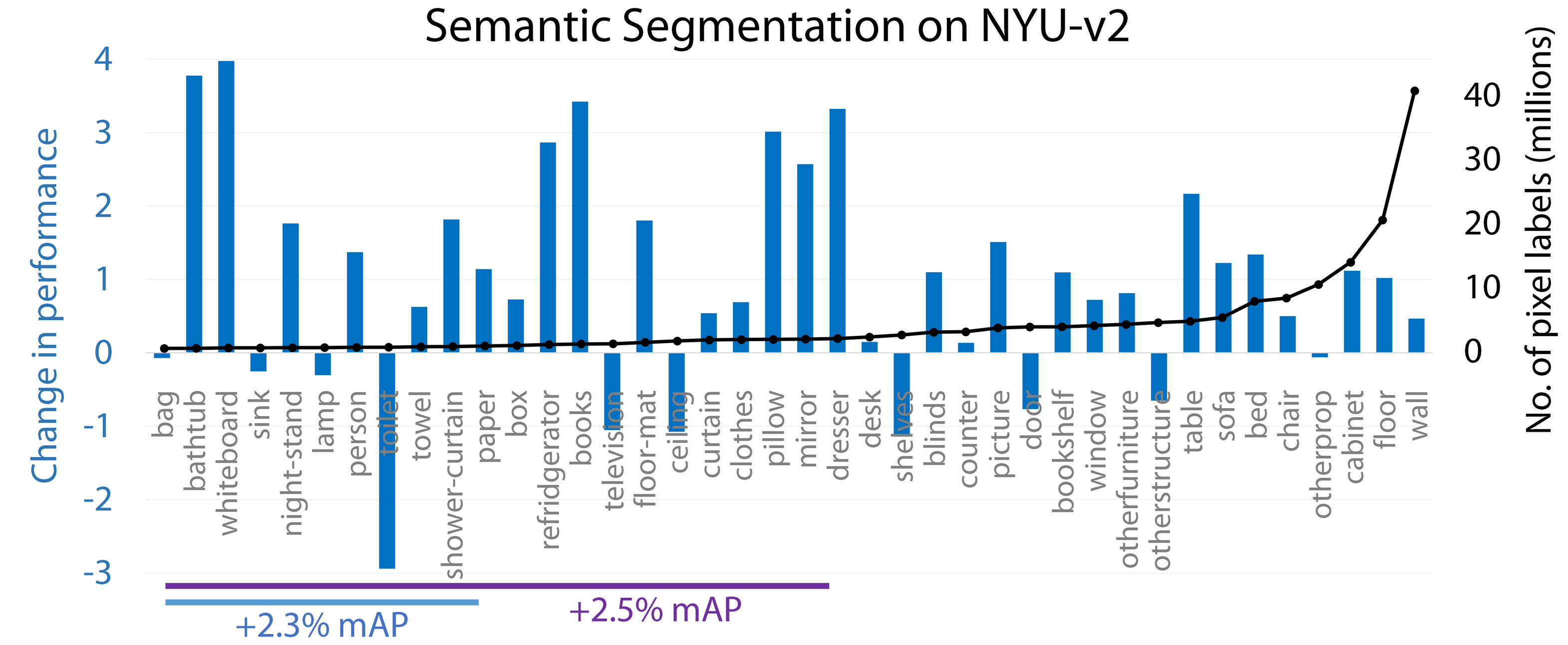
Surface Normal and Semantic Segmentation on NYUv2

Method	Surface Normal Median Error (lower is better)	Sem. Segmentation Mean IU (higher is better)
One-Task Network	19.0	18.4
MTL Baseline [Zhou et al., 2013]	18.9	16.6
Ensemble of two Networks	18.5	18.9
Split architecture (brute-force)	19.1	19.2
Cross-stitch (Ours)	**18.2**	**19.3**

Classes with less data

Sort classes according to no. of instances, and plot change in performance



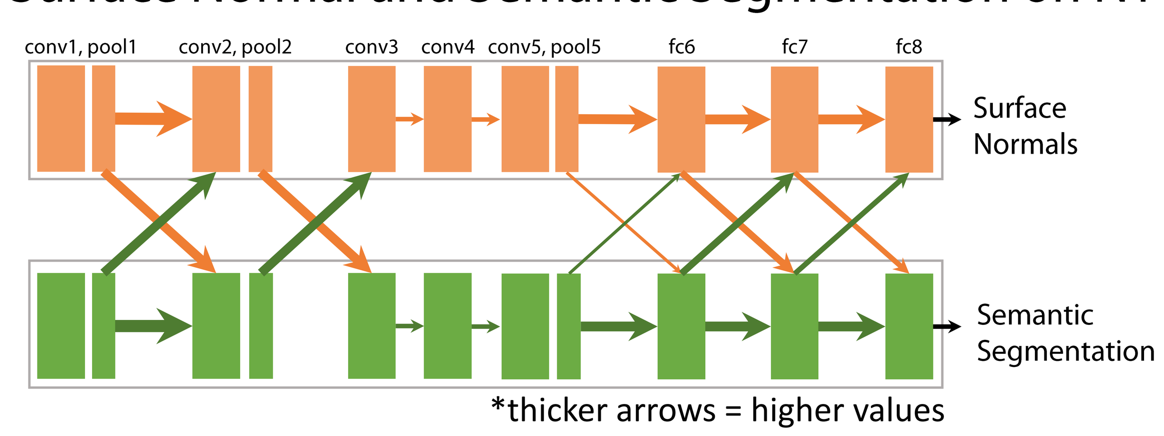


How to Learn?

- Initialize unit learning rates to be higher than layer learning rates.
- Better to initialize two networks with "task-specific" networks
- Units are not too sensitive to initial α values
- Adds few parameters (maximum of one scalar per filter channel)
- Enforcing convexity in units does not help

What do these units learn? Visualizing and more.

Surface Normal and Semantic Segmentation on NYUv2



- More sharing at lower layers – pool1, pool2
- Less sharing at higher layers – fc6, fc7
- Least sharing at pool5
- Similar trend at different initializations

Learned cross-stitch units