Principles of Robot Autonomy I

Modern robotic perception

Stanford University

Autonomous Systems Lab

Today's lecture

- Aim
  - Gain a high-level understanding of how modern techniques from computer vision are applied in robotic systems

- Readings
  - CS 231A Course Notes
  - CS 231N Course Notes
  - Various computer vision conference papers

Where is AA 274A in the perception timeline?

- AA 274A covers computer vision from ~1540s to ~2000s

- What happened from the 2000s to now, and how have these advancements been used in robotics?

- We’ll explore methods from 2000s to 2012, and from 2012 to now.

Early 2000s Computer Vision

- Heavily based on structure and inductive biases
  - Feature engineering reigned supreme!

- Edge Detection
- Corner Detection
- Blob Detection
- Keypoint Detectors and Descriptors
  - SIFT, SURF, ORB, etc. How can you best describe a point in an image?

Cool methods, but what do we use them for?

- These are all tools for image processing, what problems are we even trying to solve in computer vision? Why do we care for robotics?

- Shifting from a method-centric view to a problem-centric one.

- Will describe important problems and approaches used to solve them up to a certain special date.

Object Detection and Classification

pre-2012

- Find if there is an object of interest in an image. If there is, classify what it is.

General solution idea:

1. Describe a region of an image
2. Classify that description
3. Shift your focus to the next part of the image

Object Detection and Classification

pre-2012

General solution idea:

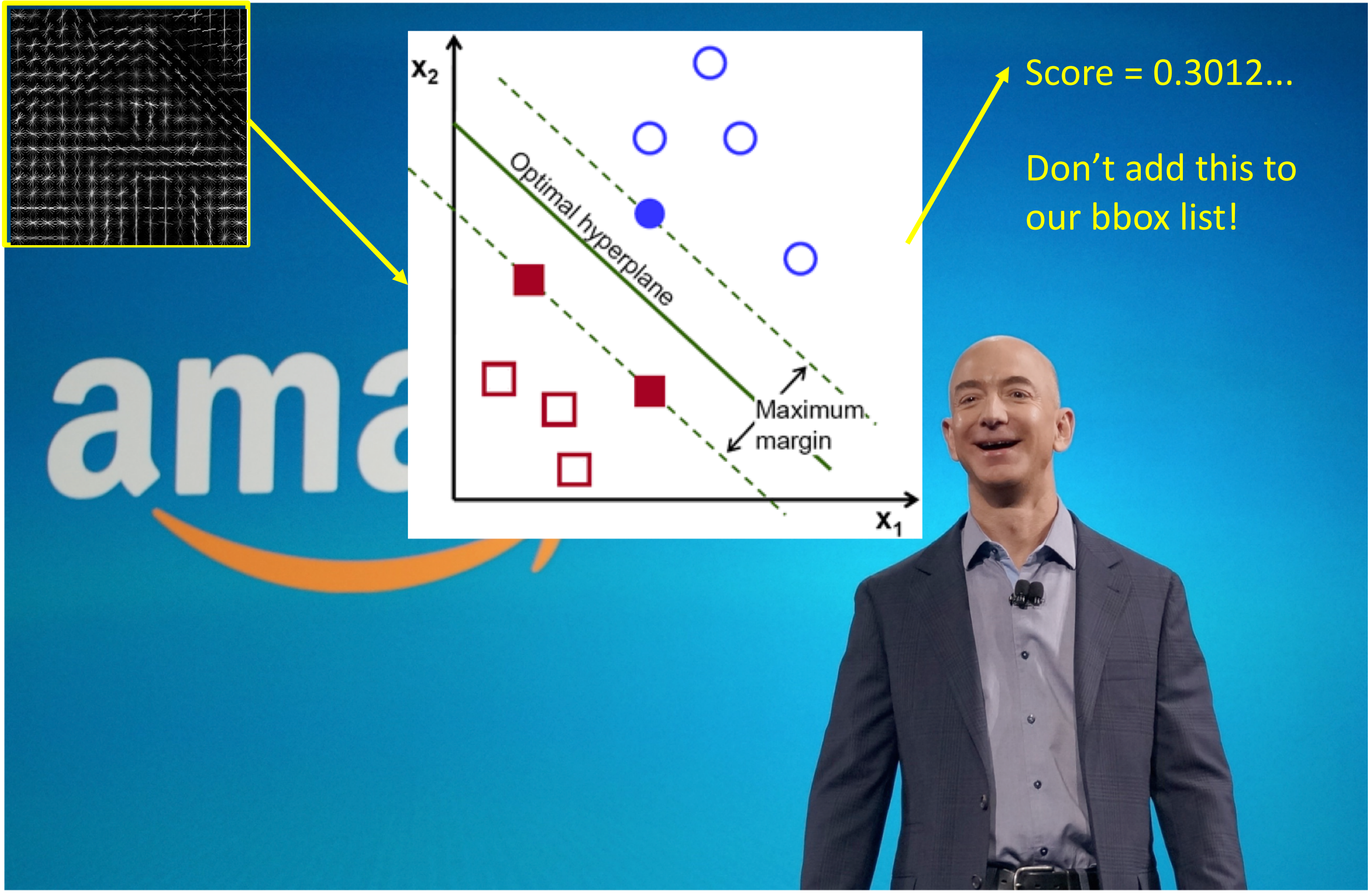
1. Describe a region of an image
   - HoG, SIFT, Patch Similarity, Codewords, etc.
2. Classify that description
   - Naïve Bayes, Nearest Neighbor, SVMs, Structural SVMs, Boosting, etc.
3. Shift your focus to the next part of the image
   - Sliding window

Let's see what HoG + SVM looks like, as an exemplary method

HoG + SVM (Pictorally)

HoG + SVM (Pictorally)

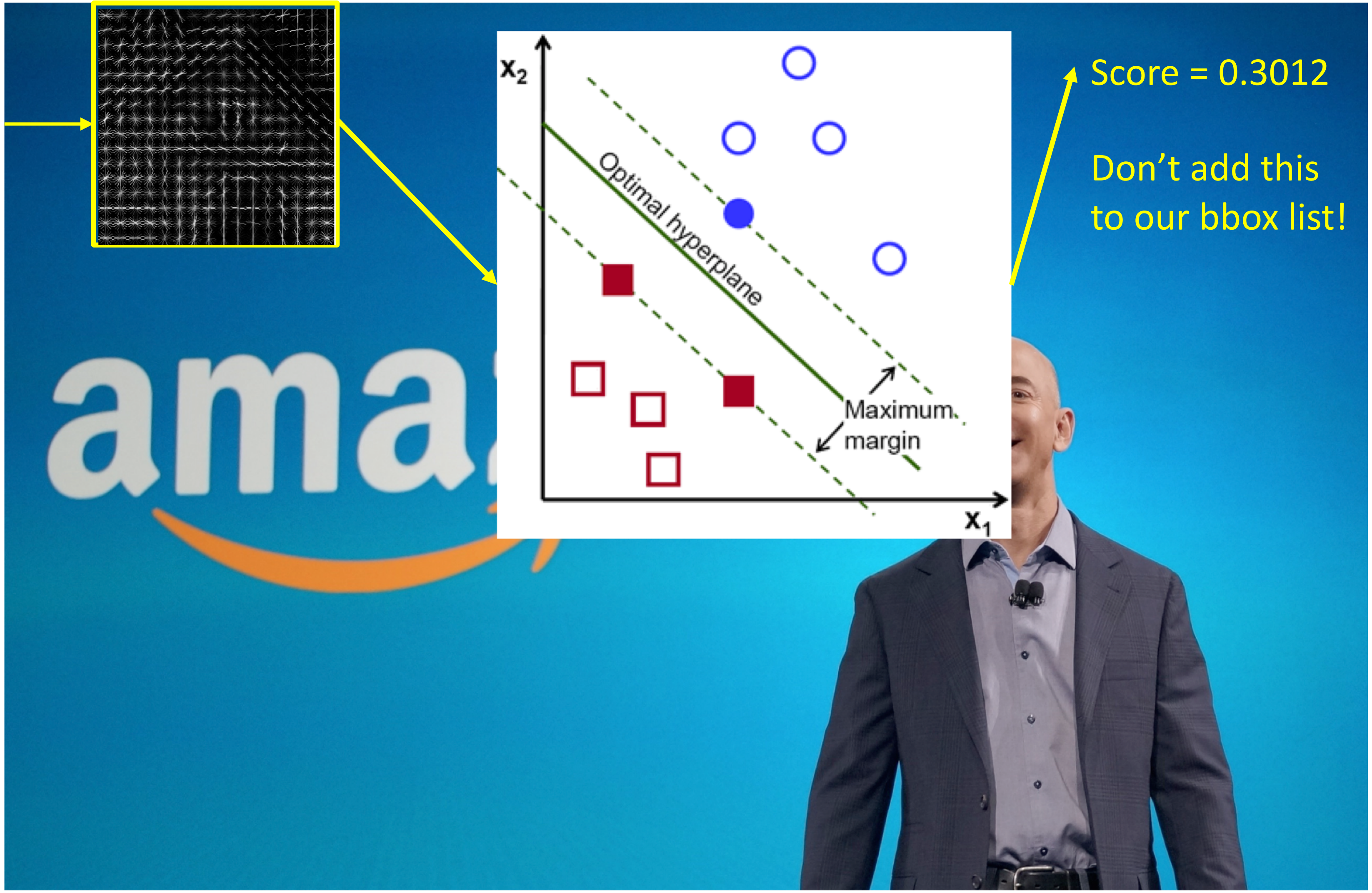
HoG + SVM (Pictorally)



HoG + SVM (Pictorally)

HoG + SVM (Pictorally)

HoG + SVM (Pictorally)



Score = 0.3012

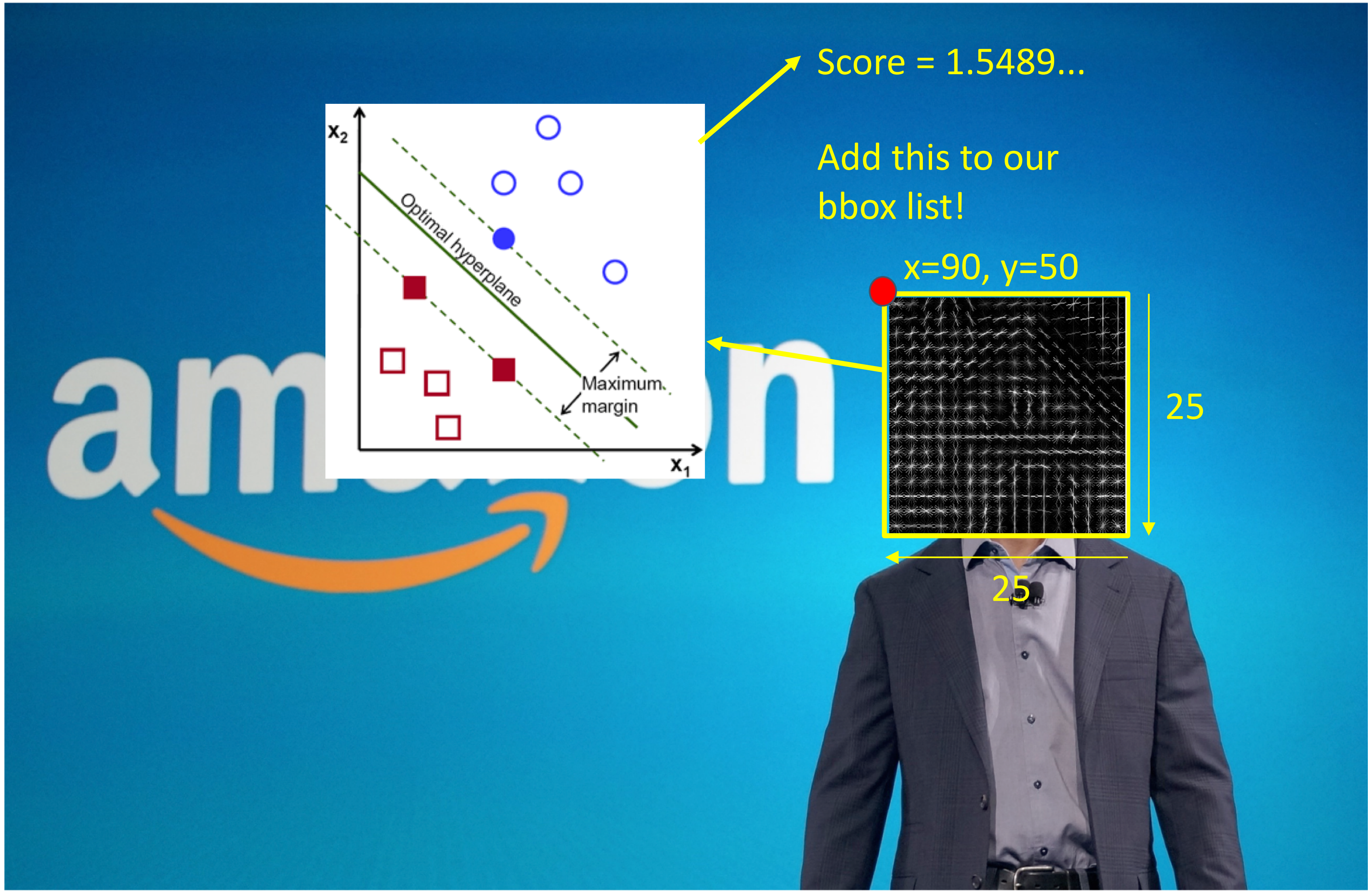
Don’t add this to our bbox list!

HoG + SVM (Pictorally)

…

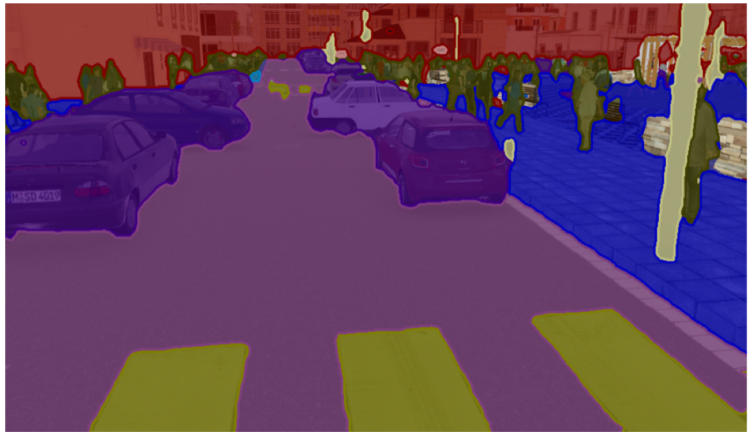
HoG + SVM (Pictorally)

HoG + SVM (Pictorally)



Scene Understanding: Semantic Segmentation pre-2016

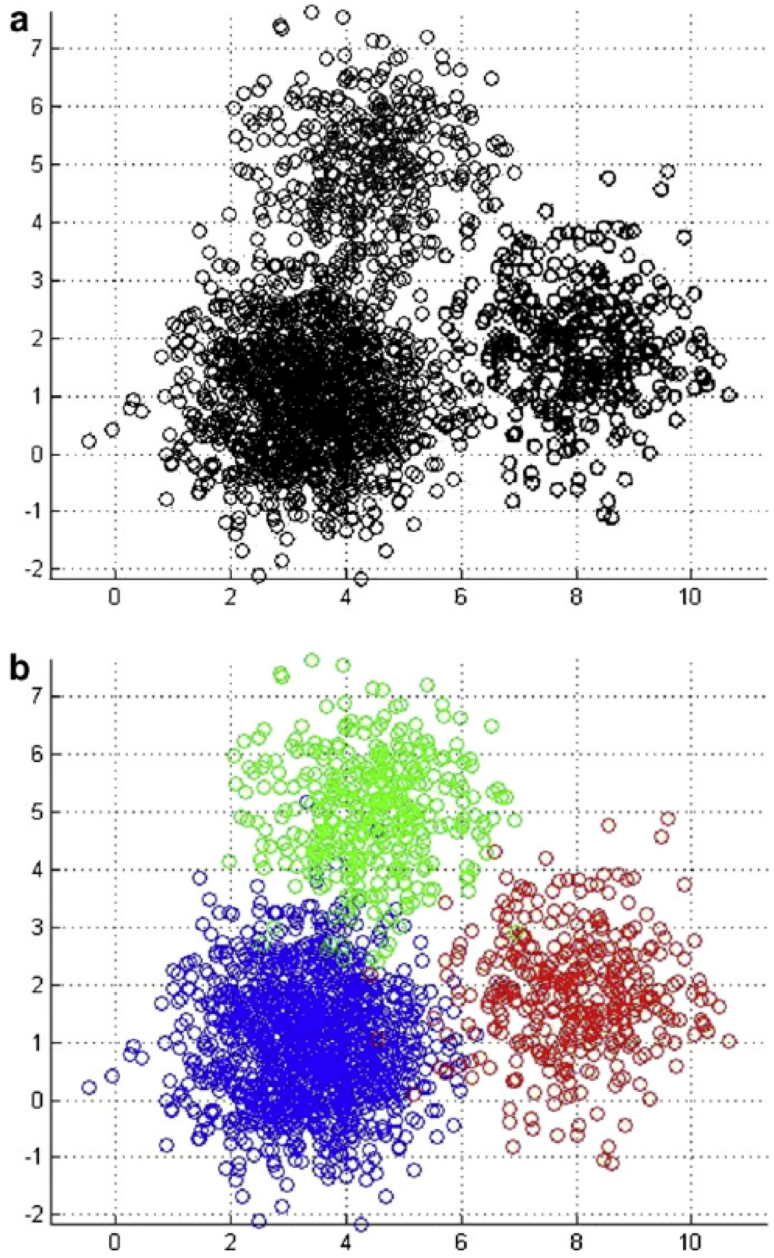
- Develop an understanding of what's going on in a scene by classifying every observed point.



- Clustering-based Segmentation
    - K-means, Mean Shift
- Graph-based Segmentation
    - PGMs (CRFs specifically)

K-Means Clustering

- A technique to cluster data for which you have no labels.
- For us: A method of grouping together "like" features in feature space.
- Called "K"-means because there's K clusters (a hyperparameter we have to choose before running the algorithm)
- Quite a simple algorithm internally!
  - Feel free to look up how it works

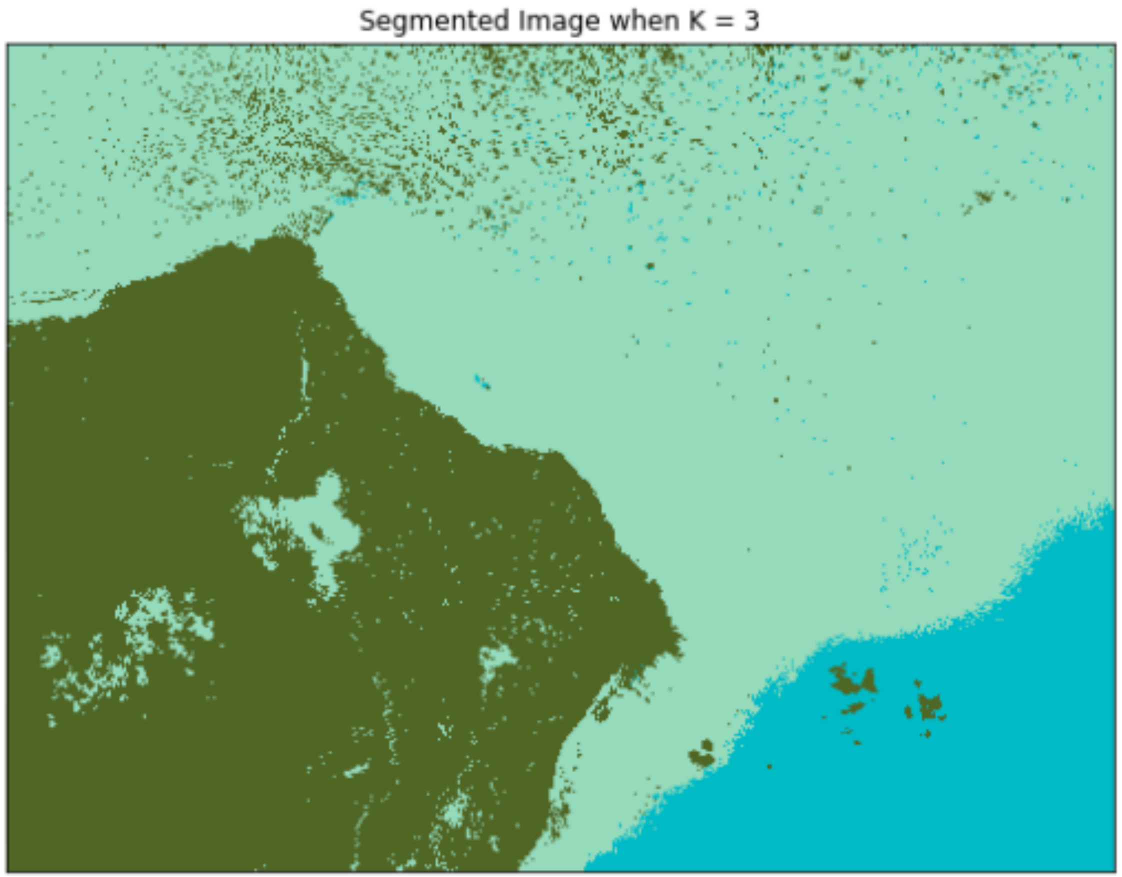


Semantic Segmentation via K-Means Clustering

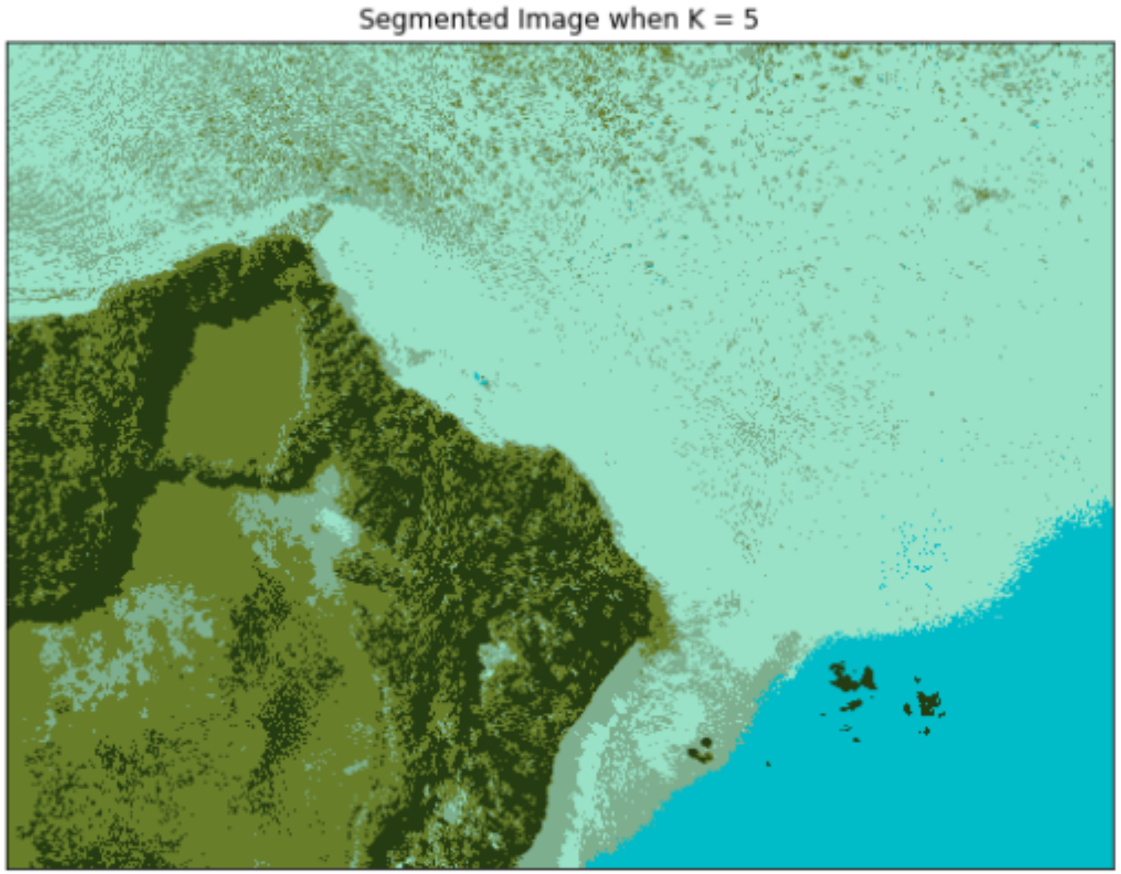
- The key idea is to map each pixel to a 5D feature, [X, Y, R, G, B], and then cluster pixels in this space, assigning them their classification label at the end of the clustering.

- Why?
  - X, Y capture spatial locality
  - R, G, B capture visual locality

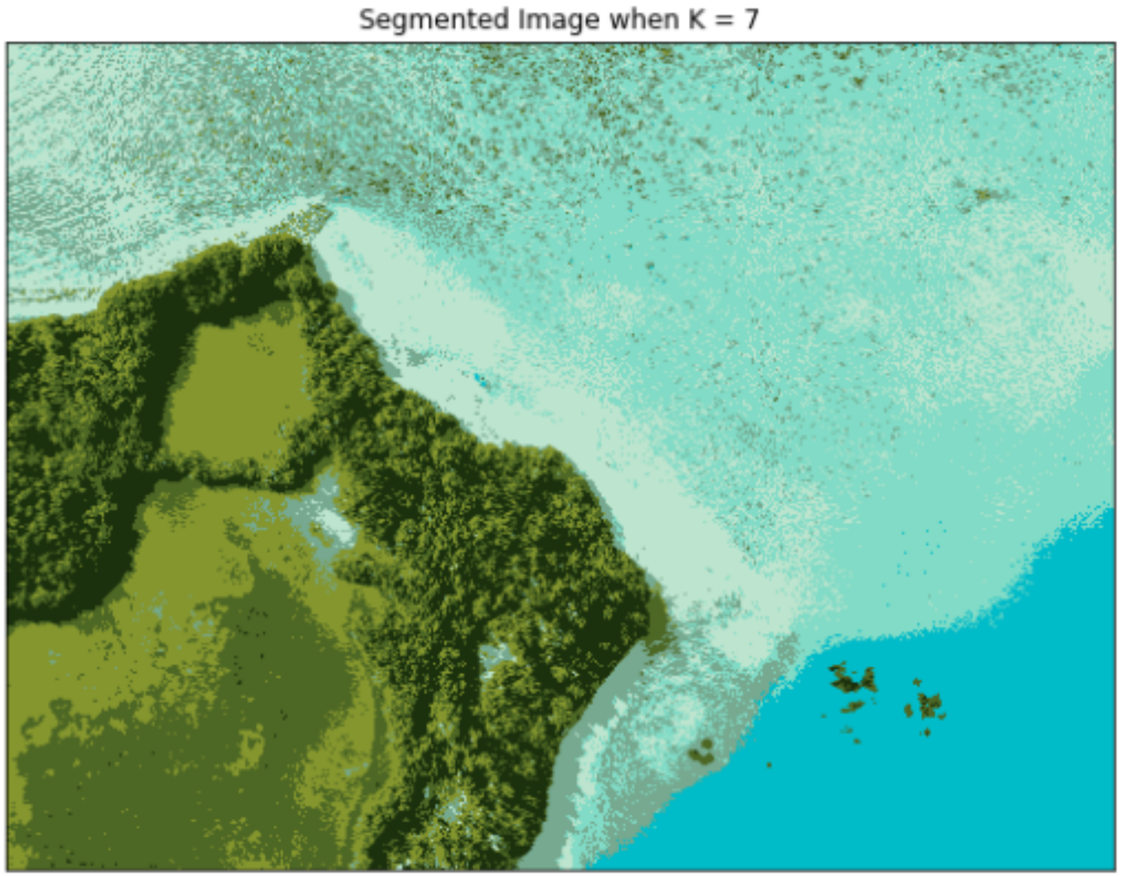
Semantic Segmentation via K-Means Clustering



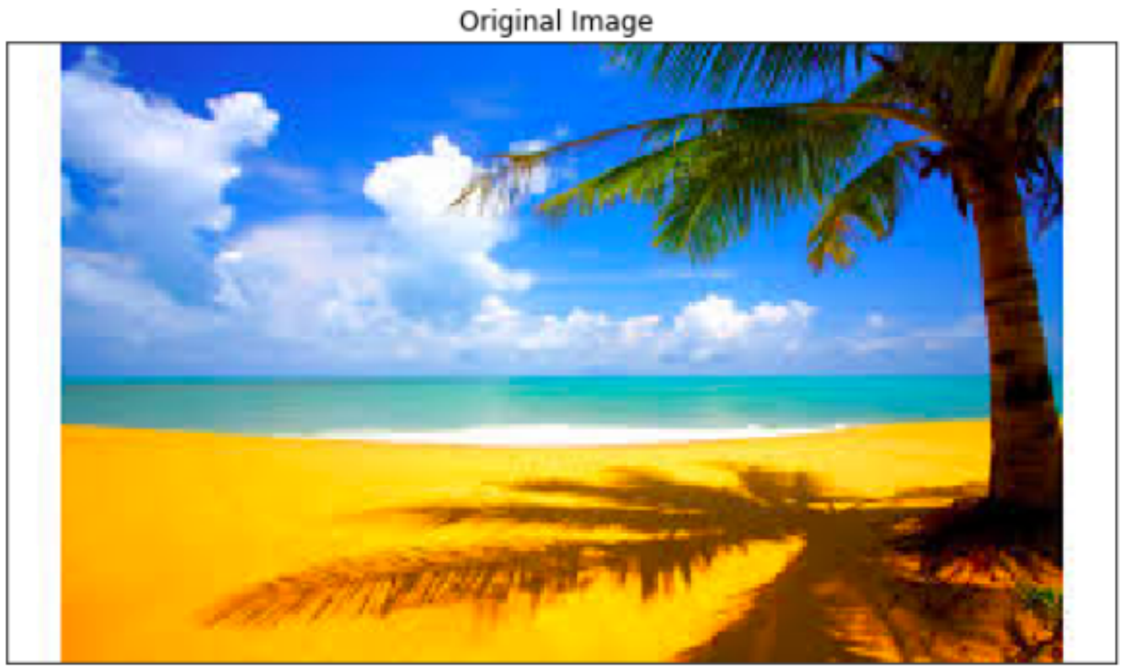
Semantic Segmentation via K-Means Clustering

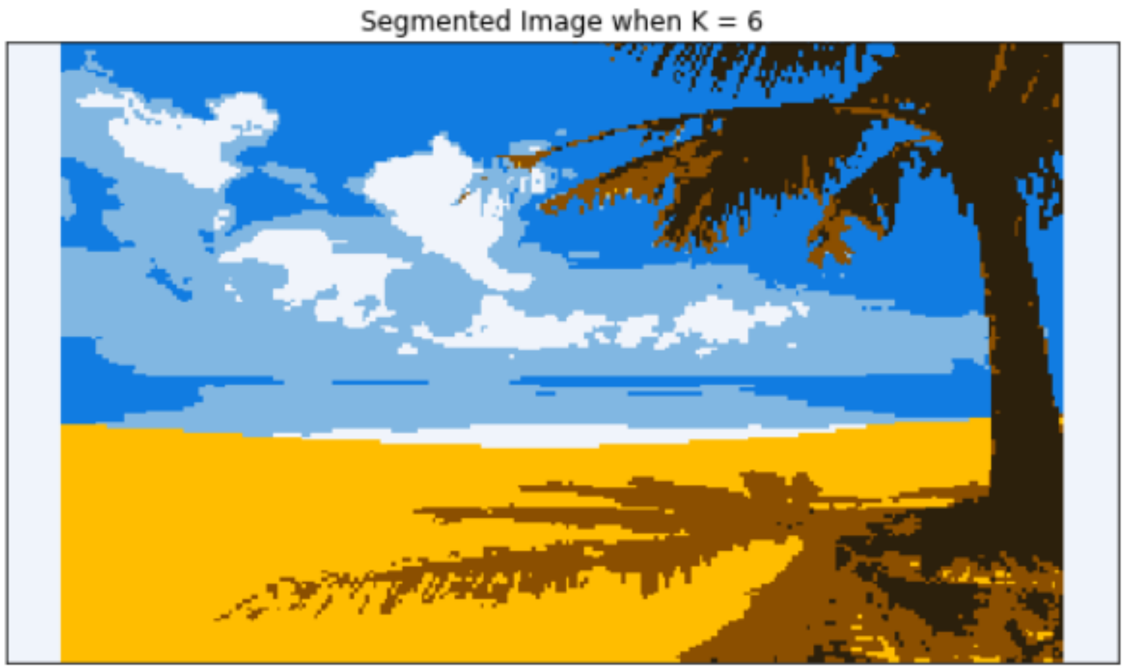


Semantic Segmentation via K-Means Clustering



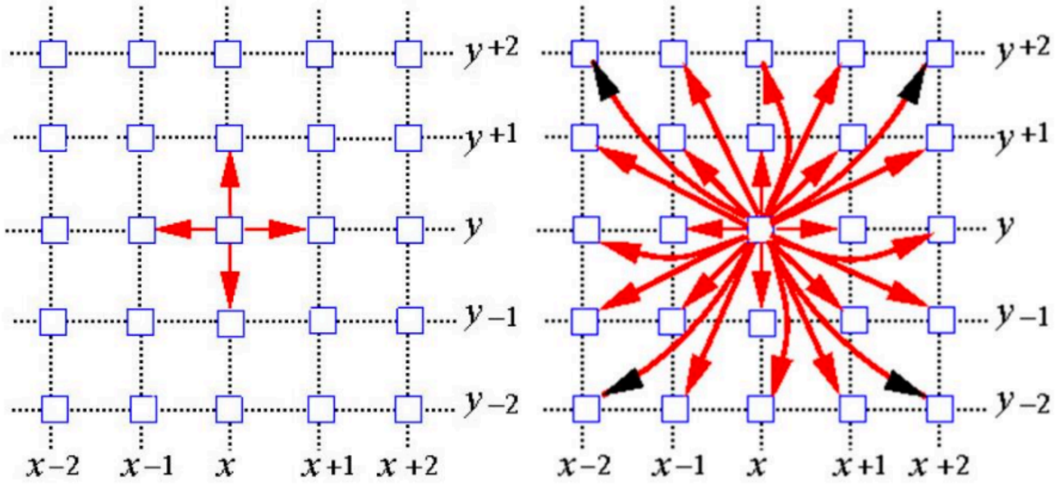
Semantic Segmentation via K-Means Clustering





Semantic Segmentation via Graph-Based Methods

- Take advantage of the grid-like structure of images, reason probabilistically about each pixel and how it's influenced by its neighbors!
- Probabilistic graphical model of choice was a "Conditional Random Field"

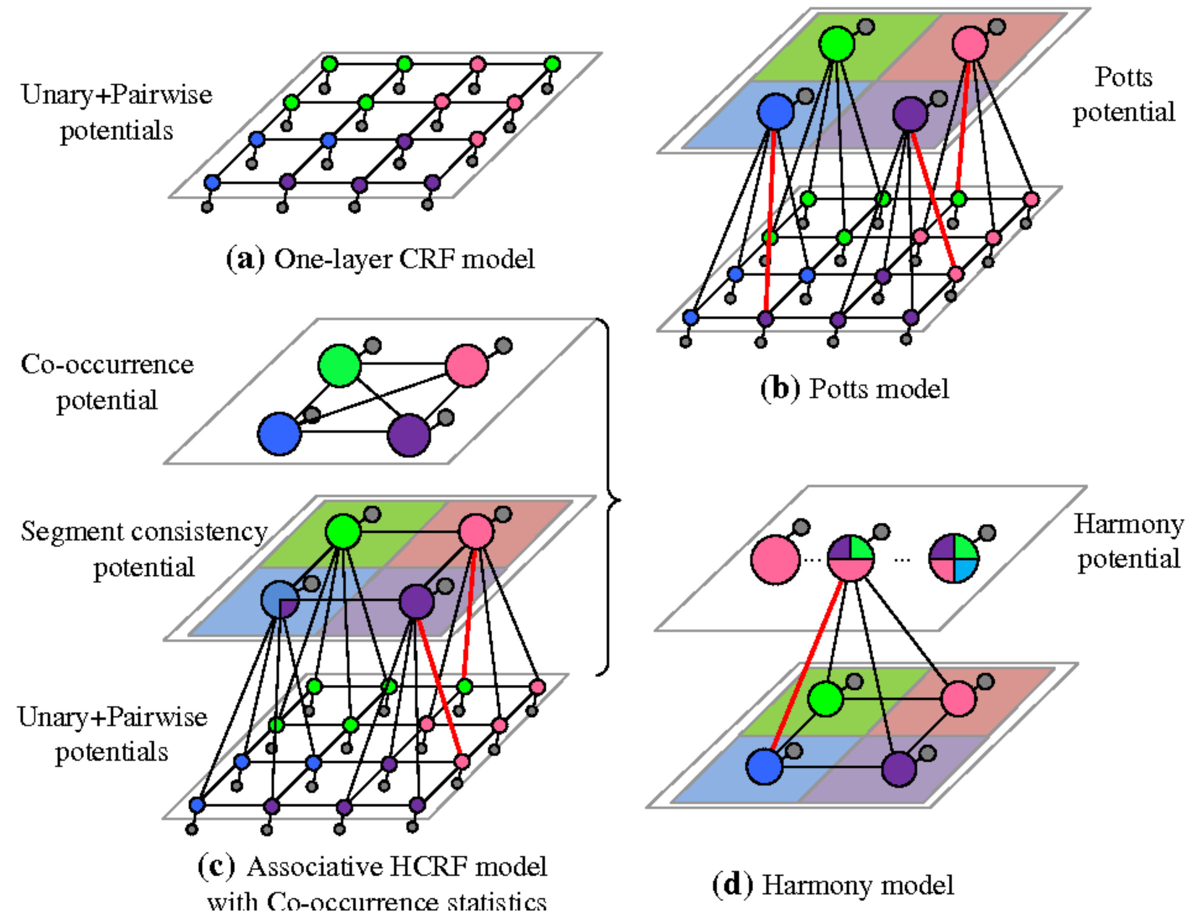


(a) (b)

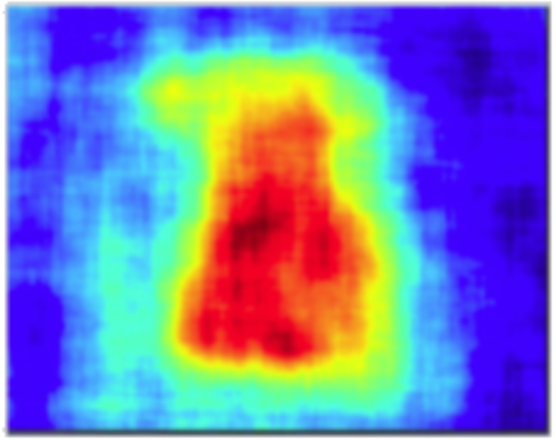
- Anyone here take Mykel's DMU course? It's a Bayes Net on steroids

Semantic Segmentation via Graph-Based Methods

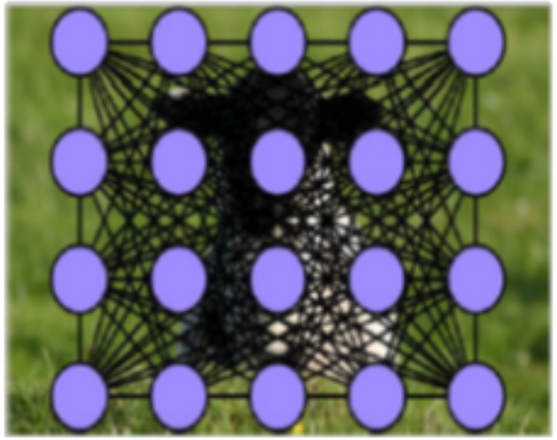
- Don't need to simply view images as grids with simple connectivity.
- Literature is quite deep on this topic, with many different hierarchical structures proposed and math developed for them.



Semantic Segmentation via Graph-Based Methods



Coarse output from the pixel-wise classifier



MRF/CRF modelling

Output after the CRF inference

Simultaneous Localization and Mapping

- A key component of robotic motion planning is knowing where you are in the world as well as what's around you. This is the problem of simultaneous localization and mapping (SLAM).
- You'll learn this in AA 274A! ☺

- Traditionally, this was 100% a game of geometry.
  - Remember structure from motion and 3D reconstruction? Exactly that.
- Nowadays, mostly complex large-scale SLAM pipelines.
  - *Significant* engineering effort required for state-of-the-art results.
  - Usually small teams of software engineers and researchers.

Simultaneous Localization and Mapping

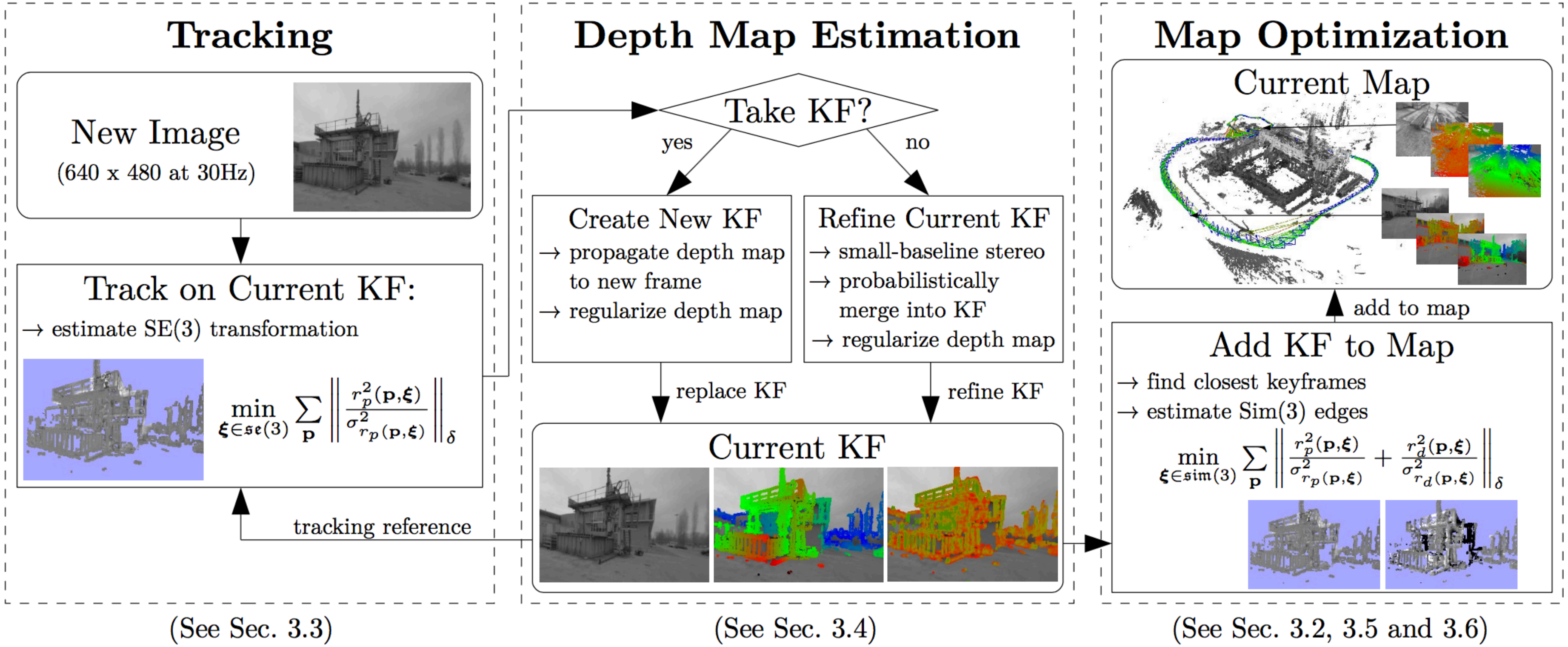


Fig. 3: Overview over the complete LSD-SLAM algorithm.

- https://www.youtube.com/watch?v=ufvPS5wJAx0 ORB-SLAM
- https://www.youtube.com/watch?v=C6-xwSOOdqQ DSO

What happened in 2012?

- While the world didn't physically end, it might as well have for a large portion of computer vision research prior to then… ☹

- Very large paradigm shift occurred, spinning the entire field on its head within the span of one year.

- What happened??

Background to Understand the *Paradigm Shift*

It is important to understand another popular field of research that existed around this time (and for decades earlier): Machine Learning.

Core idea, and a *ridiculously large* oversimplification:

We wish to predict a quantity $\boldsymbol{y}$ from input $\boldsymbol{x}$. Namely, $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$

1. Formulate a function L that encodes how "bad" our predictor f is.
2. Minimize L with respect to the parameters $\boldsymbol{\theta}$ that define f.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

Background to Understand the *Paradigm Shift*

- Machine Learning (ML) has existed for decades, if not more, but under different names: Applied Probability, Statistics, and Linear Algebra

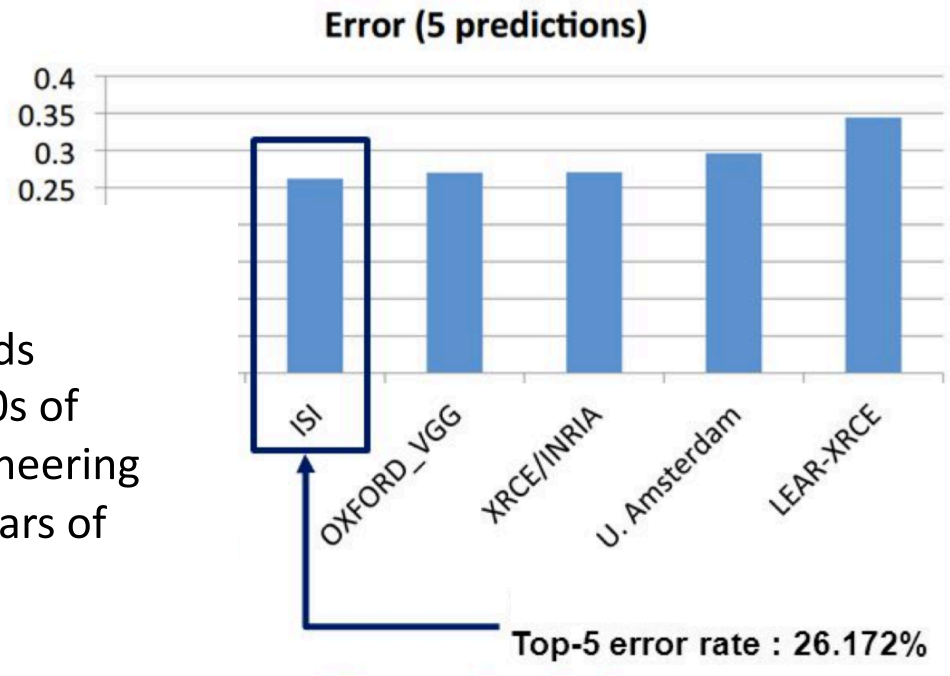
- Typically concerned with modeling continuous data (e.g. regression tasks) or performing discrete segmentation (e.g. classification tasks)

- Images were definitely considered as inputs and objects of interest (they are just 2D matrices / 3D tensors). However, existing methods in ML haven't taken advantage of inherent structure in vision.

Computer Vision’s Recent Paradigm Shift

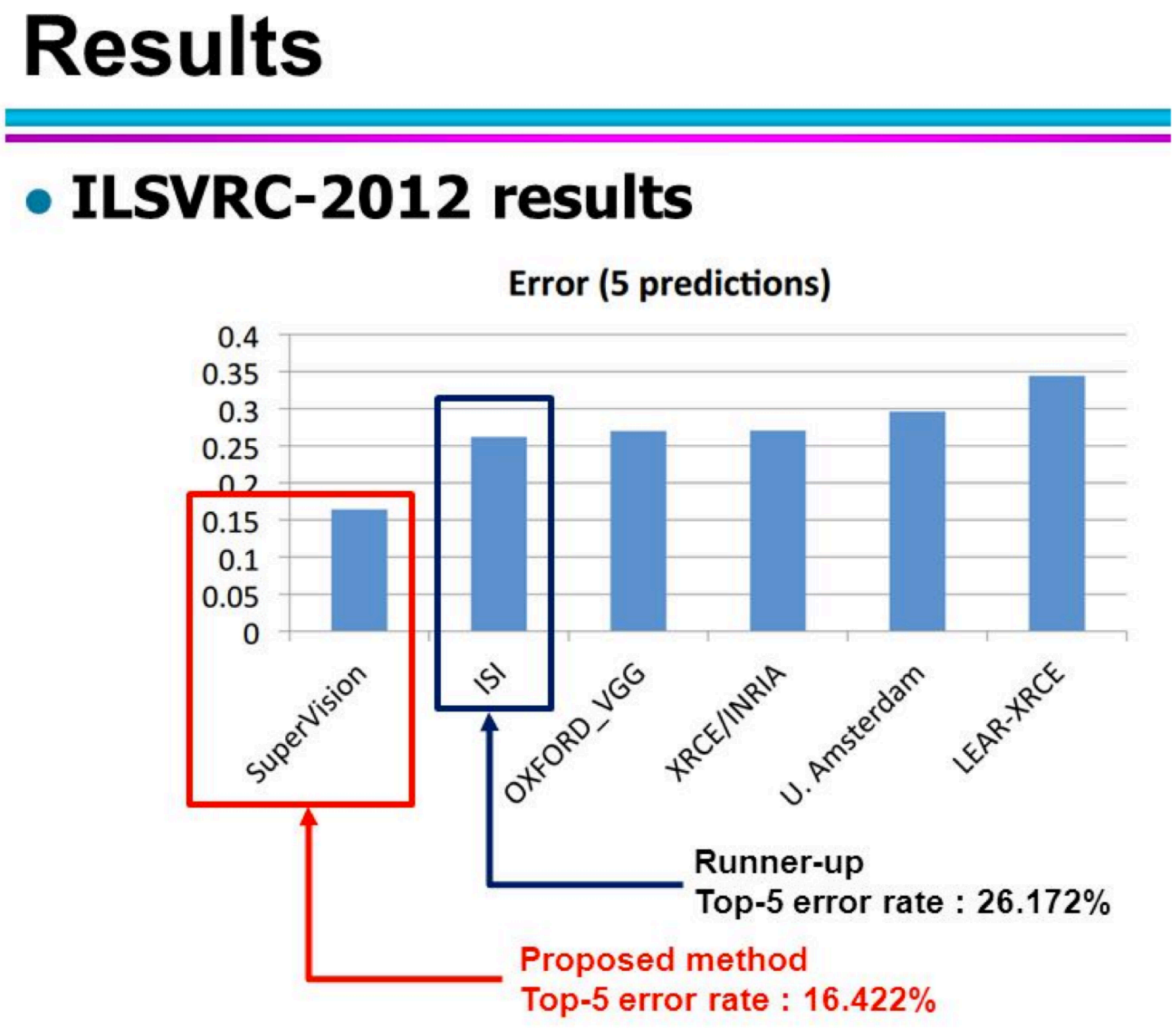
Results

- **ILSVRC-2012 results**



These methods represent 100s of hours of engineering effort, and years of fine-tuning + optimization.

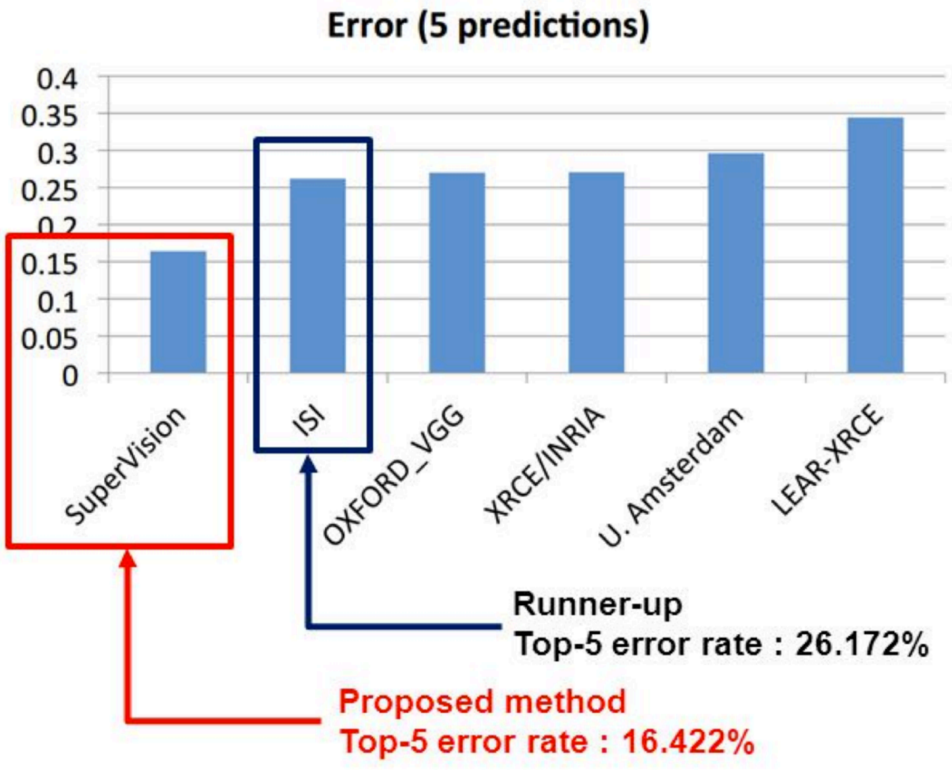
Computer Vision’s Recent Paradigm Shift



Computer Vision's Recent Paradigm Shift

Results

- **ILSVRC-2012 results**



UNIVERSITY OF TORONTO

Convolutional Neural Networks

We wish to predict a quantity $\boldsymbol{y}$ from input $\boldsymbol{x}$. Namely, $\boldsymbol{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$

1. Formulate a function L that encodes how "bad" our predictor f is.
2. Minimize L with respect to the parameters $\boldsymbol{\theta}$ that define f.

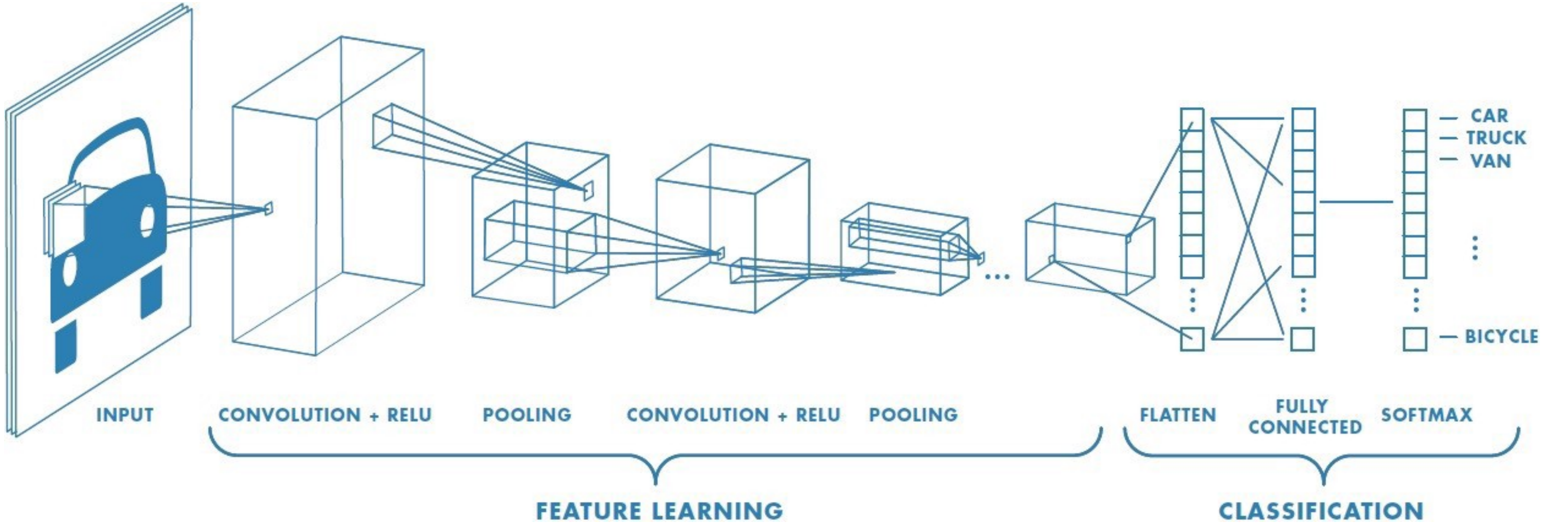
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

Convolutional Neural Networks are a specific structure of the predictor function f, one that reigns supreme for visual data.

Convolutional Neural Networks

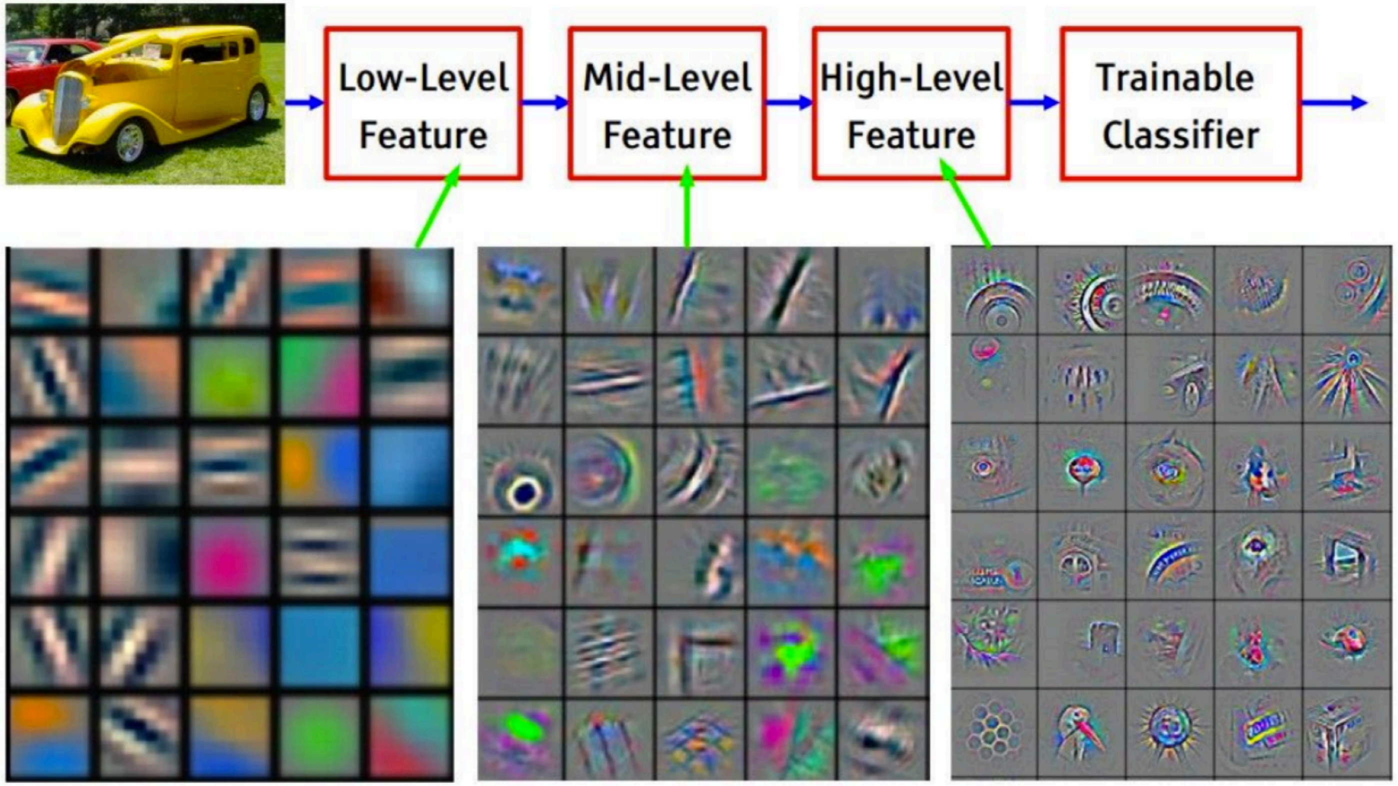
Roughly, they are a hierarchical set of successive convolutions with learnable filters (the filters are the $\boldsymbol{\theta}$ here).

Modeled after how animals and humans process visual information in the brain.



Convolutional Neural Networks

- Once trained to minimize L, they develop some very interesting filters



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

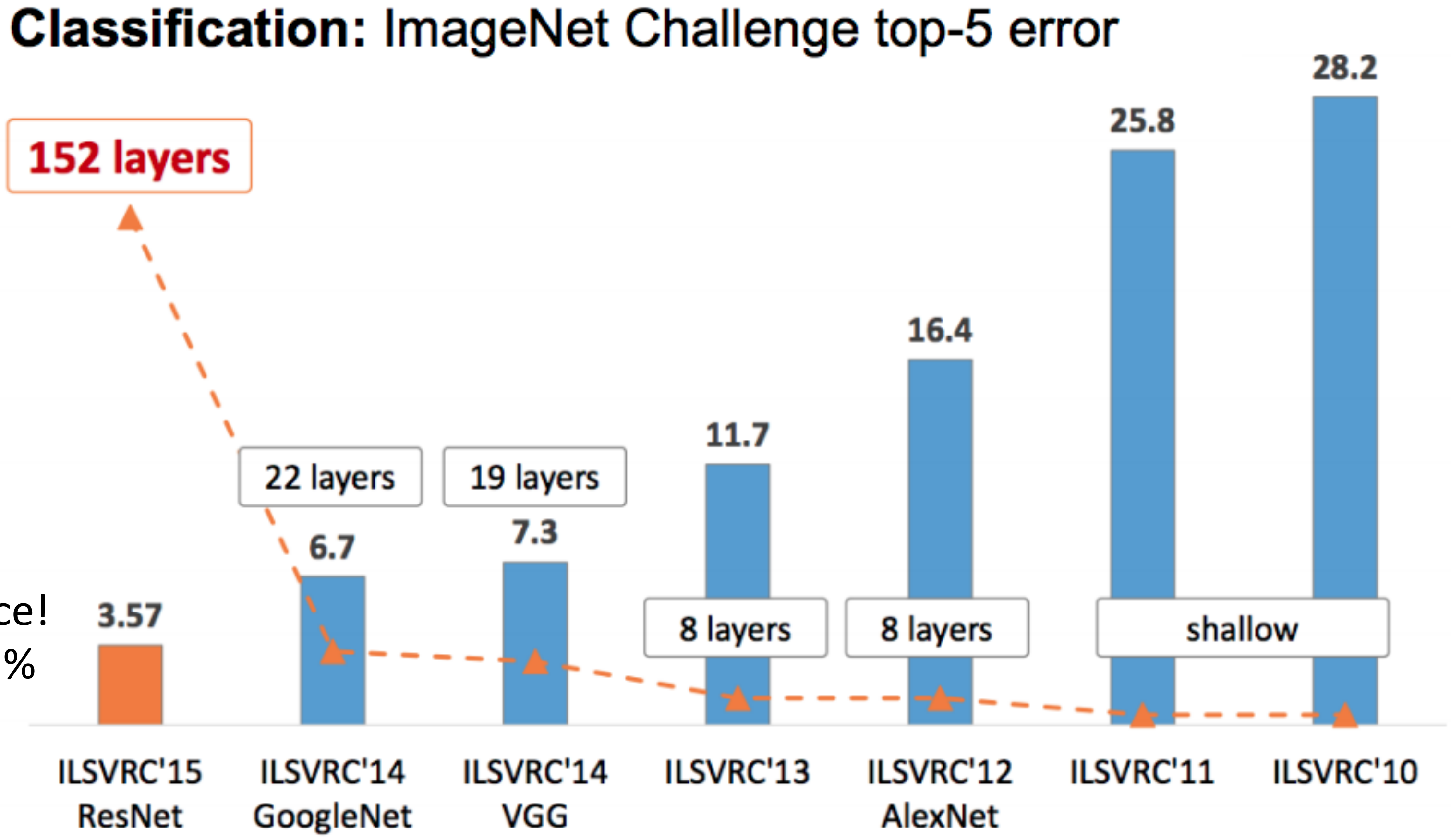
Live Demo – Inner Workings of a CNN

- http://scs.ryerson.ca/~aharley/vis/conv/

- There's also a flat version: http://scs.ryerson.ca/~aharley/vis/conv/flat.html

Race to the Lowest Error

- On problems like image recognition, CNNs are extremely powerful.

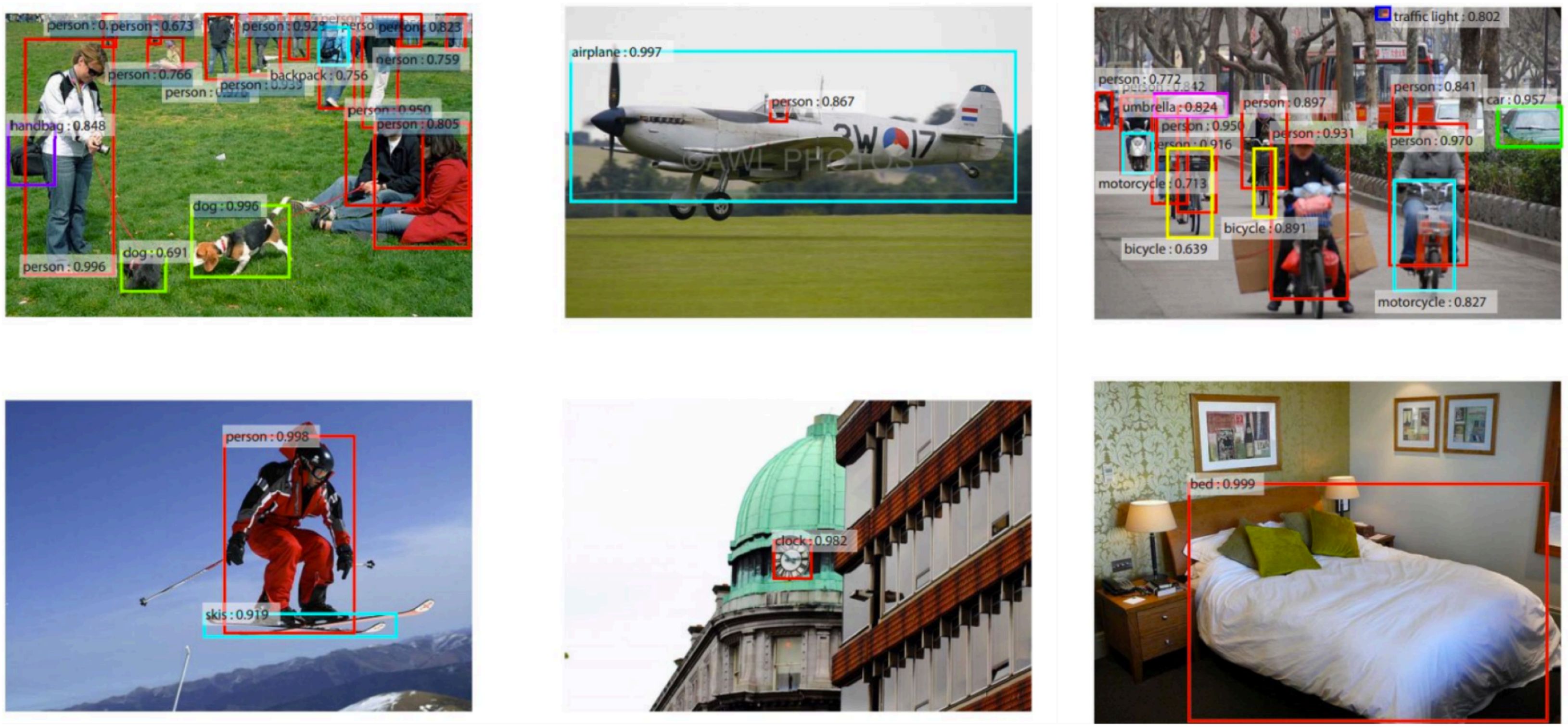


Superhuman performance!
Human performance is 5%

Object Detection and Classification

post-2012

- Methods now focus on having a CNN architecture both detect regions of interest in images *and* classify them.

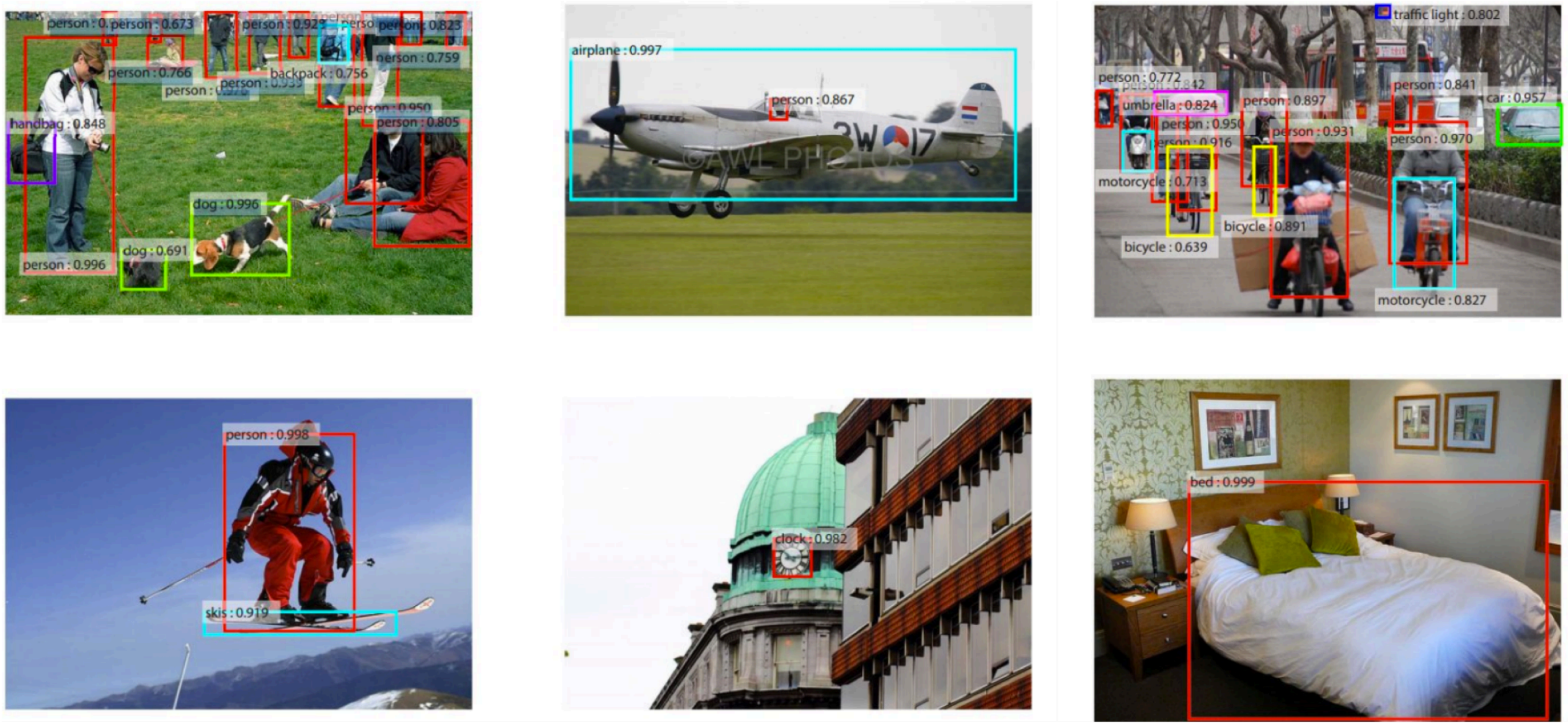


Results from Faster R-CNN, Ren et al 2015

Object Detection and Classification

post-2012

- Can do this by making L incorporate both a classification term and a regression term (for the bounding box locations).

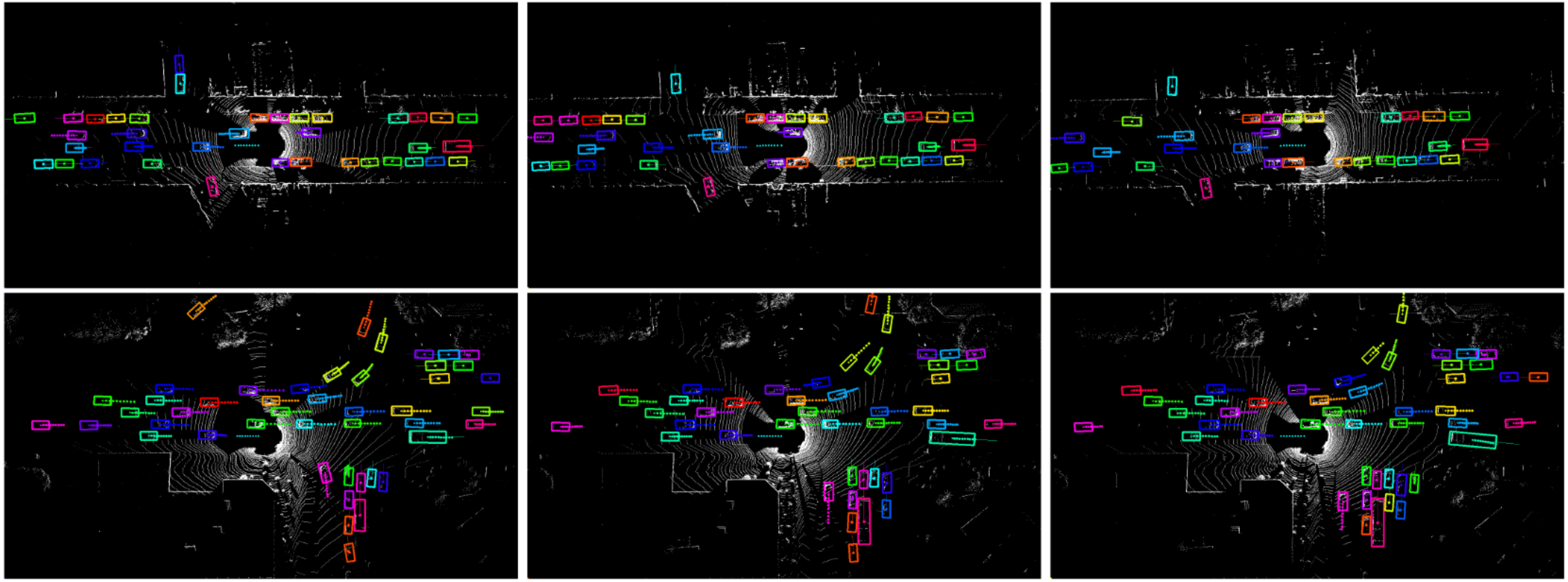


Results from Faster R-CNN, Ren et al 2015

Object Detection and Classification

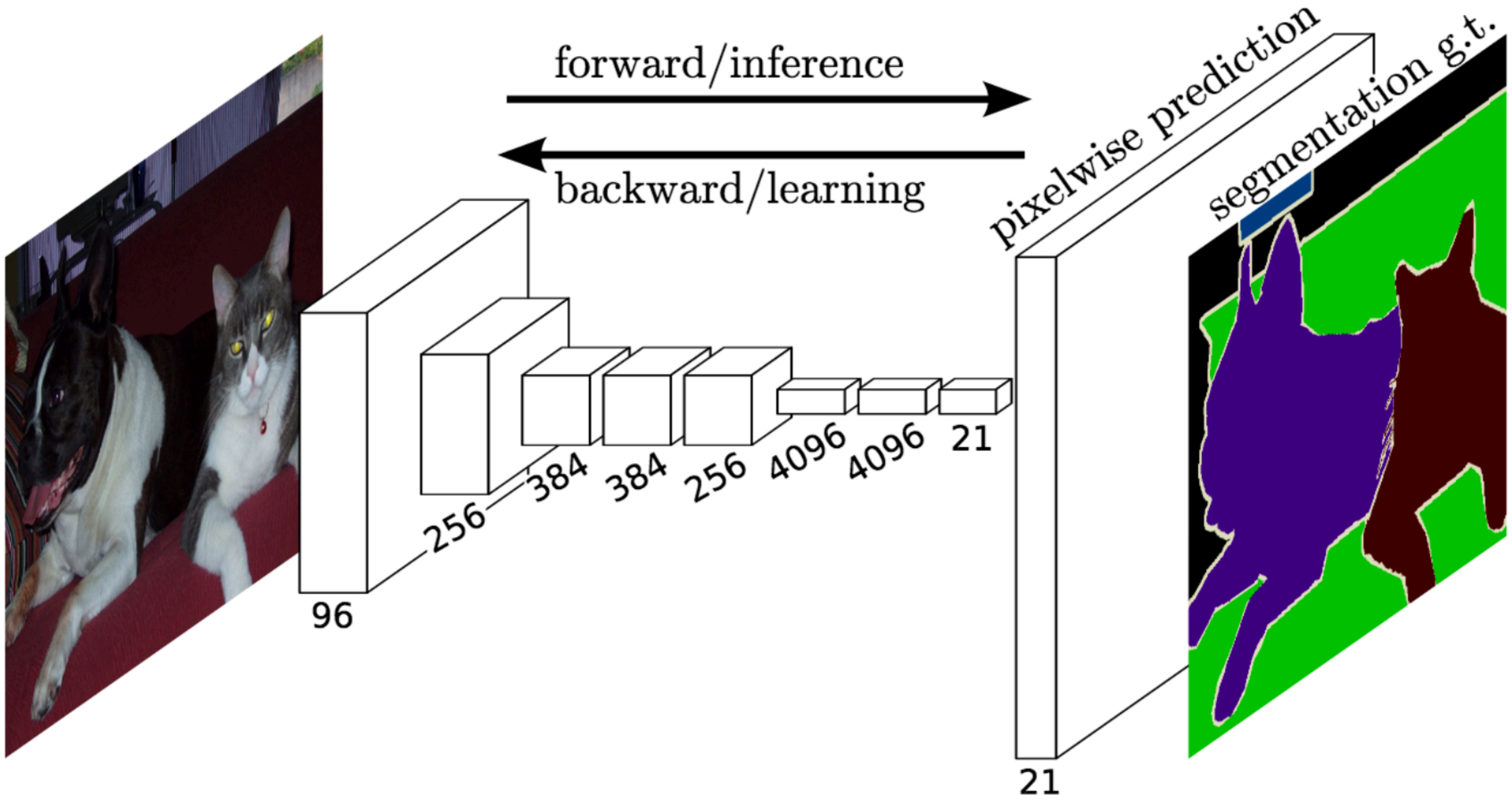
post-2012

3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net (from Uber's Advanced Technologies Group)



Semantic Segmentation with CNNs

CNNs for Semantic Segmentation (from T. Darrell's group at Berkeley)



Semantic Segmentation with CNNs

CRFs still exist as a post-hoc corrective step (if any), but vastly less research effort towards solely using them.

- Ideas and methods from CRFs absolutely live on in probabilistic inference, which is a popular topic due to deep generative models.

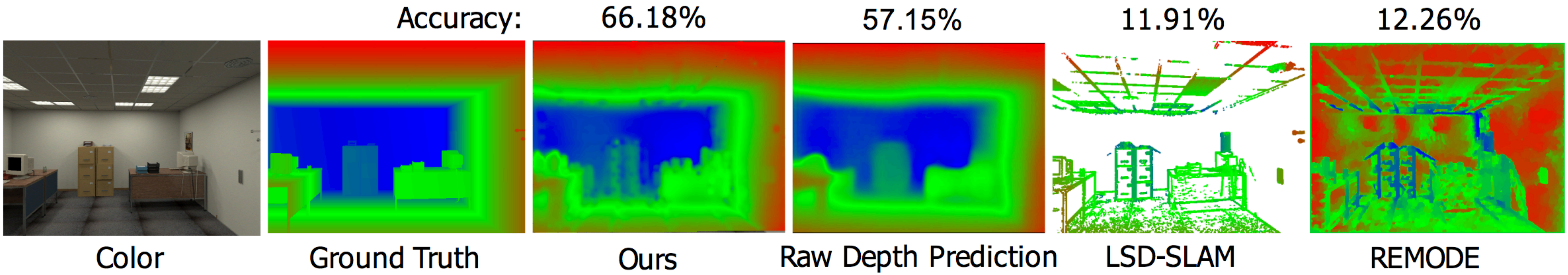
Methods like K-means or mean shift are seldom used nowadays for semantic segmentation.

- Both methods are useful standalone algorithms in their own right, and are more known for clustering use cases anyways.

SLAM with CNNs

- There's definitely still hefty software engineering going on, especially with regards to global map updates, loop closure, etc.

Real-time Dense Monocular SLAM with a CNN for Depth Prediction



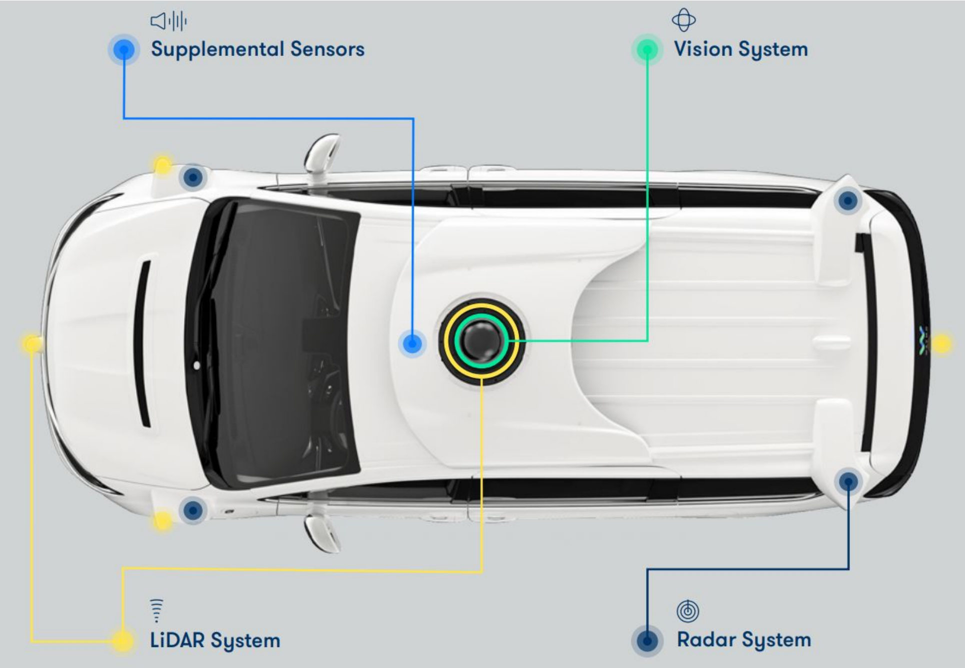
3-4x better than LSD-SLAM and ORB-SLAM from earlier, and we thought they were amazing!

Heterogeneity of Data

Modern robotic platforms have a wide array of sensors and data modalities available to them.

For example, Waymo vehicles have:

- 1 mid-range lidar
- 4 short-range lidars
- 5 cameras (front and sides)
- Synchronized lidar and camera data
- Sensor calibrations and vehicle poses



Heterogeneity of Data

In pre-2012 computer vision, we would be assigning feature engineering teams to *each* sensor and data modality.

Now, it's as easy* as adding additional input channels, vectors, etc to an existing model (potentially altering L) and retraining.

*Easy is used loosely here, it's not trivial to make CNNs behave the way you want them to.

Summary

- Pre-2012, computer vision was engineering-heavy. Published methods were a result of clever feature selection, detection, description, etc.
- Now, a single CNN model takes care of that and optimizes for the best features/descriptor all at the same time.

- New wave of deep learning has revolutionized computer vision, the best of which has now trickled down to related fields such as robotics.
- Still lots of research to be done, much is still unknown!