Deep Learning for Computer Vision

commercial-in-confidence

DATA TO DECISIONS CRC

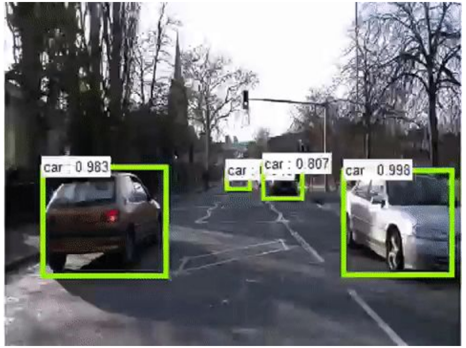
Introduction to Computer Vision & Deep Learning

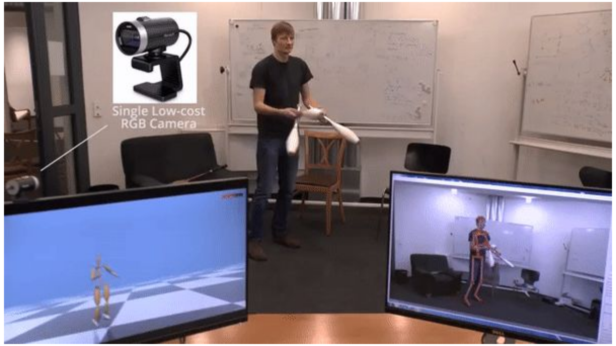
Presented by Hayden Faulkner

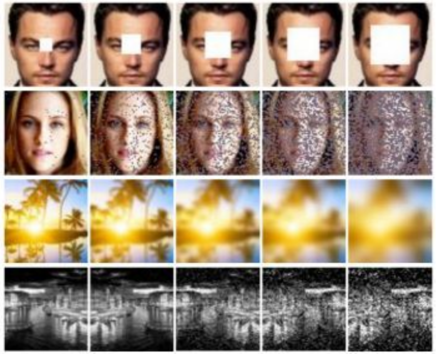
What Is Computer Vision?

What is Computer Vision?

Using computers to understand (process) imagery



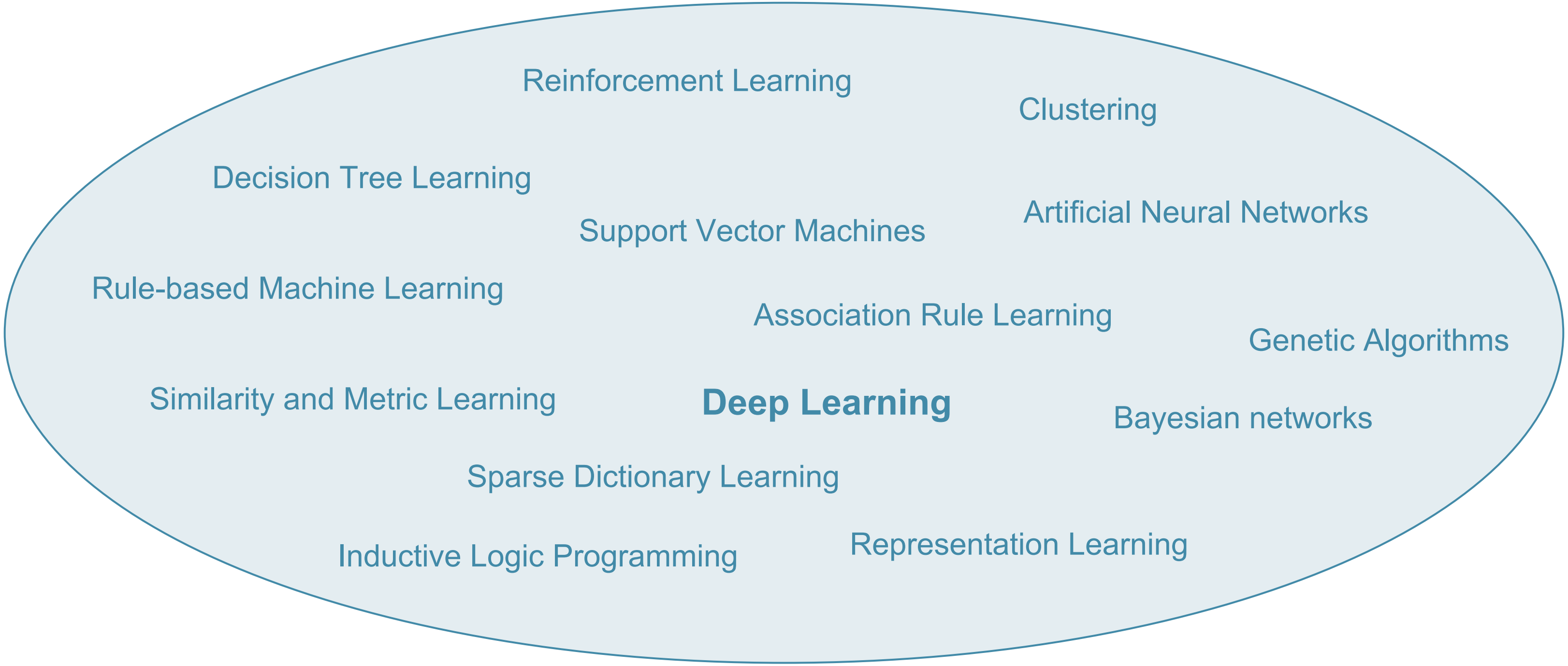




What Is Deep Learning?

What is Deep Learning?

Part of a broader set of **Machine Learning** methods



What is Deep Learning?

Deep Learning methods focus on
learning data representations via a set of
many sequential operations*

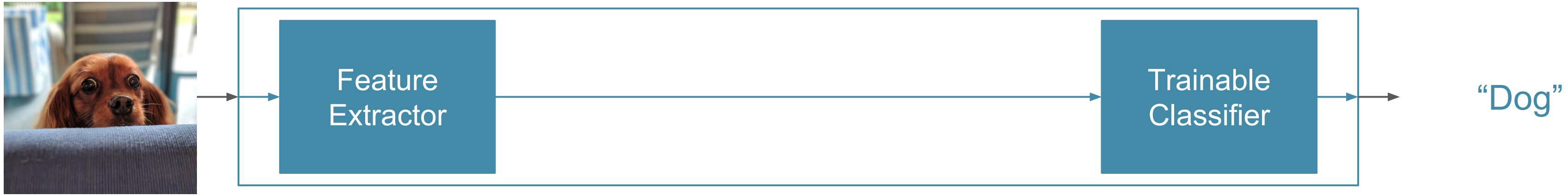
**Many experts have their own definition*

Image Classification: A Fundamental Computer Vision Problem

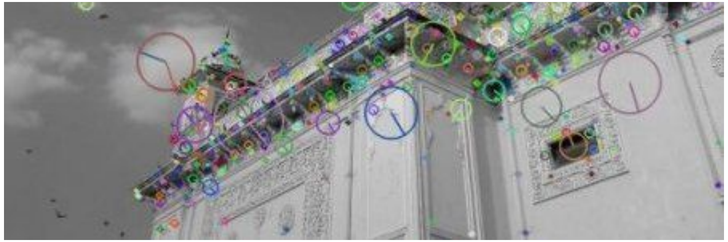
Process / Model

"Dog"

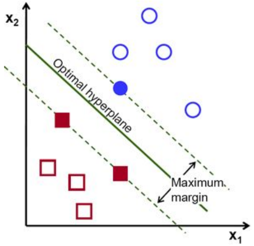
Image Classification: A Fundamental Computer Vision Problem



- Process the image into a lower dimensional space more useful for classification
- Features hand-crafted (designed) by researchers
- Used for picking up image properties such as edges or patterns
- Some features: SIFT, HOG, LBP, MSER, Color-SIFT …

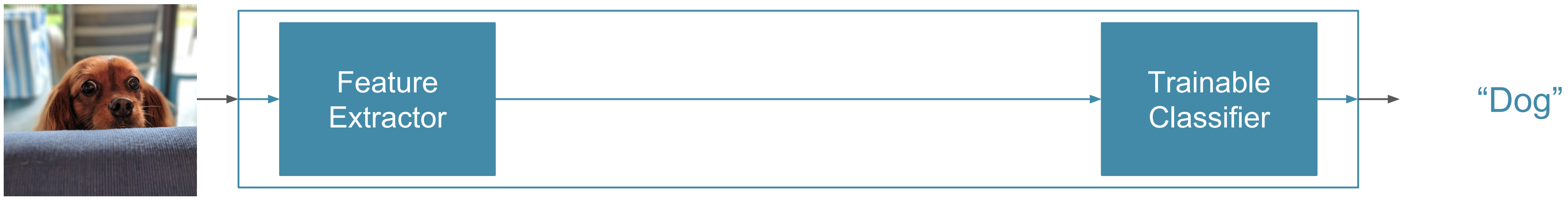


- Classifier uses image features to decide a label
- Utilises machine learning to learn classifier parameters, but it’s not deep learning
- Different classifiers learn and classify in different ways, a popular choice has been Support Vector Machines (SVM)
- SVMs attempt find hyperplanes in the high dimensional feature space to separate features from different classes



Left from: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_sift_intro/py_sift_intro.html
Right from: https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

Image Classification: A Fundamental Computer Vision Problem

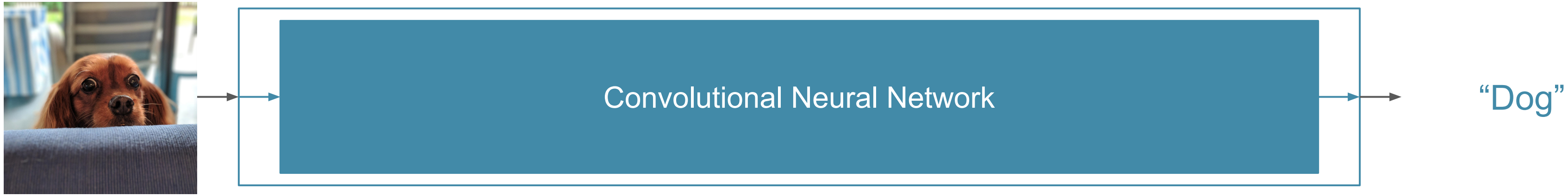


- Process the image into a lower dimensional space more useful for classification
- Features hand-crafted (designed) by researchers
- Used for picking up image properties such as edges or patterns
- Some features: SIFT, HOG, LBP, MSER, Color-SIFT, …

- Classifier uses image features to decide a label
- Utilises machine learning to learn classifier parameters, but it’s not deep learning
- Different classifiers learn and classify in different ways, a popular choice has been Support Vector Machines (SVM)
- SVMs attempt find hyperplanes in the high dimensional feature space to separate features from different classes

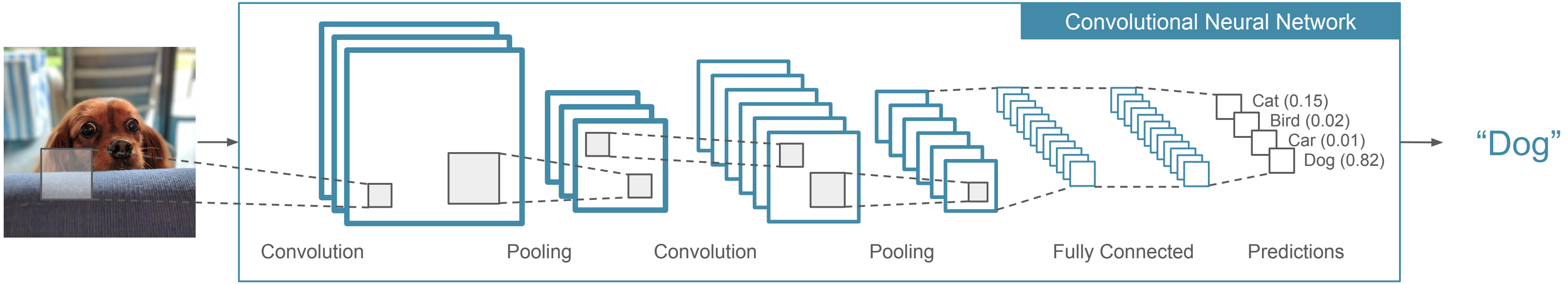
This worked okay, but it wasn’t very scalable, the hand-crafted features weren’t rich enough to handle many different object types and object appearance variations (pose, lighting, orientation, scene)

Image Classification: A Fundamental Computer Vision Problem



- Filters (that make features) are learnt by the computer so they are most useful for classification
- The classifier is in-built as part of the Convolutional Network architecture, no need for two separate stages
- Learnt end-to-end, the entire process from the input image to the label is learnt together, providing better relations between the features and the classifier

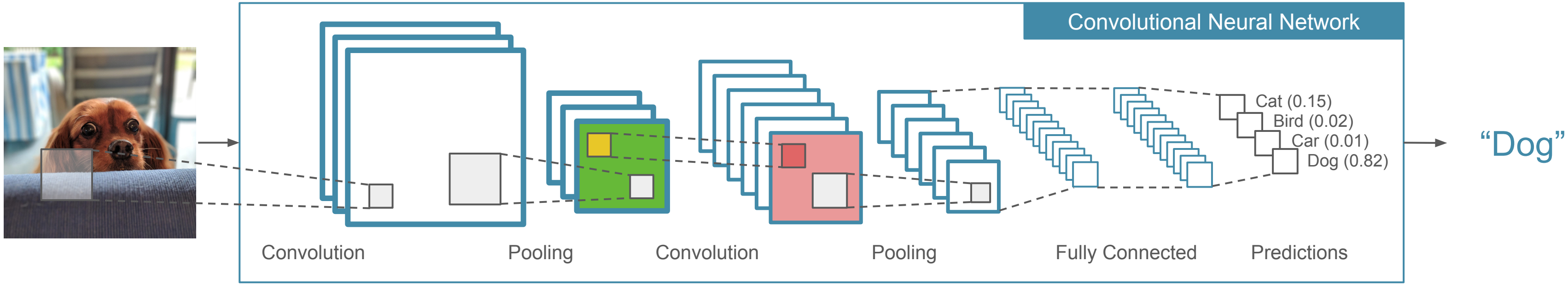
A Deeper Look at Convolutional Neural Networks: Structure



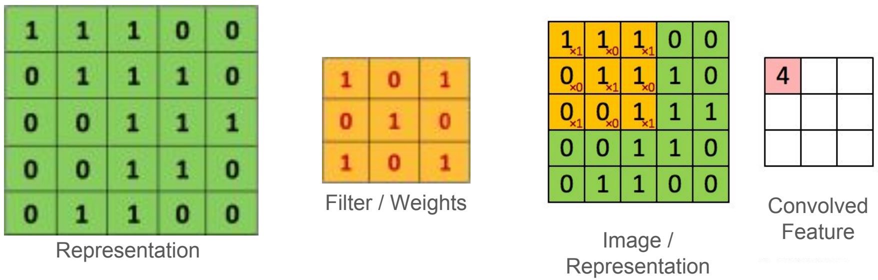
- Built of many layers that process image from pixels to label in a hierarchical and sequential manner
- What makes it deep learning is the sequential layer operations to learn different data representations based on previous layers
- So many parameters to learn, need lots of data, and lots of compute power, this is a key reason for its rise now

Diagram adapted from: https://www.clarifai.com/technology

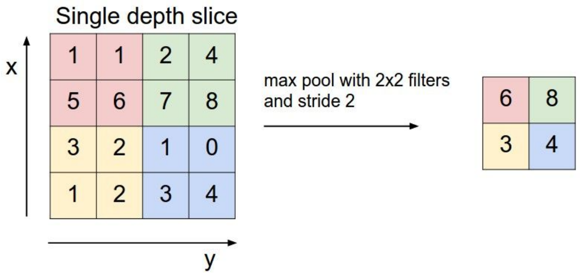
A Deeper Look at Convolutional Neural Networks: Operations



2D Convolution Operation



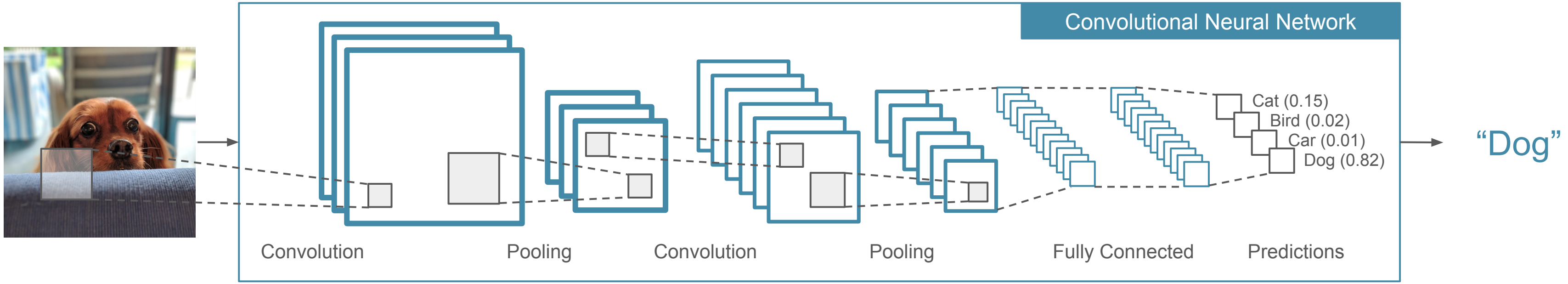
2D Max Pooling Operation



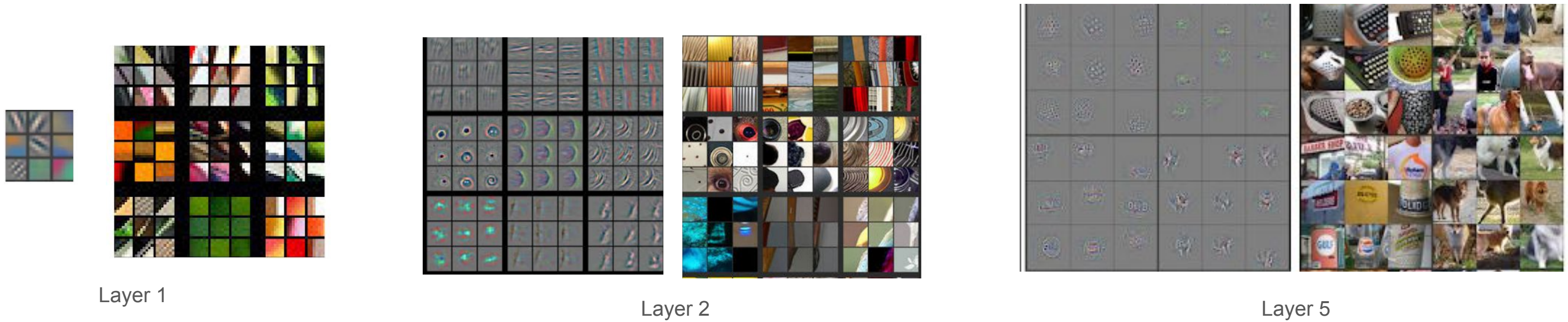
commercial-in-confidence

Left from: http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution || Right from: http://cs231n.github.io/convolutional-networks/

A Deeper Look at Convolutional Neural Networks: Features

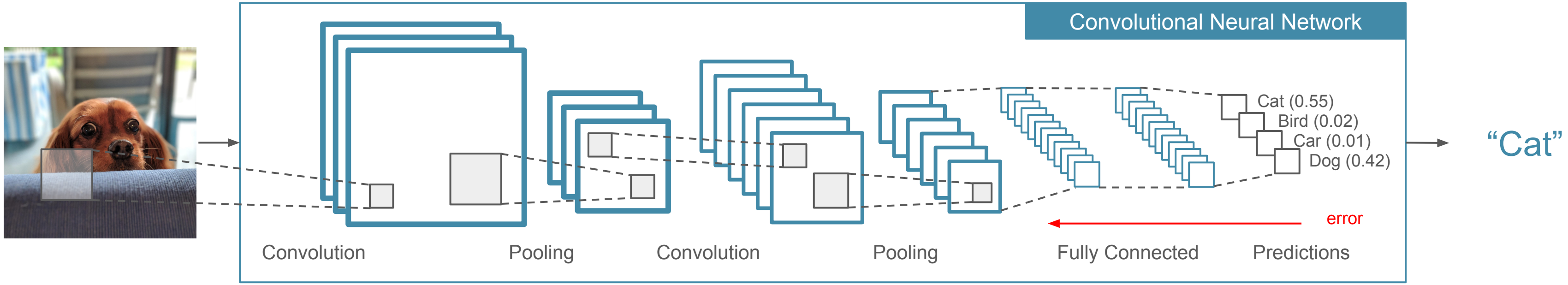


- Learns hierarchical features



Images: Matthew D Zeiler, Rob Fergus, "Visualizing and Understanding Convolutional Networks"

A Deeper Look at Convolutional Neural Networks: Training



- These models have huge numbers of parameters that need to be tuned, all of the convolutional filters, and all of the connections between the fully connected layers
- Training all of these parameters is done by using a method called backpropagation with gradient descent
- Pre-labelled images are fed to the network and predicted on, at the end of the network it's calculated how 'wrong' the network was using a loss function
- This amount of error is then back-propagated backwards through the network, slightly changing all the filter weights to be more correct for that example
- So over time we minimise the loss function across a large dataset of labelled examples
- Training needs many examples (thousands or even better millions) and takes a long time (days or even weeks) with heavy usage of GPU resources

So how Good are they at Classification?

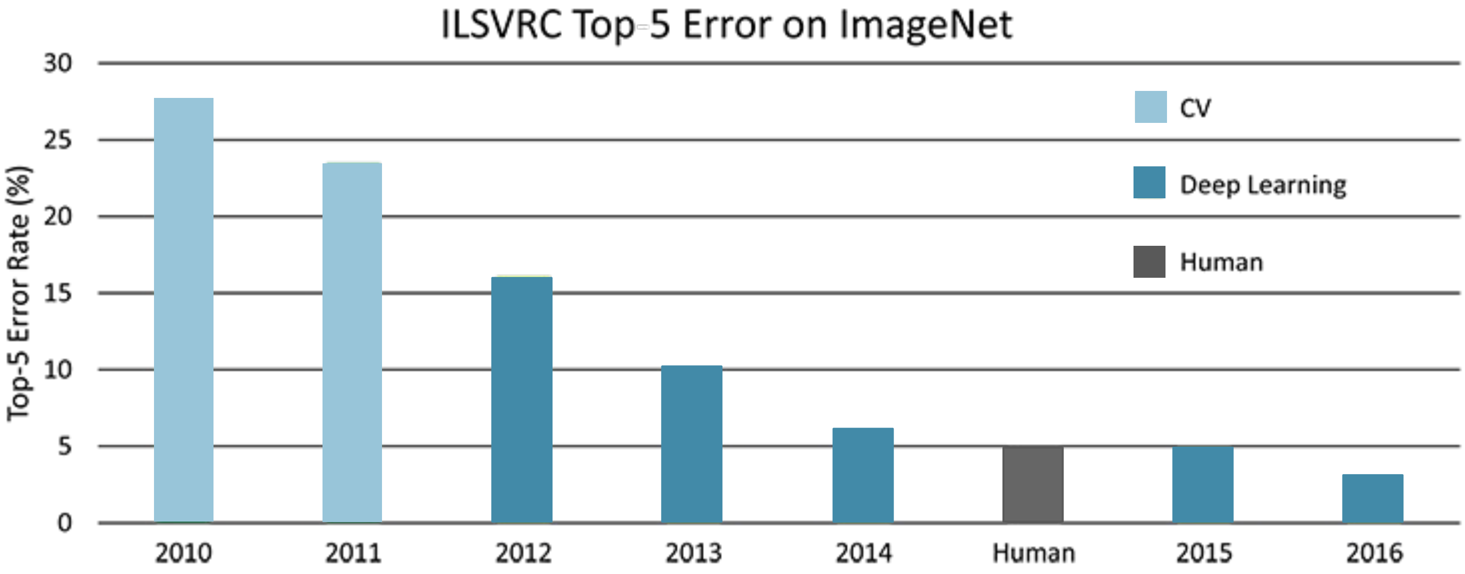


Image adapted from: https://www.dsiac.org/resources/journals/dsiac/winter-2017-volume-4-number-1/real-time-situ-intelligent-video-analytics

DATA TO DECISIONS CRC

Deep Learning Applications

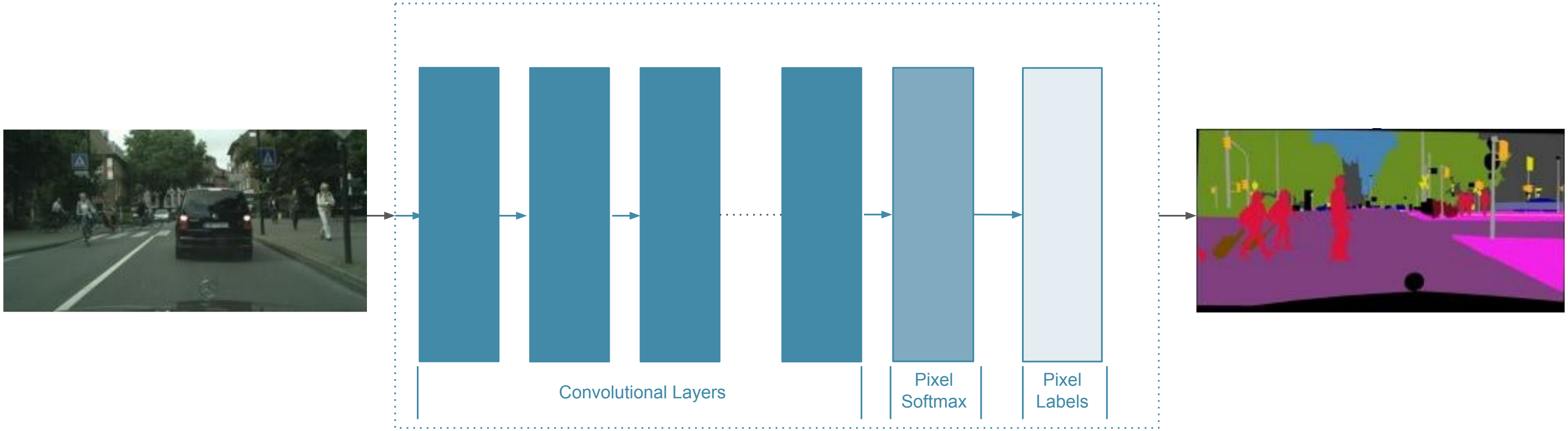
Presented by Adrian Johnston

Object Detection

- Rather than just classifying the images as “Car” or “Road” we can train the Neural Network to predict bounding boxes for the objects of interest
- State of the Art: Faster-RCNN, SSD, YOLO9000

Image:https://shaoanlu.wordpress.com/2017/05/07/vihicle-detection-usi ng-ssd-on-floybhub-udacity-self-driving-car-nano-degree/

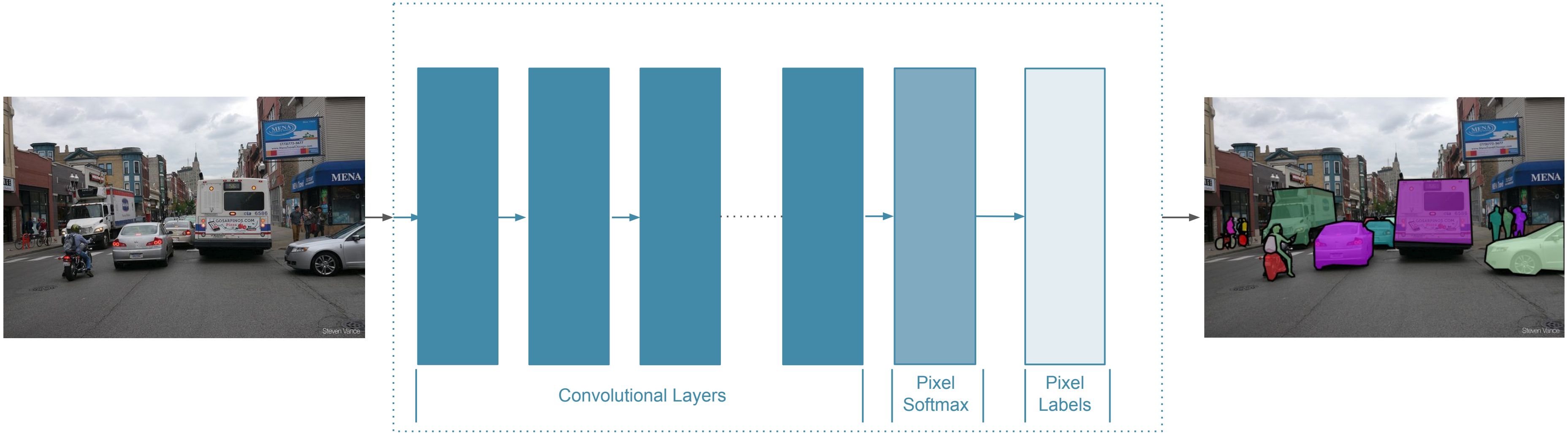
Semantic Segmentation



- We can also perform semantic segmentation
  - Train the network to classify each pixel in the image to separate sections into semantic classes e.g. Road, Car, Sky, Person

- State of the Art: FCN, SegNet, RefineNet, DeepLabv3, PSPNet

Instance Segmentation



- Instance Segmentation: Classify pixels to specific instances of a Category rather than just the semantic category
- One way is to combine Object Detection with Semantic Segmentation: Mask R-CNN

Instance Segmentation

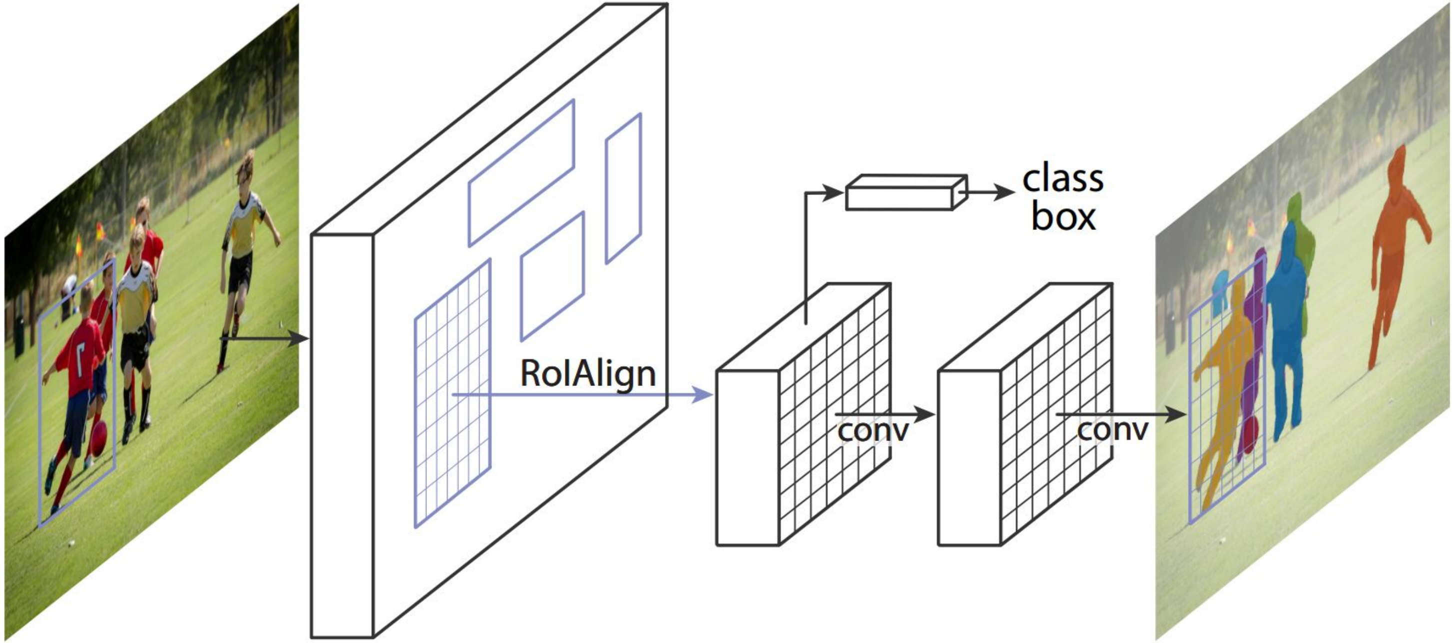


Figure 1. The **Mask R-CNN** framework for instance segmentation.

Image : Mask R-CNN, https://arxiv.org/pdf/1703.06870.pdf

Depth Estimation

Image: https://github.com/tinghuiz/SfMLearner/blob/master/misc/cityscapes_sample_results.gif

- We don’t always want to classify things
- Depth Regression: Predict the depth (continuous) per pixel in the image
- Supervised:
  - Capture ground truth depth from sensors:
    - Microsoft Kinect
    - Lidar
    - Stereo/Multi camera rig
  - Train the network to minimize the distance between the predicted depth and the ground truth depth from the sensor data
- Unsupervised using geometry:
  - Train the network to predict the depth given a video or stereo image with known or predicted camera pose
  - Difficult, but can be trained without ground truth depth

Simple Self Driving Car in GTA V

Image: https://github.com/Sentdex/pygta5/blob/master/self-driving-car-grand-theft-auto-5.gif

Imitation Learning on Real Data

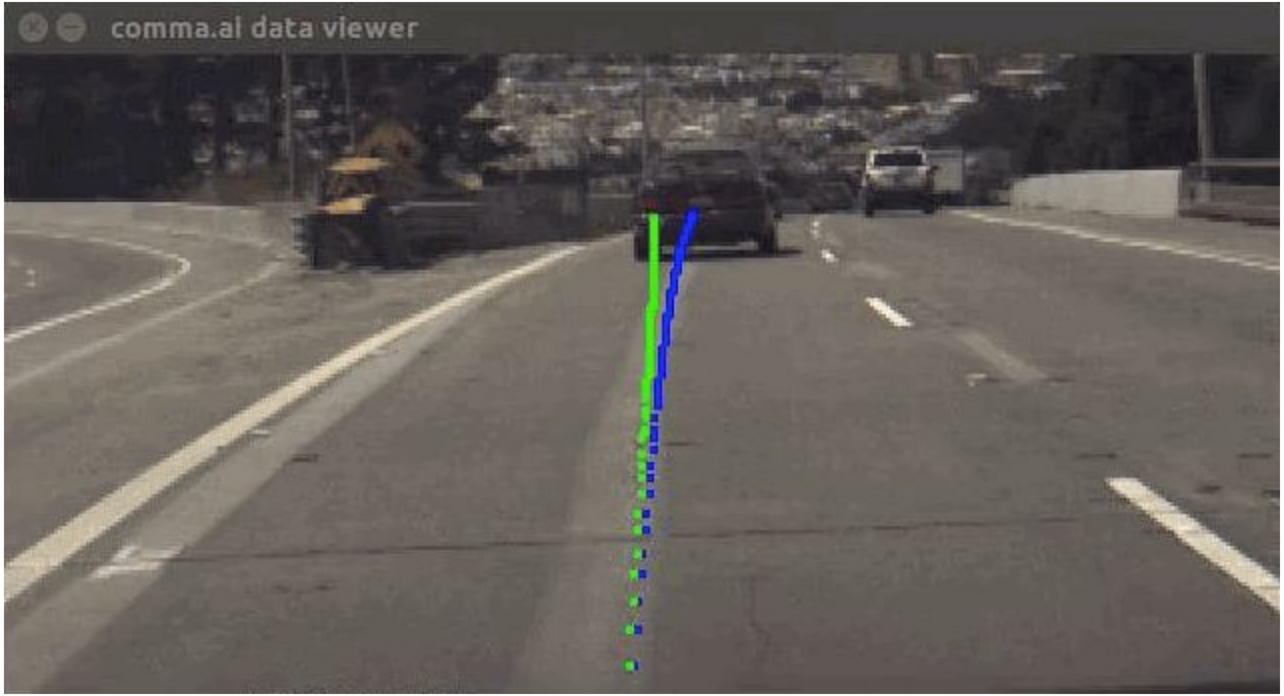
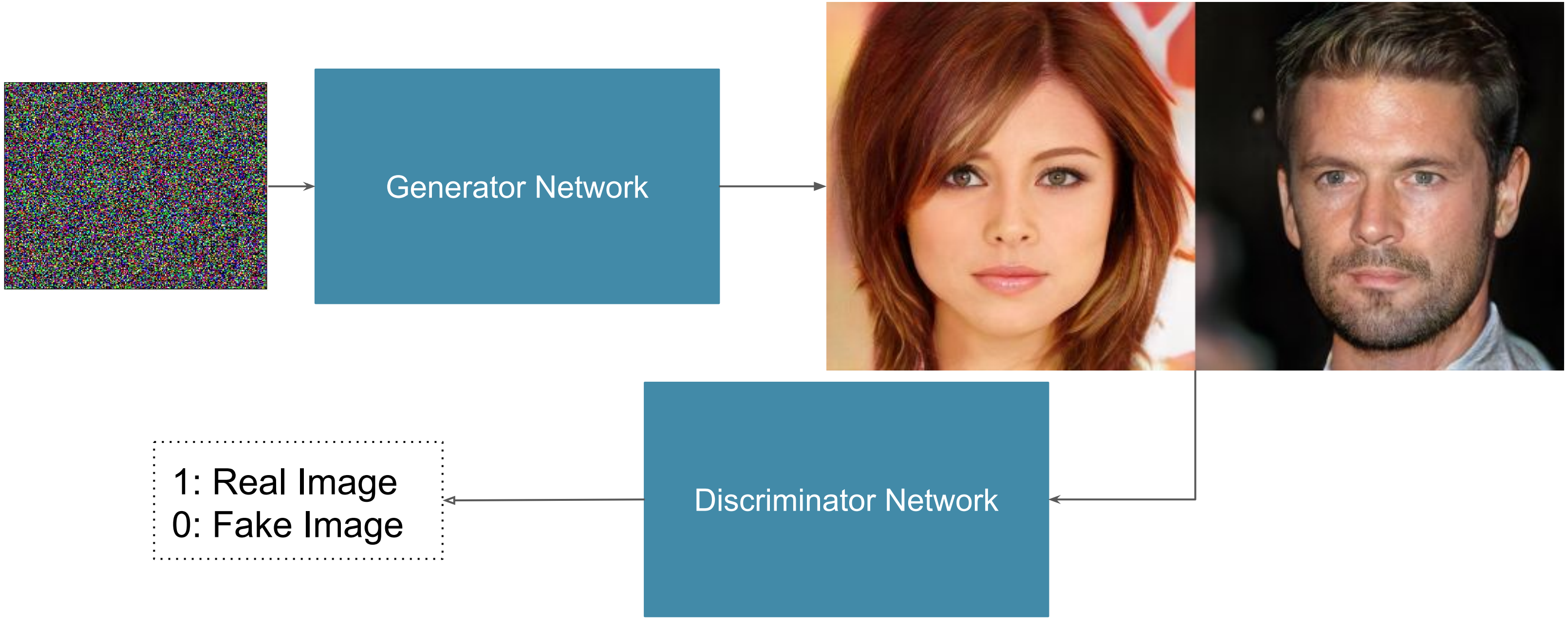
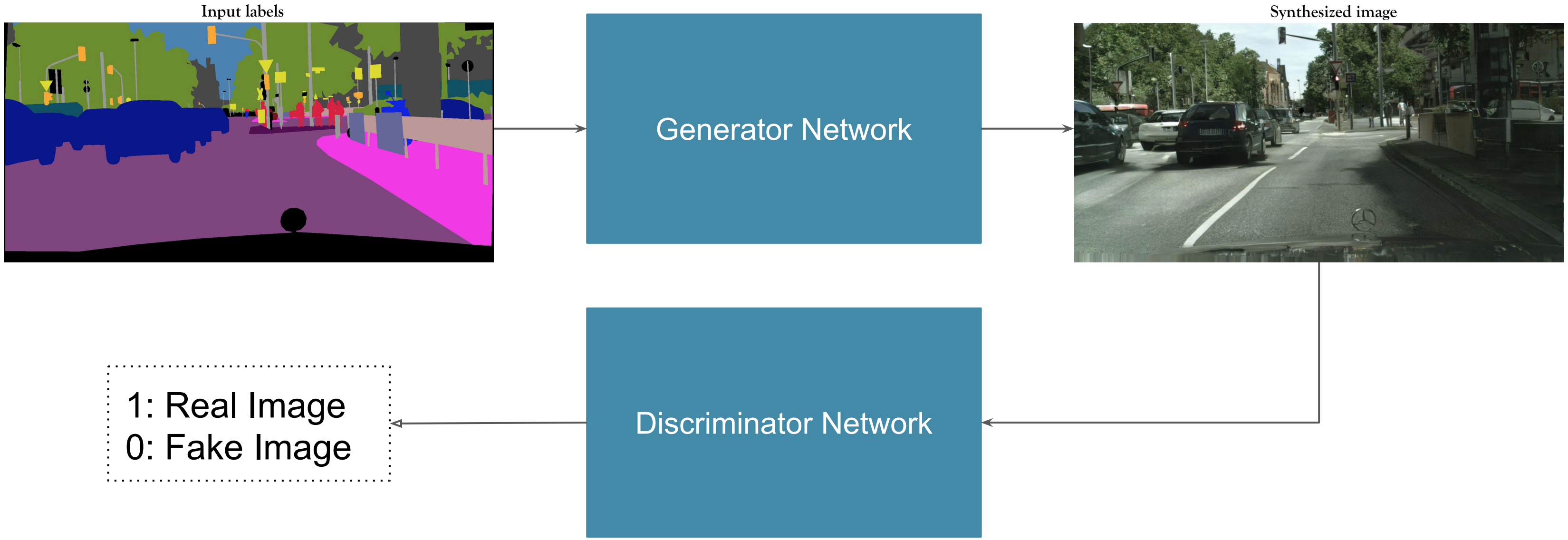


Image: https://github.com/commaai/research/blob/master/images/selfsteer.gif

Generative Adversarial Network (GAN)



Conditional GAN



Conditional GAN

So is Computer Vision a solved problem?

Is Computer Vision a solved problem?

- No! Lots of challenges still remain:
  - High Level Reasoning
    - E.g. Understanding how other drivers behave on the road
  - Interpretability
    - How do we interpret the decisions made by a AI system?
    - Useful after an accident
  - Uncertainty estimation
    - Teaching our models to “understand what they don’t know”

Other Challenges

- Data
  - These models are data hungry
    - Need thousands of examples
  - We have lots of data, but it still is not enough in lots of domains
  - Improved algorithms that can learn from smaller amounts of data
- Compute Resources
  - These models use immense amounts of computer resources
    - Graphics Processing Units (GPU's)
- Others

Thanks for Listening!