# Lecture 1 - Introducing the deep learning revolution

DD2424

March 17, 2019

# Why all the excitement about Deep Learning?

1. Astonishing empirical results.

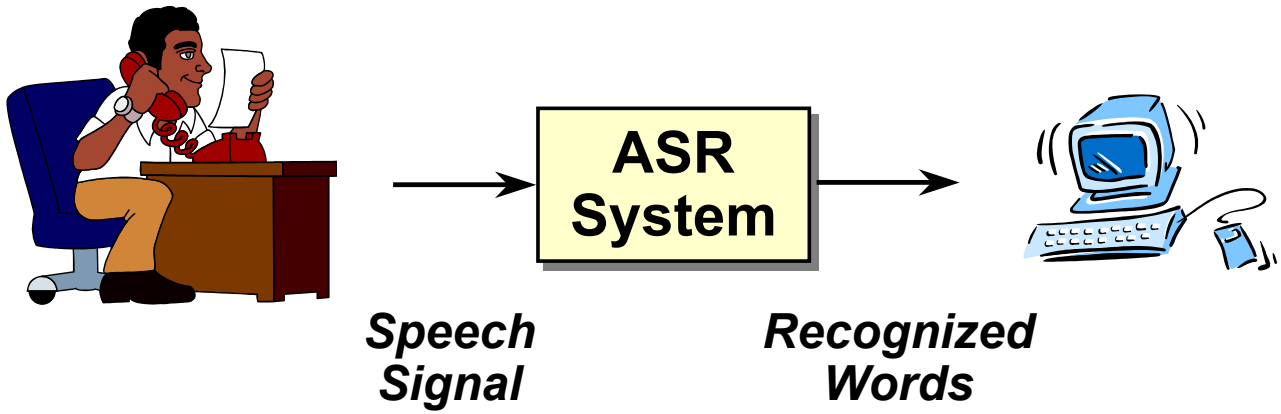
2. Similar solutions for different tasks in different domains.

3. General formula for improving results:

   - deeper network ($_{\text{upto a point}}$) +
   - more training data +
   - more computations.

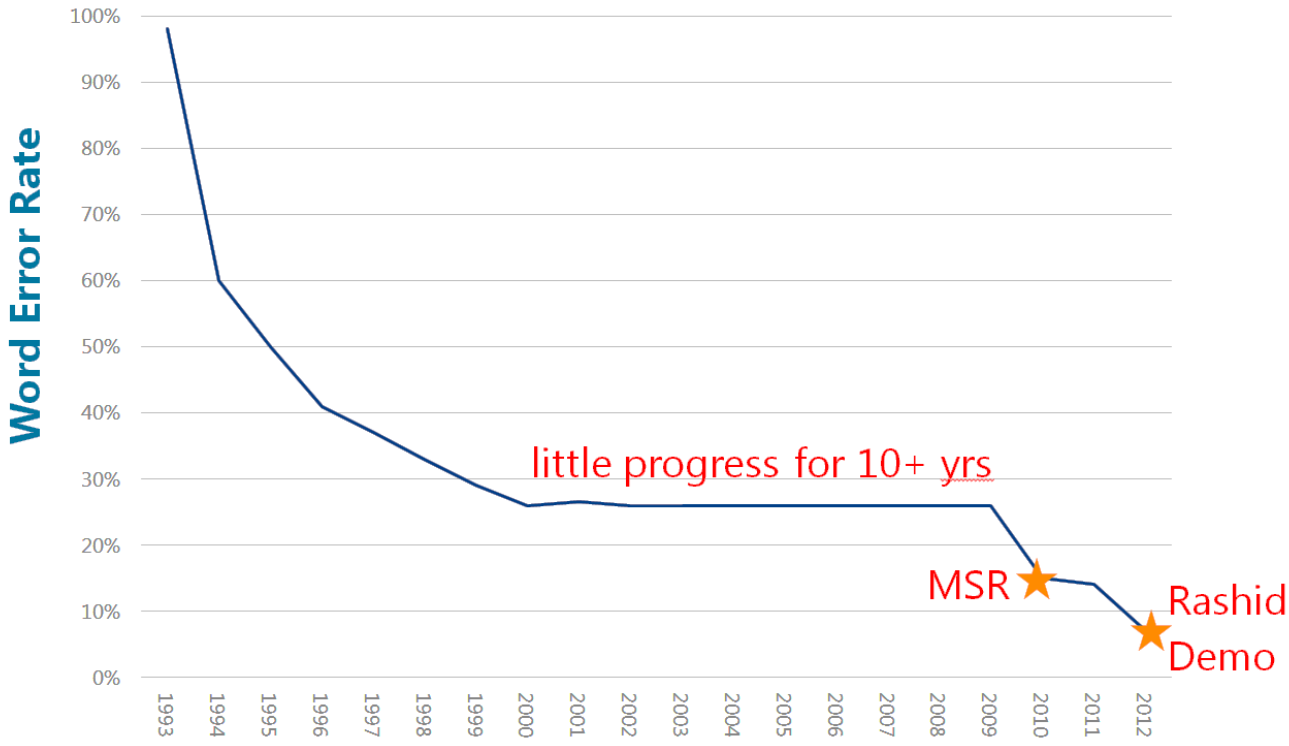
Snapshot of astonishing concrete results

# Speech: Spontaneous Speech Recognition




ASR system's challenge is to convert the speech signal into words.

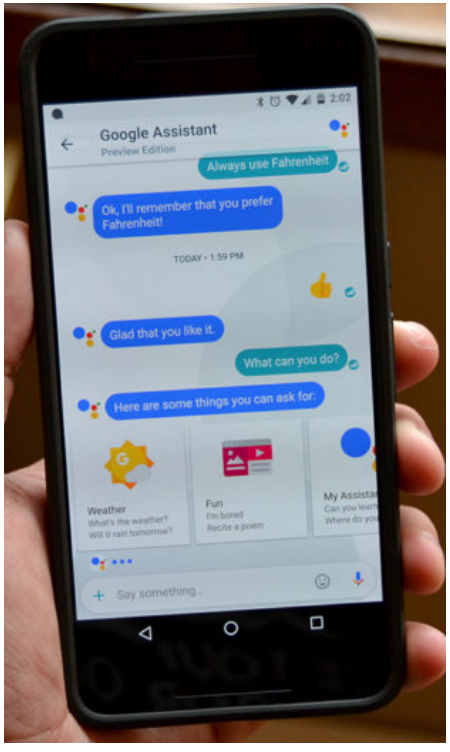
# Deep learning $\longrightarrow$ better speech recognition




After no improvement for 10+ years by the research community . . . deep learning brings large improvements to speech recognition.

amazon alexa


**Google's assistant**




Now we're having *conversations* with our phones and speakers.

# Computer Vision: Image Classification

ImageNet: Large Scale Visual Recognition Challenge

Steel drum

| **Output:** |
| --- |
| Scale |
| T-shirt |
| Steel drum |
| Drumstick |
| Mud turtle |

✔

| **Output:** |
| --- |
| Scale |
| T-shirt |
| Giant panda |
| Drumstick |
| Mud turtle |

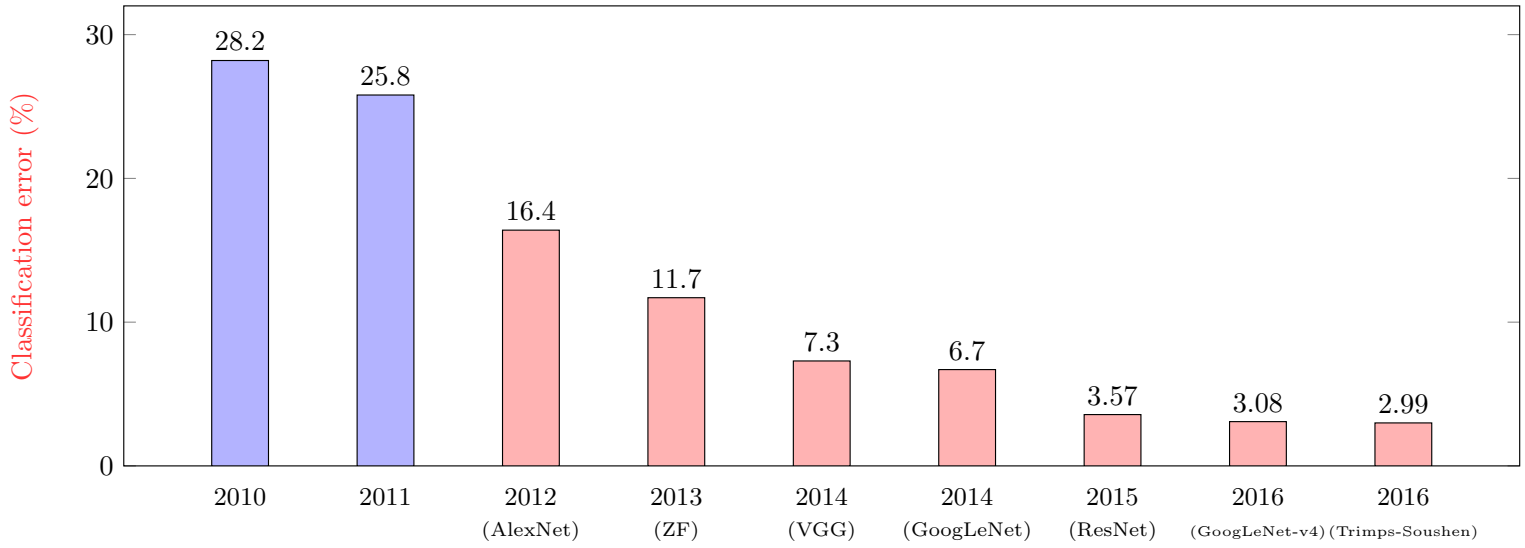
✗

$$\text{Error} = \frac{1}{100,000} \sum_{100,000 \text{ images}} 1(\text{incorrect on image } i)$$

**Source**: Detecting avocados to zucchinis: what have we done, and where are we going? O. Russakovsky et al., ICCV 2013

# Deep Learning $\rightarrow$ much better image classification

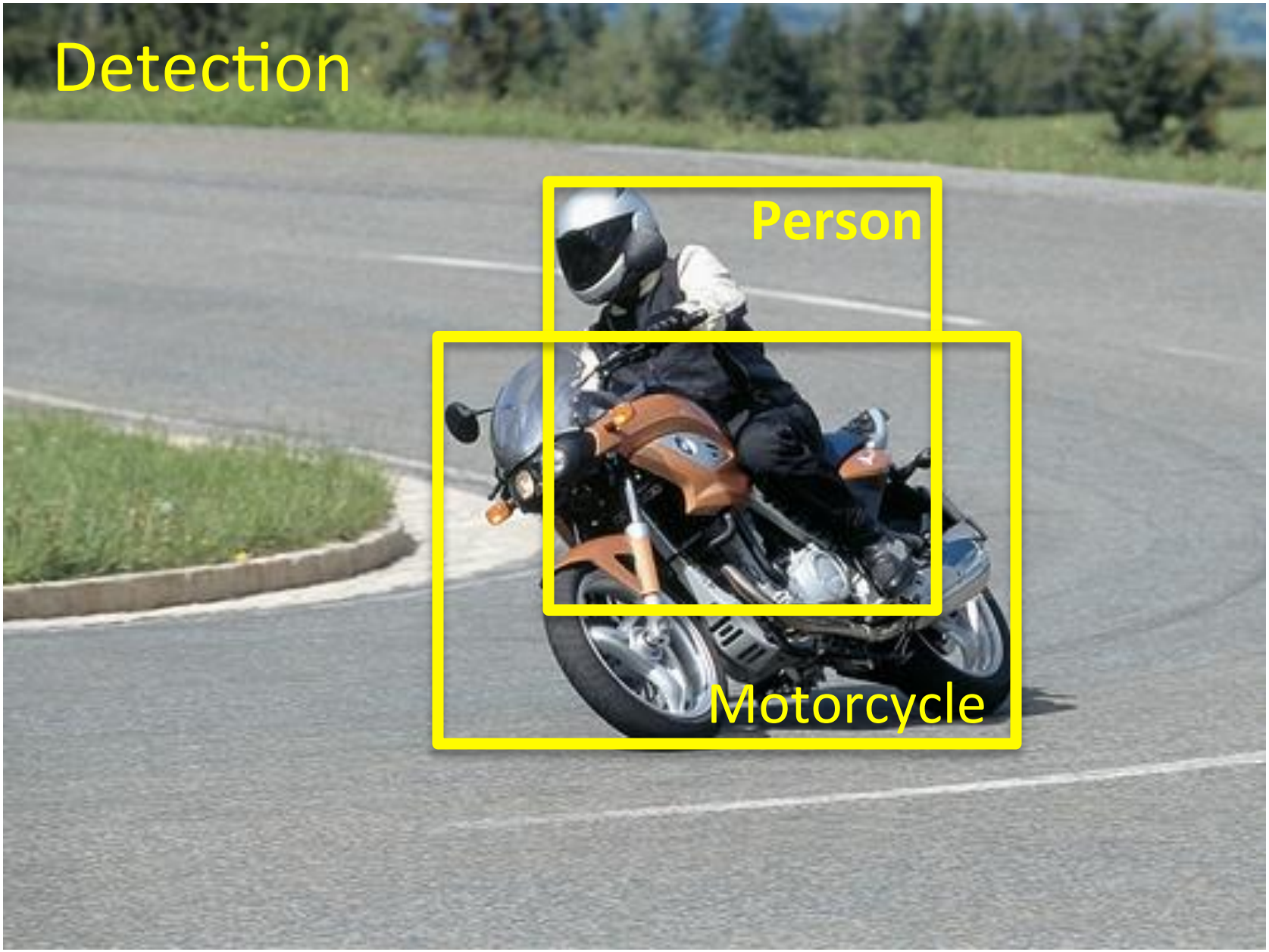
- ImageNet Large Scale Visual Recognition Challenge
- 1000 object classes, 1.4 million labelled training images



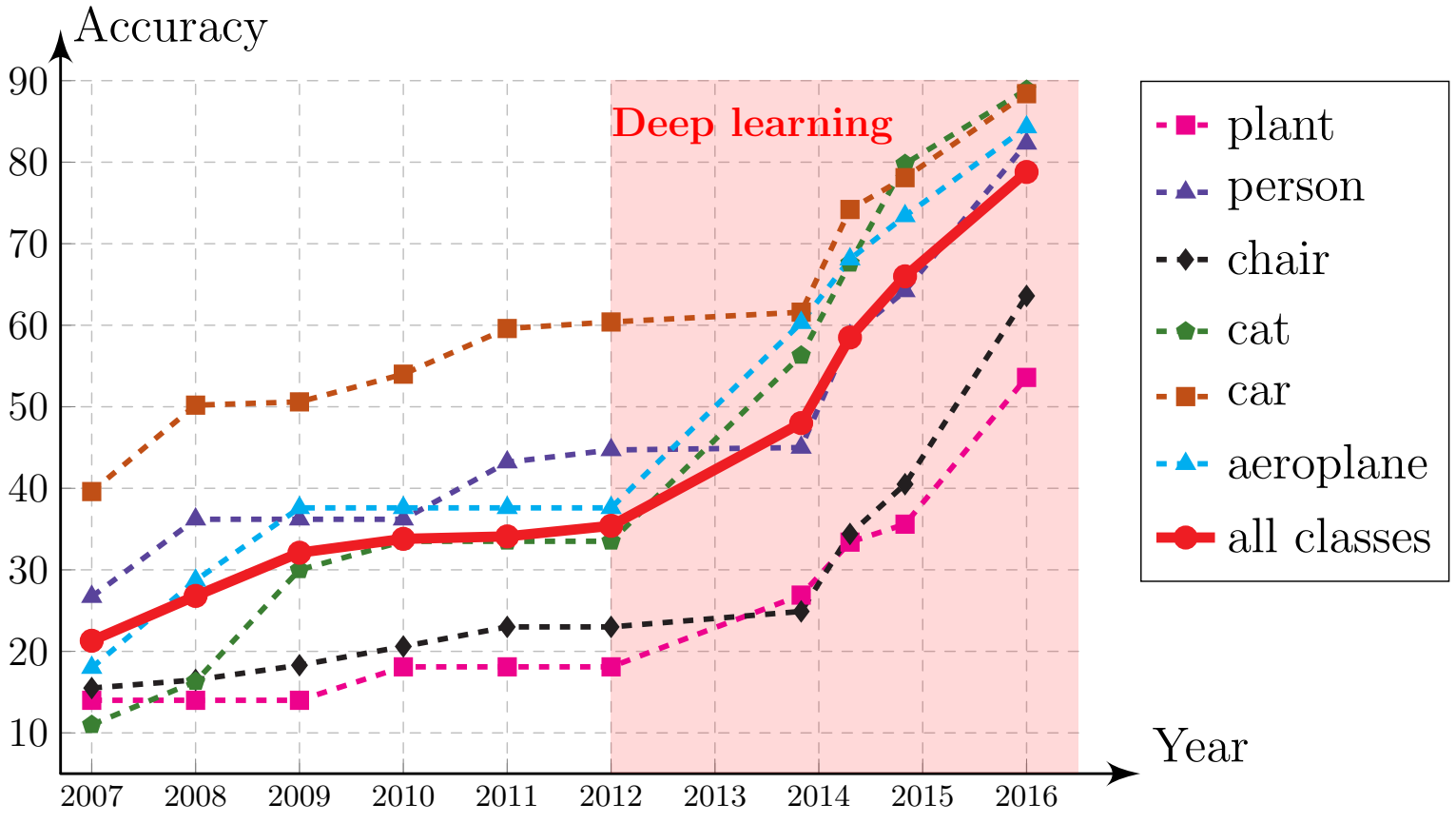

High performing systems on the ILSVRC datasets (2010-16).

Pink indicates a deep learning based solution.

Deep Learning (ConvNets) $\Longrightarrow$ great image classification

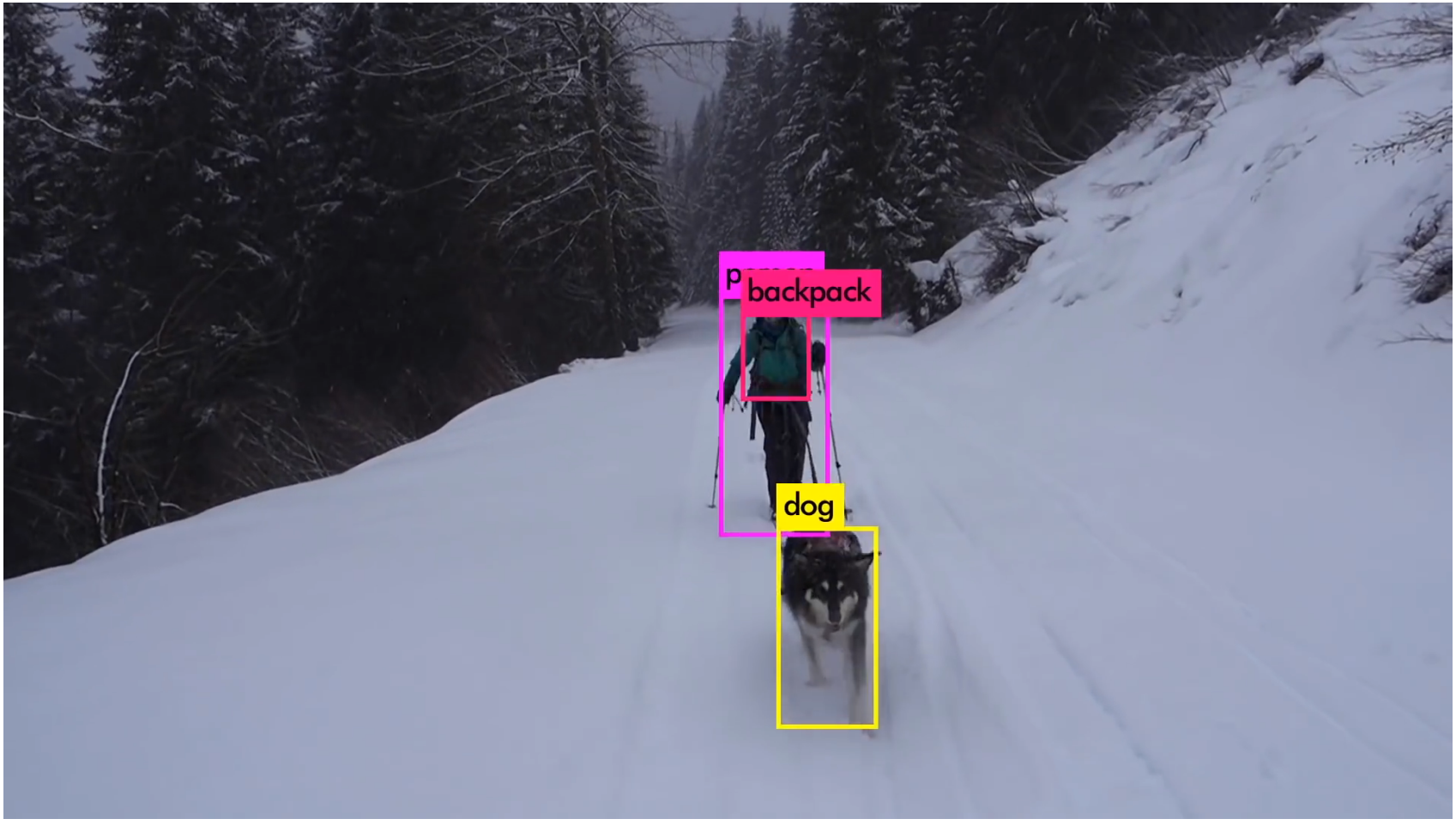

Detection
Person
Motorcycle

# Deep Learning → much better object detection




Progress of object detection for the Pascal VOC 2007 challenge.

# Deep Learning $\rightarrow$ object detection kinda works

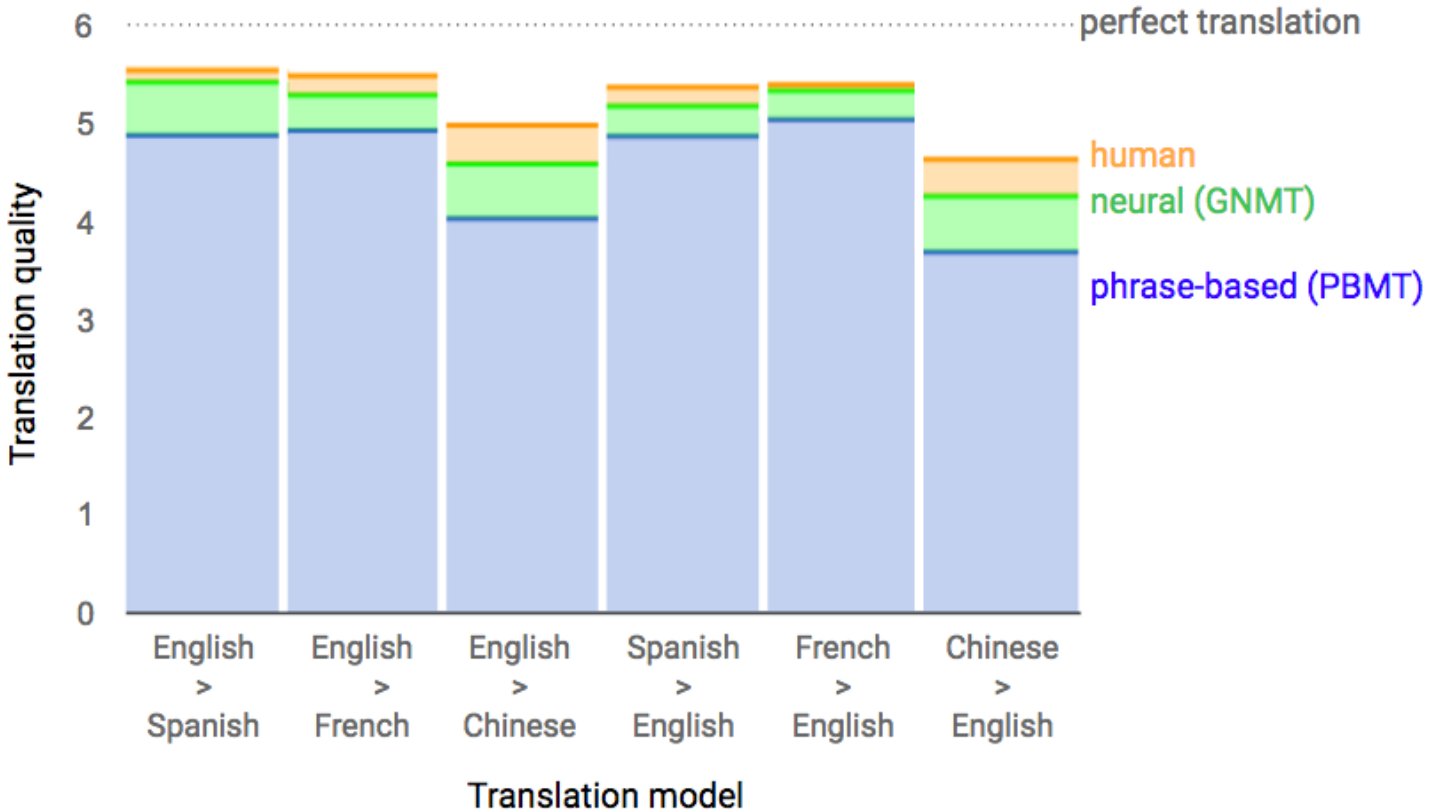


# Natural Language Processing: Text translation

- You have all probably used *Google Translate*
- It's been around for $\sim$10 years.
- Up until Autumn 2016 it used **Phrase-Based Machine Translation**.
- But now ...

# Deep Learning $\longrightarrow$ better machine translation

## Google Neural Machine Translation system




- Human raters compare the quality of translations of a source sentence.
- Scores range from 0 to 6.
  - 0 $\equiv$ nonsense translation
  - 6 $\equiv$ perfect translation

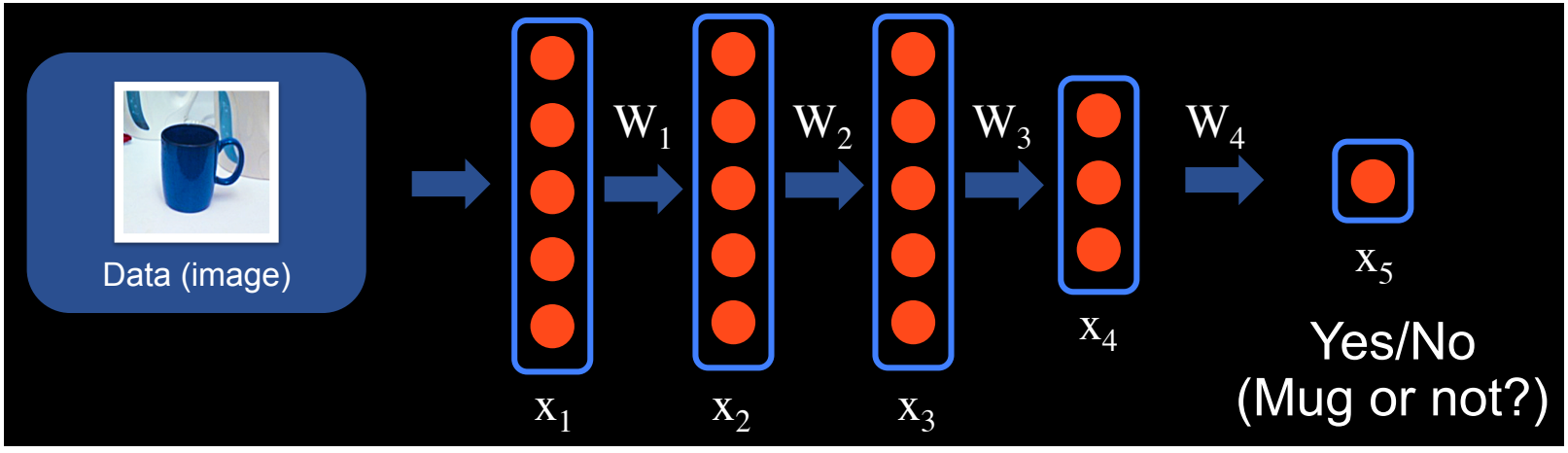
# Google NMT example of improvement

Which methods/networks are producing these results?

Which methods/networks are producing these results?

Neural Networks trained with lots of labelled data

# What is a neural network?




- Represents **non-linear function** from input to output space.
- Neural network functions have a deceptively simple form:

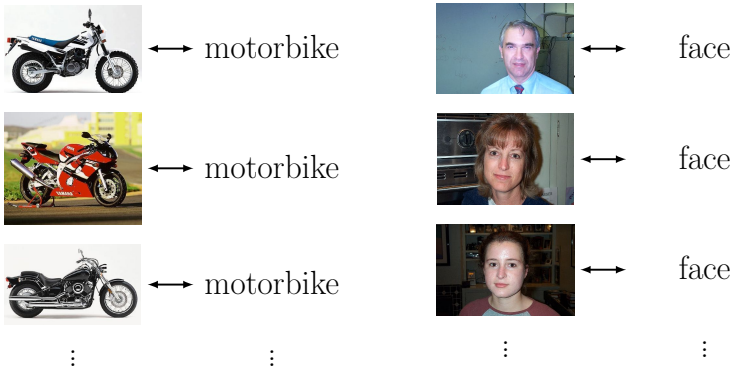
$$g(\mathbf{x}) = \sigma(f(W_{L-1} \cdots f(W_3\, f(W_2\, f(W_1 \mathbf{x})) \cdots ))$$

- Repeatedly apply:

*linear transformation* + *simple non-linear function*

# Deep Learning approach to supervised learning

- Given labelled training data




- Learning equals finding values for $W_1, W_2, \ldots, W_L$ s.t.

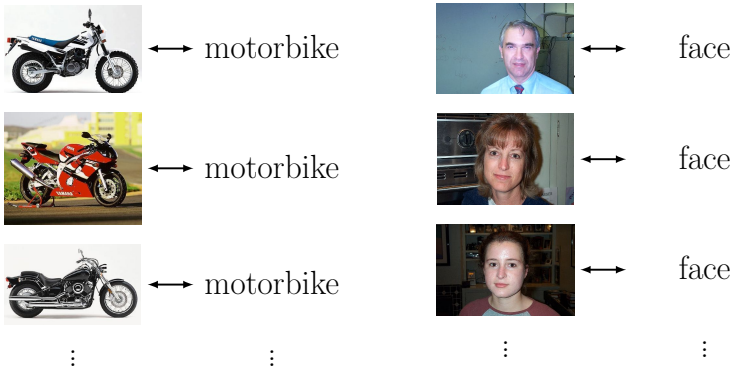
$$g\left(\;\right) \approx \begin{pmatrix}0\\1\end{pmatrix},\; g\left(\;\right) \approx \begin{pmatrix}0\\1\end{pmatrix},\; g\left(\;\right) \approx \begin{pmatrix}0\\1\end{pmatrix} \cdots$$

$$g\left(\;\right) \approx \begin{pmatrix}1\\0\end{pmatrix},\; g\left(\;\right) \approx \begin{pmatrix}1\\0\end{pmatrix},\; g\left(\;\right) \approx \begin{pmatrix}1\\0\end{pmatrix} \cdots$$

- The more labelled training data and computational resources
  - the better you learn $W_1, W_2, \ldots, W_L$,
  - the faster you learn $W_1, W_2, \ldots, W_L$ and
  - the larger $L$ can be.

# Deep Learning approach to supervised learning

- Given labelled training data




- Learning equals finding values for $W_1, W_2, \ldots, W_L$ s.t.

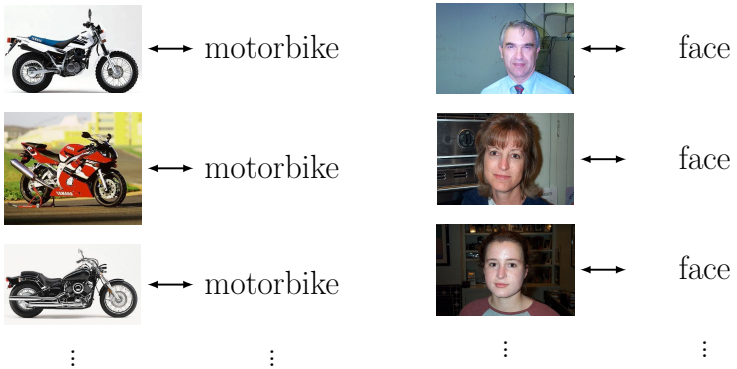
$$g\left(\text{  }\right) \approx \begin{pmatrix}0\\1\end{pmatrix},\ g\left(\text{  }\right) \approx \begin{pmatrix}0\\1\end{pmatrix},\ g\left(\text{  }\right) \approx \begin{pmatrix}0\\1\end{pmatrix}\ \cdots$$

$$g\left(\text{  }\right) \approx \begin{pmatrix}1\\0\end{pmatrix},\ g\left(\text{  }\right) \approx \begin{pmatrix}1\\0\end{pmatrix},\ g\left(\text{  }\right) \approx \begin{pmatrix}1\\0\end{pmatrix}\ \cdots$$

- The more labelled training data and computational resources
  - the better you learn $W_1, W_2, \ldots, W_L$,
  - the faster you learn $W_1, W_2, \ldots, W_L$ and
  - the larger $L$ can be.

# Deep Learning approach to supervised learning

- Given labelled training data




- Learning equals finding values for $W_1, W_2, \ldots, W_L$ s.t.

$$g\left(\text{  }\right) \approx \begin{pmatrix}0\\1\end{pmatrix},\ g\left(\text{  }\right) \approx \begin{pmatrix}0\\1\end{pmatrix},\ g\left(\text{  }\right) \approx \begin{pmatrix}0\\1\end{pmatrix}\ \cdots$$

$$g\left(\text{  }\right) \approx \begin{pmatrix}1\\0\end{pmatrix},\ g\left(\text{  }\right) \approx \begin{pmatrix}1\\0\end{pmatrix},\ g\left(\text{  }\right) \approx \begin{pmatrix}1\\0\end{pmatrix}\ \cdots$$

- The more labelled training data and computational resources
  - the better you learn $W_1, W_2, \ldots, W_L$,
  - the faster you learn $W_1, W_2, \ldots, W_L$ and
  - the larger $L$ can be.

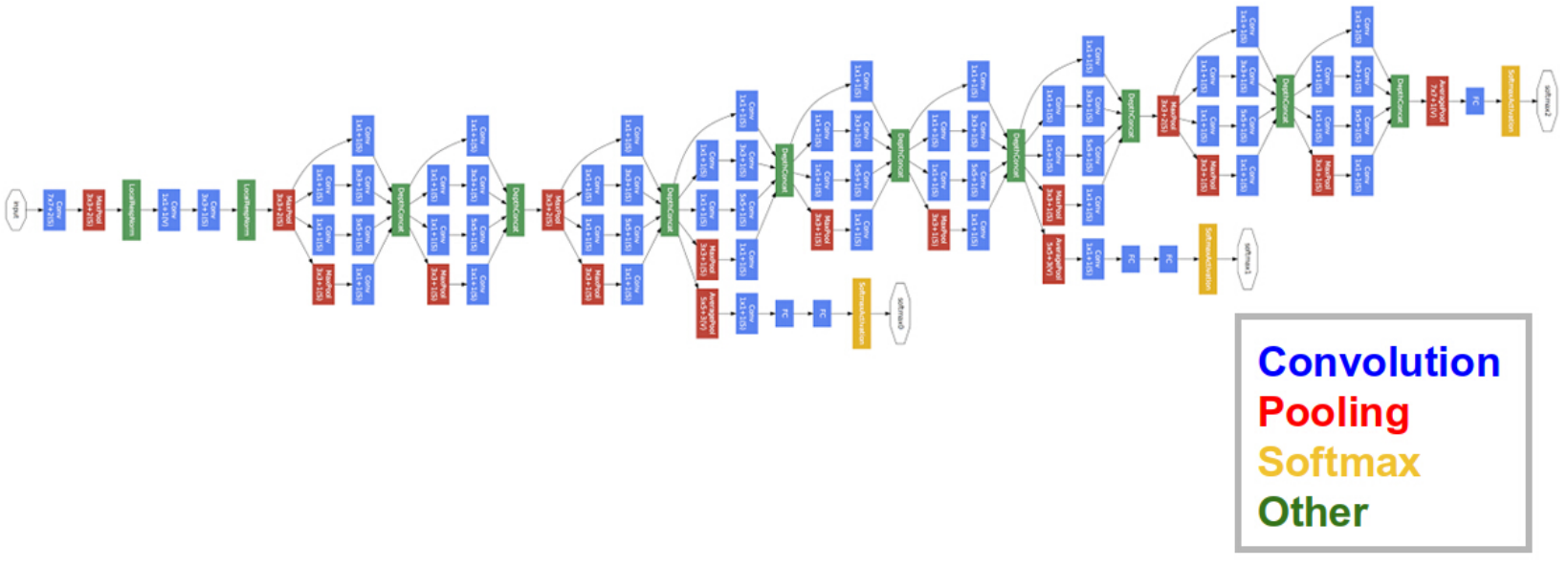
Which methods/networks are producing these results?

Which methods/networks are producing these results?

**Deep** Neural Networks trained with lots of labelled data

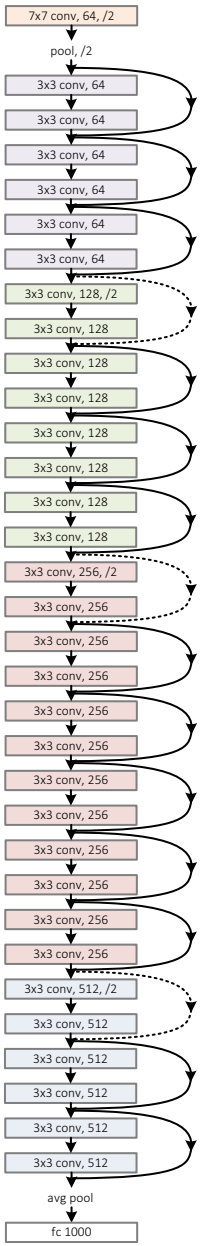
# Example modern networks

**GoogLeNet (2014)**



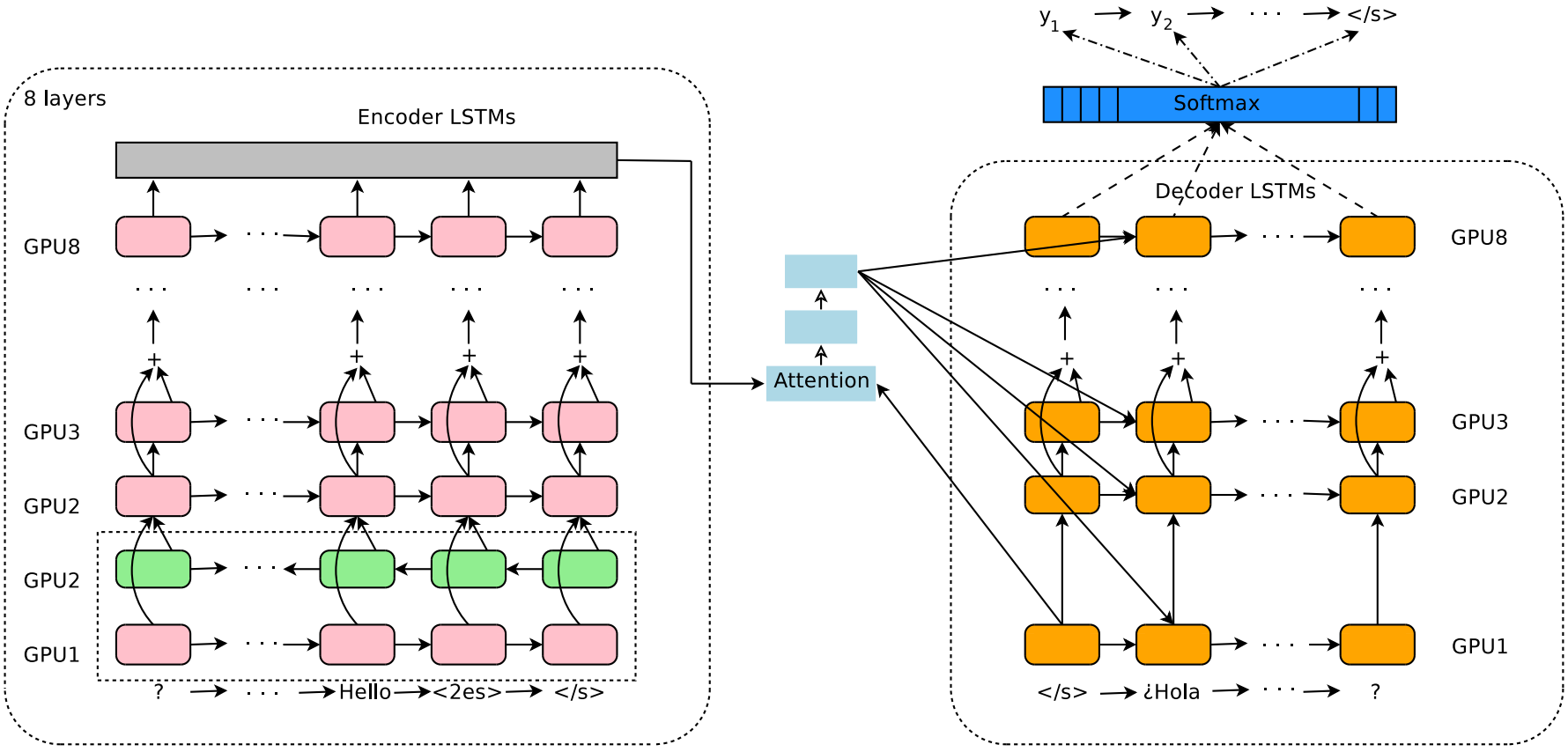

Trained using ImageNet to perform image classification.

# Example modern networks




- **ResNet** - a convolutional neural network with skip connections.
- Introduced in **Deep Residual Learning for Image Recognition**, by He, Zhang, Ren, Sun, CVPR 2016
- Trained for image classification, but similar structures have been transferred to image generation, speech recognition, NLP, ...

# Example modern networks

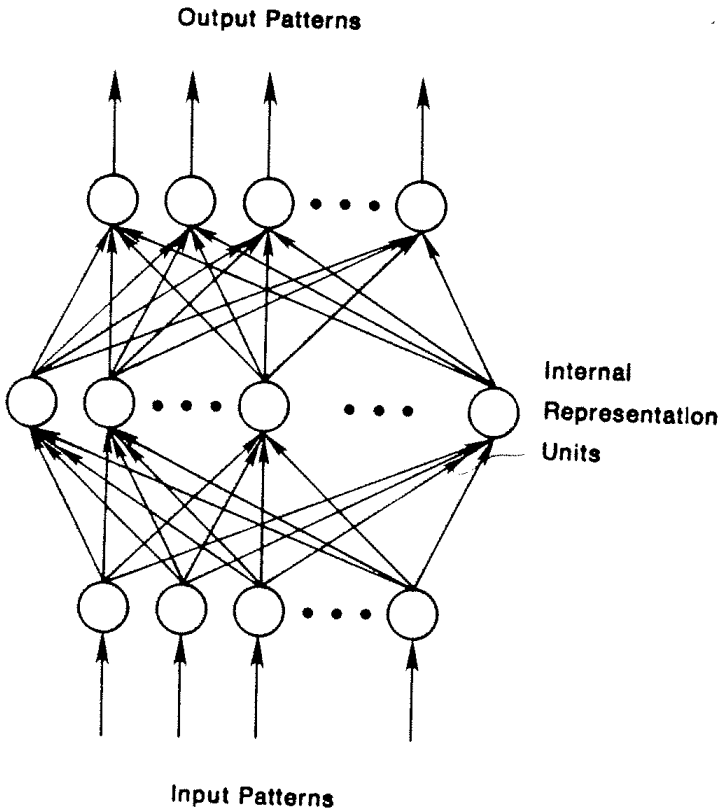



**Google's Multilingual Neural Machine Translation system**

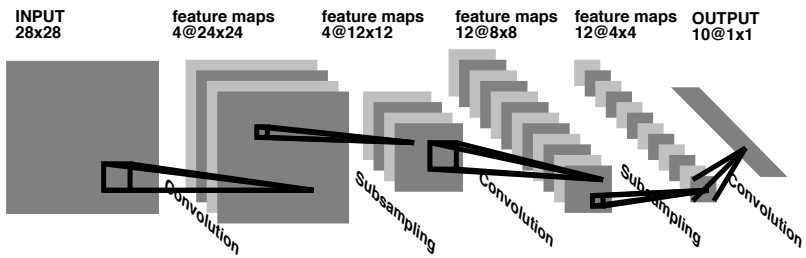
# Why the great successes now?

**Question**:

Haven't these basic networks and their training algorithms been around for decades?




Back-prop Rumelhart in '86




LeCun's LeNet-1 '90

# Why the great successes now?

**Question**:
Haven't these basic networks and their training algorithms been around for decades?

**Answer**:
Yes but now have

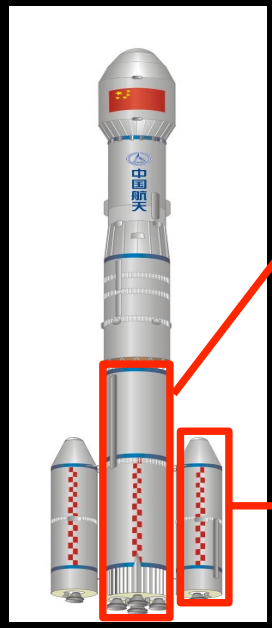
1. Explosion of labelled digital data available for training.
2. Deeper networks (networks with more layers)
3. Better understanding & procedures for learning the parameters of the networks.
4. GPUs $\Longrightarrow$ can exploit above to train deep networks in a "*reasonable*" time.

# Why is Deep Learning taking off?

Baidu Research

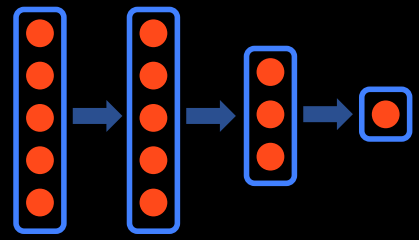
Andrew Ng

# Why is Deep Learning taking off?



Engine

Fuel

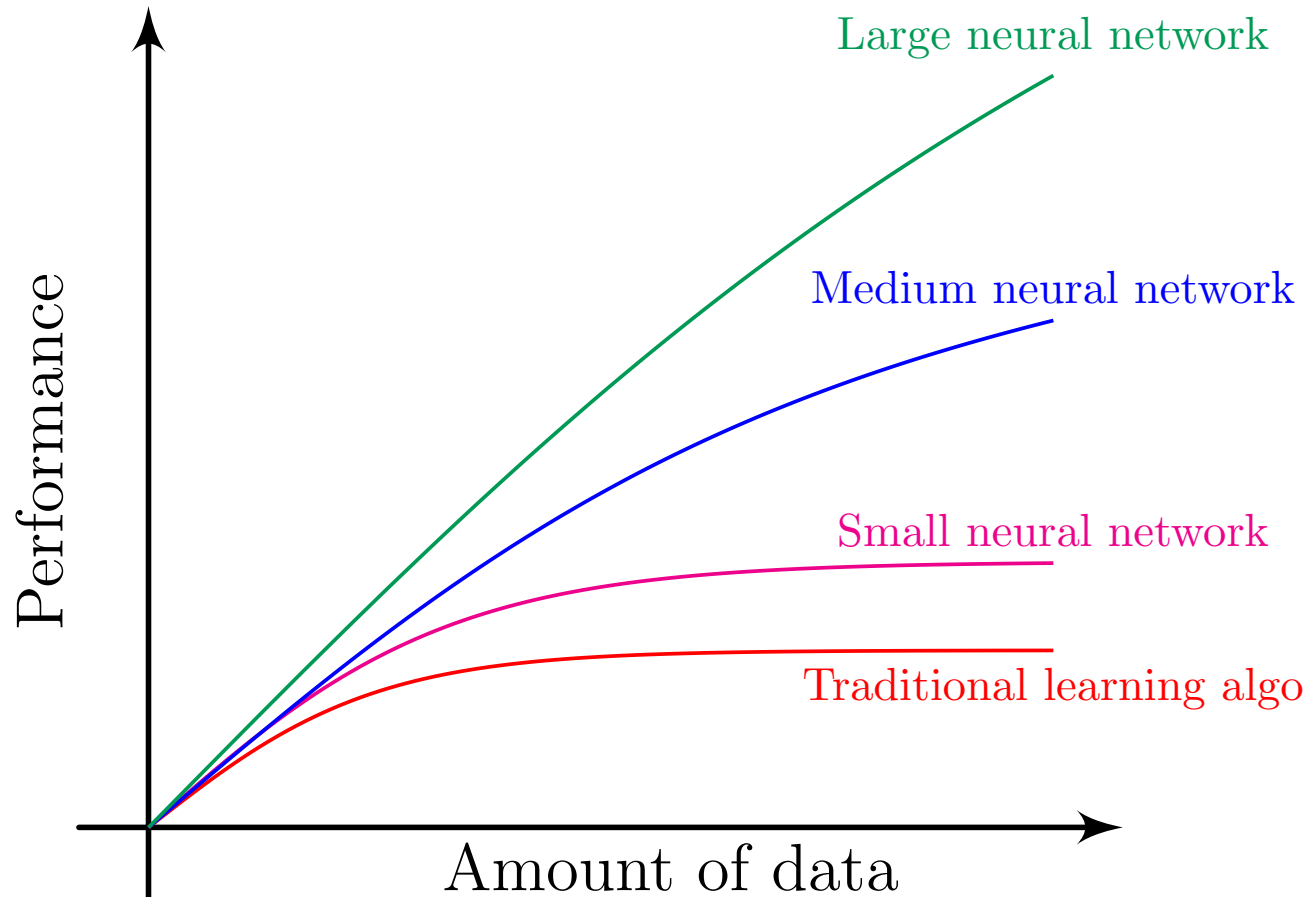


Large neural networks


Data

Why didn't other approaches exploit these developments?

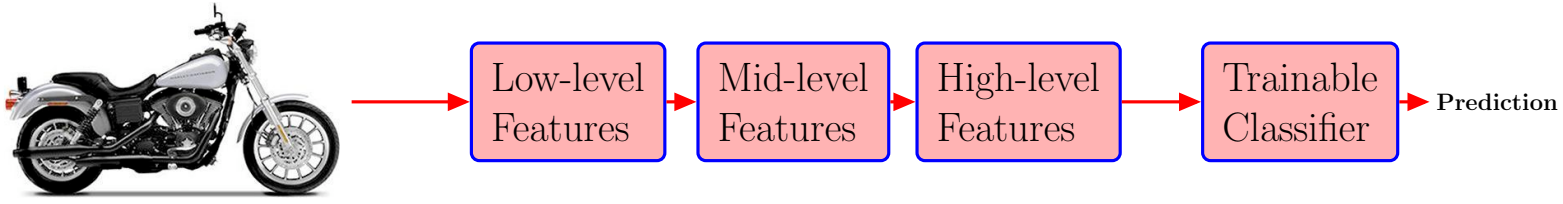
# Other methods couldn't take advantage of large datasets



# Reason 1: Deep Learning does end-to-end training

- Learn everything from the raw input data to desired output.

- Train hierarchical representations




as opposed to prior approach of hand engineering features


- Engineers job transferred from signal processing to network architecture & training algorithms.

# Reason 2: Deep networks have efficient high capacity

- A neural network is a function $g$

$$g : \text{input space} \rightarrow \text{output space}$$

- **Universal approximation** (both shallow & deep):
  Given a sufficient number of hidden nodes a 2-layer network can approximate any function.

- **Shallow networks not efficient representation**:
  Some functions compactly represented with $k$ layers in a network may require exponential size with 2 layers.

- $\Longrightarrow$ Deep networks are frequently a much more efficient representation of complicated functions than their shallow counterparts.

# Reason 2: Deep networks efficient high capacity

- Deep network exploit **Compositionality**.

  *Complicated features are combinations of smaller, simpler features.*

- Compositional features give an exponential gain in representional power over shallow representations.

- Compositionality is useful to describe the world around us efficiently.

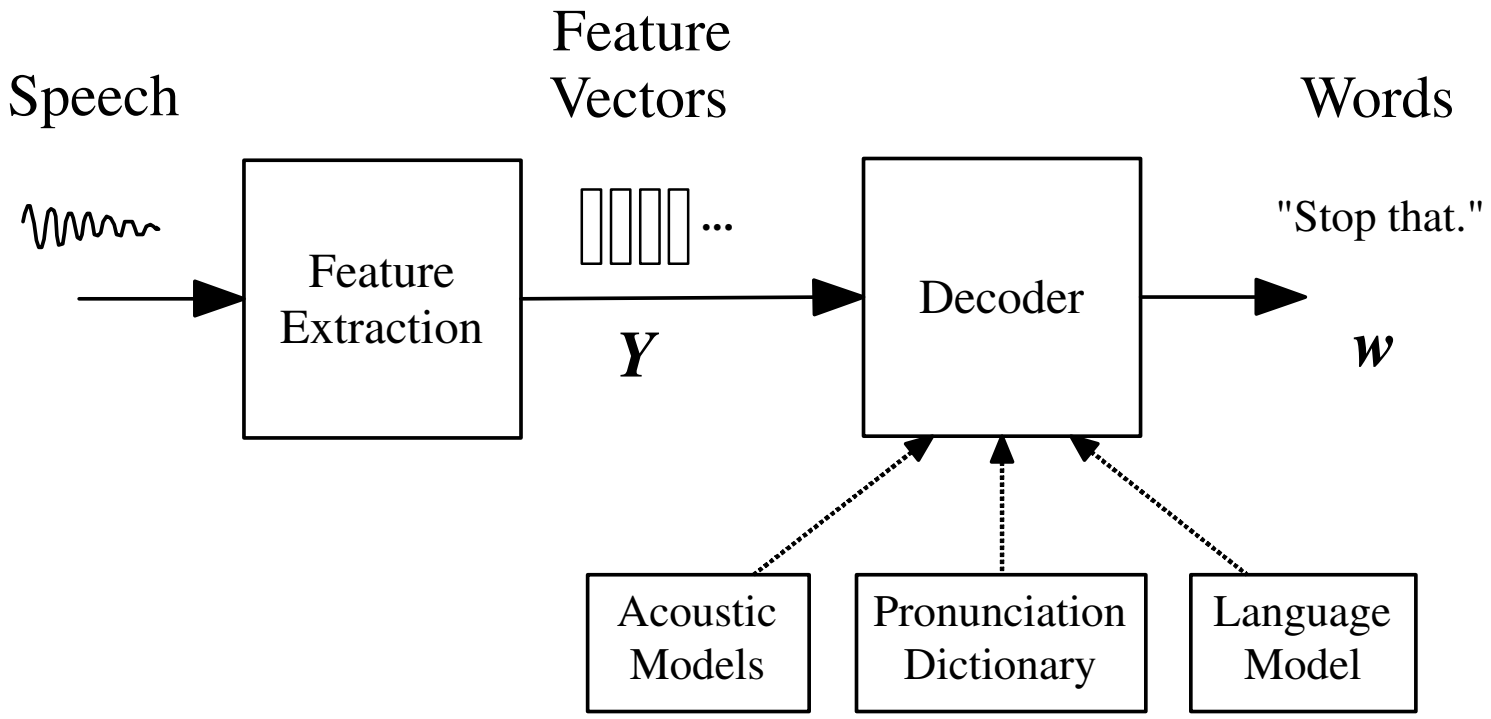
First Success Story of Deep Learning: **Speech Recognition**

# Problems in speech processing & recognition

- Speech (continuous time series) $\rightarrow$ Speech (continuous time series)
  - Speech Enhancement, Voice Conversion
- Speech (continuous time series) $\rightarrow$ Text (discrete symbol sequence)
  - Automatic speech recognition (ASR), Voice Activity Detection (VAD)
- Text (discrete symbol sequence) $\rightarrow$ Speech (continuous time series)
  - Text-to-speech synthesis (TTS)
- Text (discrete symbol sequence) $\rightarrow$ Text (discrete symbol sequence)
  - Machine translation (MT)

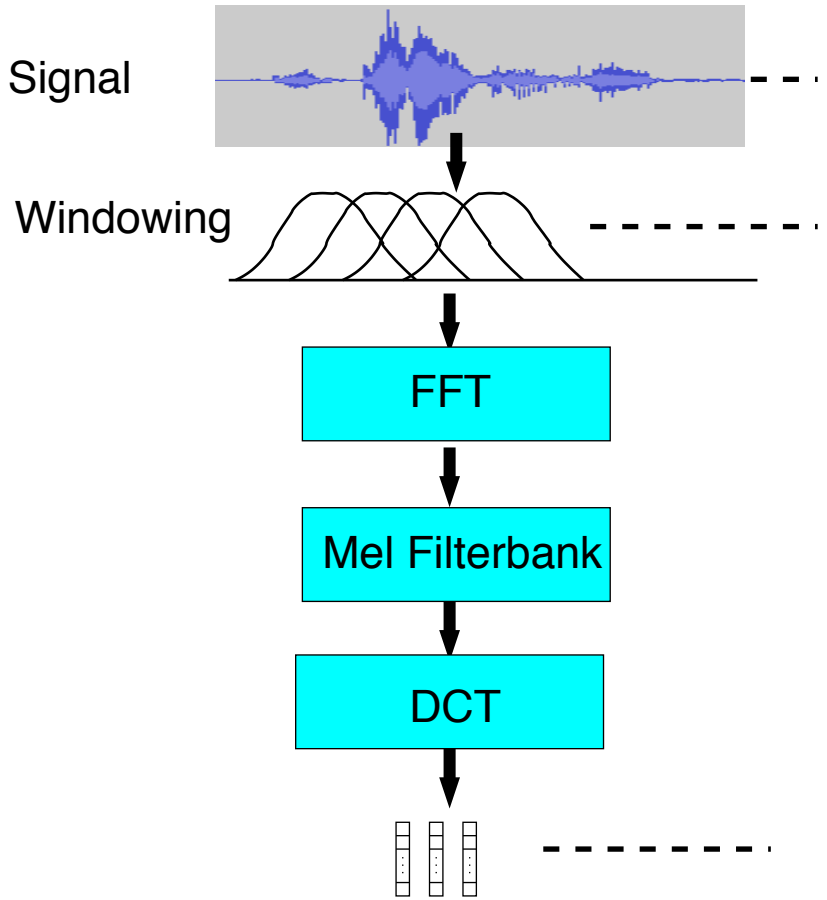
**Will present a short history of recent developments in ASR.**

# State-of-the-Art circa 2009




*The Application of Hidden Markov Models in Speech Recognition* by Mark Gales and Steve Young, Foundations and Trends in Signal Processing Vol. 1, No. 3, 2007.

# State-of-the-Art circa 2009




Common hand-crafted feature descriptors:

- **MFCCs**: *mel-frequency cepstral coefficients* (figure above shows processing)
- **PLP**: *perceptual linear prediction*

Given a sequence of fixed size acoustic vectors

$$\mathbf{Y} = \mathbf{y}_1, \ldots, \mathbf{y}_T$$

the decoder attempts to find the sequence of words

$$\mathbf{w}_{1:L} = w_1, \ldots w_L$$

by solving this optimization problem:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} p(\mathbf{Y} \mid \mathbf{w})\, P(\mathbf{w})$$

where

- $p(\mathbf{Y} \mid \mathbf{w})$ is determined by the **acoustic model** and
- $P(\mathbf{w})$ is determined by a **language model**.

## Acoustic Model (Basic model but gives the idea)

- Each word $w$ decomposed into a sequence of $K_w$ *phones*: $\mathbf{q}^{(w)} = q_1, \ldots, q_{K_w}$.

- The likelihood is computed over multiple possible pronunciations

$$p(\mathbf{Y} \mid \mathbf{w}) = \sum_{\mathbf{Q}} p(\mathbf{Y} \mid \mathbf{Q})\, P(\mathbf{Q} \mid \mathbf{w})$$

  where each $\mathbf{Q}$ contains a possible pronunciation for each word in $\mathbf{w}$.

- Given $\mathbf{Q}$ the set of possible phones then

$$p(\mathbf{Y} \mid \mathbf{Q}) = \sum_{\boldsymbol{\theta}} p(\boldsymbol{\theta}, \mathbf{Y} \mid \mathbf{Q})$$

  where $\boldsymbol{\theta} = \theta_0, \theta_1, \ldots, \theta_T$ represents the phone states at the observations $\mathbf{y}_1, \ldots, \mathbf{y}_T$.

## HMM Acoustic Model (Basic model but gives the idea)

- Each word utterance is modelled by a HMM.

- **Word HMM**: Likelihood of an observation given phone state

$$p(\mathbf{y} \mid q = s_i) = \text{GMM}(\mathbf{y}; \boldsymbol{\mu}_{il}, \ldots, \boldsymbol{\mu}_{iK}, \Sigma_{il}, \ldots, \Sigma_{iK}) \quad \leftarrow \text{Gaussian Mixture Model}$$

- The transition probabilities between states is

$$P(q_t = s_j \mid q_{t-1} = s_i) = a_{ij}$$

- A composite HMM constructed for the total sequence.

- The **acoustic likelihood** is then given by:

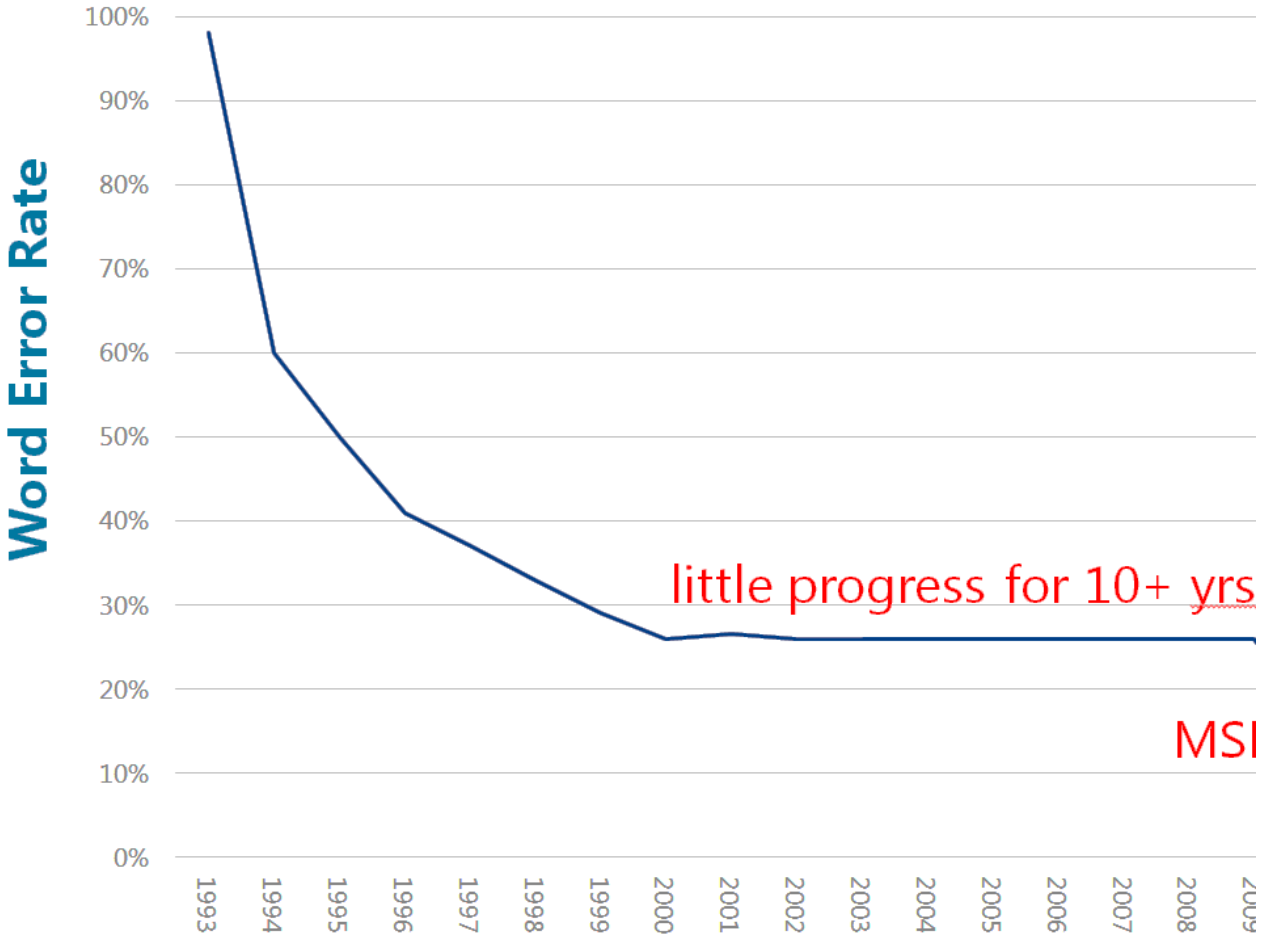
$$p(\boldsymbol{\theta}, \mathbf{Y} \mid \mathbf{Q}) = P(\theta_0) \prod_{t=1}^{T} P(\theta_t \mid \theta_{t-1})\, p(\mathbf{y}_t \mid q = \theta_t)$$

# State-of-the-Art circa 2009

- This is the GMM + HMM approach to ASR.
- Efficient algorithms exist for training the HMM models from training utterances. (forward-backward algorithm)
- Efficient algorithms exist for finding the most likely word sequence given trained HMMs by efficiently computing the likelihoods (Viterbi algorithm).
- Models of this form had brought great improvements to ASR in the 90's
- But in the naughties performance had stagnated. (even when including state-of-the-art techniques such as discriminative training, speaker adaptive training, triphone modelling etc.)

# Progress had stalled

## Progress of spontaneous speech recognition

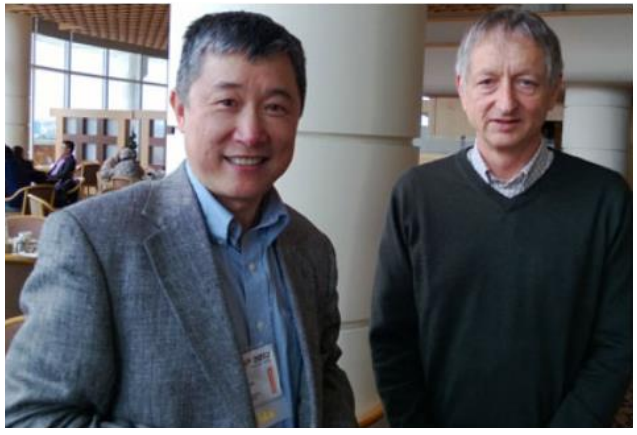



Best word error rates on the *Switchboard* dataset.

2400 two-sided phone conversations among 543 speakers (302 male, 241 female) from all areas of US.

# Pivotal Academic-Industrial Collaboration

- Geoff Hinton collaboration with MSR, Redmond 2009-2010.



Deng (MSR, Redmond) & Hinton (University of Toronto)

- Mutually beneficial academic-industrial collaboration
  - *Automatic Speech Recognition* (ASR) industry looking for new approaches as progress had stalled.
  - Hinton had developed deep learning tools (Deep belief networks 2006) to train deep networks looking for applications and data.

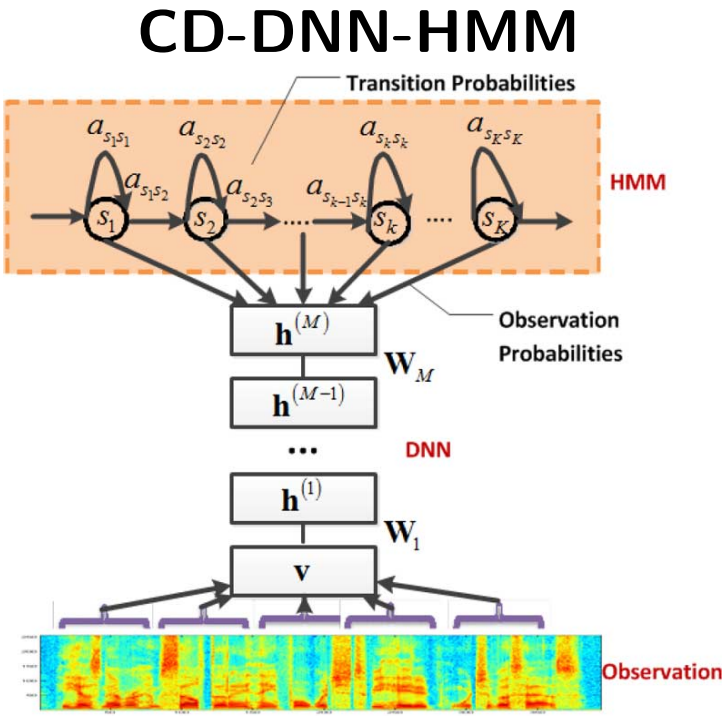
# Timing was great

- **Computing power and data available**
  - Advent of GPU computing. (Nvidia CUDA library released 2007/08)
  - Large labelled training sets data in speech were already available.

- **Algorithms/Approaches existed to train deep networks**
  - Layer-wise training of DBNs
  - Added supervised training, classic back-prop, to Hinton's deep generative models to train Deep Neural Networks.

# Exciting results

Error rate on *Switchboard* dataset down to 18.5% from 27.4%




- Replace GMM of GMM-HMM with a deep neural network (DNN)
- For input to DNN use longer MFCC/filter-back windows with no transformation.

Dahl, Yu, Deng, and Acero, *Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition*, IEEE Trans. ASLP, Jan. 2012 (also ICASSP 2011)

Seide, Li and Yu, *Conversational Speech Transcription Using Context-Dependent Deep Neural Network*,2011

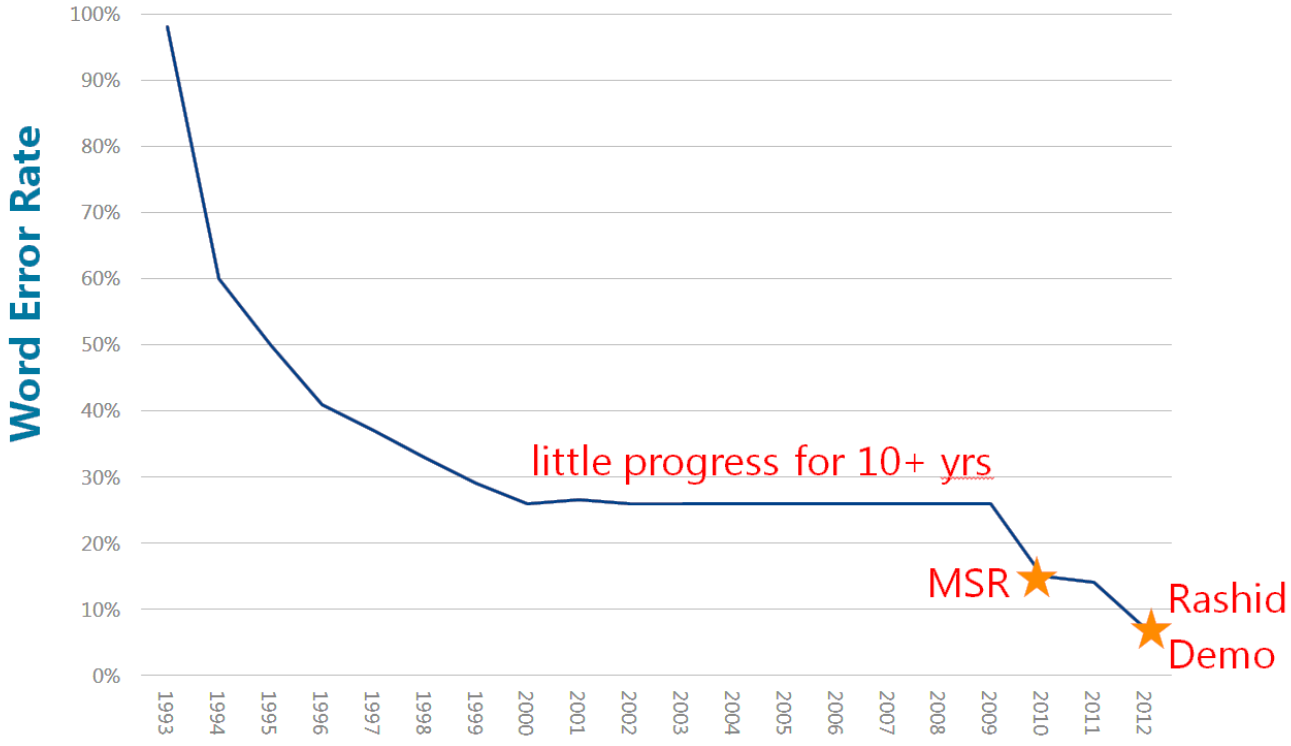
# Deep Learning Technical Revolution

- First resurgence
  - A. Mohamed, G. Dahl and G. Hinton *"Deep belief networks for phone recognition,"* In NIPS Workshop on Deep Learning for Speech Recognition and Related Applications. 2009 — 2009
- DNNs for Large Scale Tasks
  - F. Seide, G. Li, and D. Yu, "*Conversational Speech Transcription Using Context-Dependent Deep Neural Networks*," in Proc. Interspeech 2011. — 2011
- CNNs for Large Scale Tasks
  - T. N. Sainath, A. Mohamed, B. Kingsbury and B. Ramabhadran, "Deep Convolutional Neural Networks for LVCSR," in Proc. ICASSP, 2013. — 2013
- LSTMs for Large Scale Tasks
  - H. Sak, A. Senior and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in Proc. Interspeech, 2014. — 2014

[Slide: Tara Sainath, Google, Advancements in Deep Learning, SLT Keynote, Dec 2014.]

# Deep Learning has big impact

## Progress of spontaneous speech recognition




After no improvement for 10+ years by the research community . . . MSR reduced error from ~27% to $< 13\%$ (and under $< 7\%$ for Rick Rashid's demo in 2012)!

Microsoft Research

# Impact of deep learning in speech technology



Skype to get 'real-time' translator





Next Success Story of Deep Learning:
**Image Recognition & Computer Vision**

# ImageNet Large Scale Visual Recognition Challenge

Steel drum

| **Output:** |
| --- |
| Scale |
| T-shirt |
| Steel drum |
| Drumstick |
| Mud turtle |

✔

| **Output:** |
| --- |
| Scale |
| T-shirt |
| Giant panda |
| Drumstick |
| Mud turtle |

✗

$$\text{Error} = \frac{1}{100,000} \sum_{100,000 \text{ images}} 1(\text{incorrect on image } i)$$

**Source**: Detecting avocados to zucchinis: what have we done, and where are we going? O. Russakovsky et al., ICCV 2013

# How well would a human perform on ImageNet?

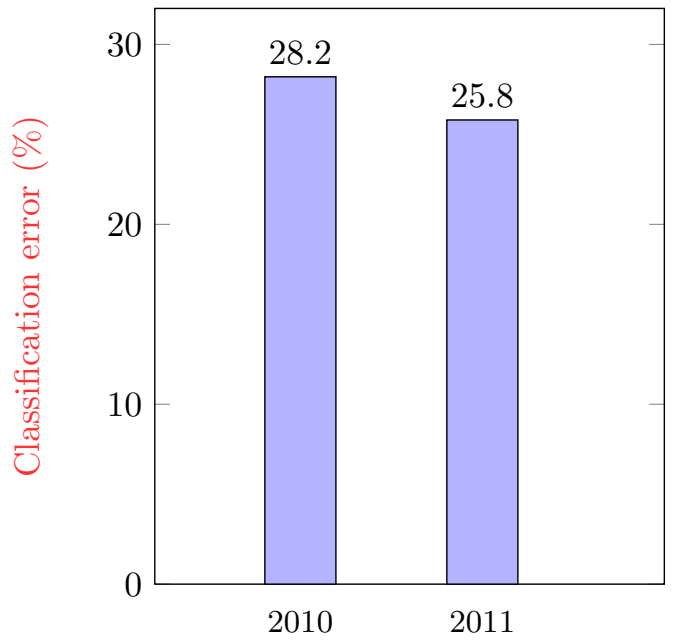
- Andrej Karpathy, Stanford, set himself this challenge.
- Replicated the 1000 way classification problem for a human.
  - Person shown image on the left of figure.
  - On the right shown 13 examples from each of the 1000 classes.
  - Must pick 5 of these classes as the potential ground truth label.

# How well would a human perform on ImageNet?

- Efforts and results reported on his blog post.
- Estimated his own accuracy on ImageNet as 5.1%. (After some training period.)
- Later conjectured (Feb 2015) a dedicated and motivated human classifier capable of error rate in the range of 2%–3%.

# State-of-the-art performance in 2011

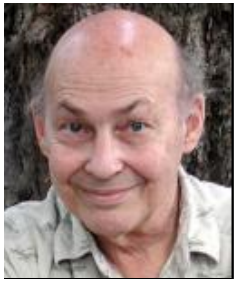
- ImageNet Large Scale Visual Recognition Challenge
- 1000 object classes, 1.4 million labelled training images




Performance of winning entry in ILSVRC competitions (2010-11) prior to deep learning.

# A whirlwind review of computer vision

# The Birth of Computer Vision



**Marvin Minsky**

**Gerald Sussman**

**1966**

Marvin Minsky (MIT) asked his undergraduate student Gerald Jay Sussman to

> *"spend the summer linking a camera to a computer and getting the computer to describe what it saw."*

# The Birth of Computer Vision

**Marvin Minsky**

**Gerald Sussman**

## 1966

Marvin Minsky (MIT) asked his undergraduate student Gerald Jay Sussman to

> *"spend the summer linking a camera to a computer and getting the computer to describe what it saw."*

Now know the problem was much more difficult.... [1]

[1] "*You'll notice that Sussman never worked in vision again!*" – Berthold Horn

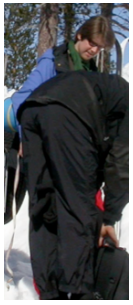
# Why is it so hard? Consider object recognition

**Challenges:**

**View Point Variation**

**Illumination Variation**

**Occlusion**



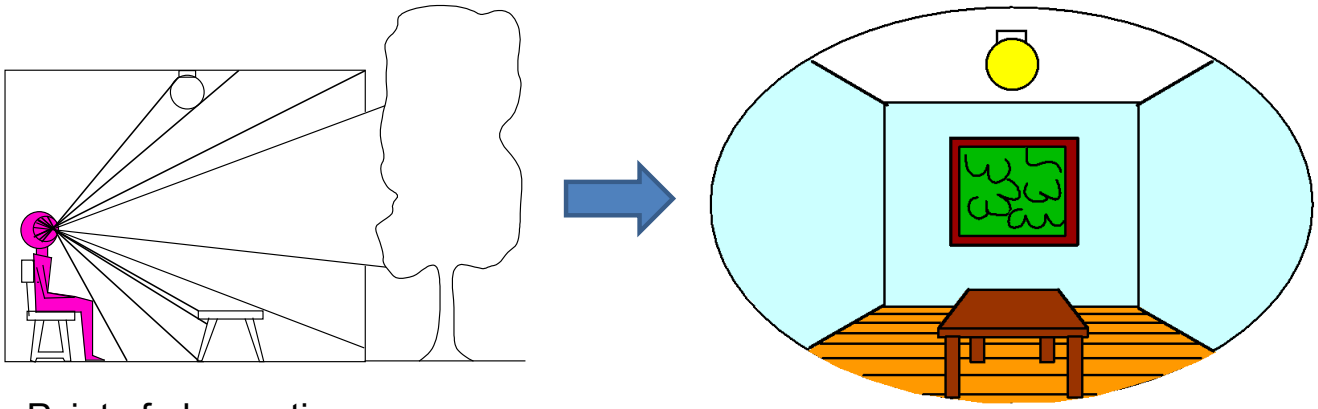
**Deformation**

**Intra-class variation**

and haven't even mentioned clutter or NLP.

**Common high-level road map for object recognition**:

*3D world* *2D image*



Point of observation

"Exploit the physics and geometry of imaging."

- **Training**: Build 3D models of objects from 2D images of them.
- **Test time**:
  - Estimate object's *pose* in the image relative to camera's position.
  - Synthesize how the model would appear in the image.
  - Compare the synthetic image to the real image.

# Successes driven by this line of research

# Mosaic Generation

(a) Matier data set (7 images)

# However progress on problem of object detection stalled

- Unclear how to mathematically model object categories.
- Unclear how best to spot and distinguish between instances of these models in images.

# Shift to learning based methods in the naughties

# What we mean by learning based methods

Solve problem by referencing to training data

> *Have a face image if it looks like an image which I know is a face.*

# What we mean by learning based methods

Solve problem by referencing to training data

> *Have a face image if it looks like an image which I know is a face.*

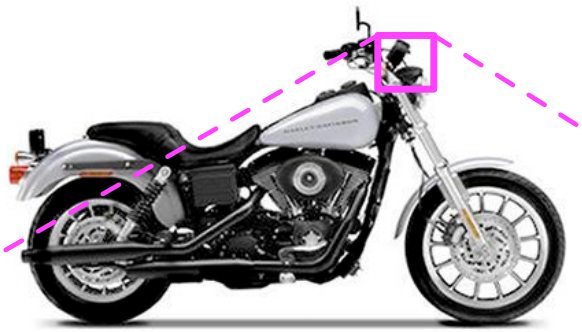
Trend fueled by the

- rapid growth of computational power,
- rapid growth of memory and
- the abundance of digital images and video and the web.

# Recognition in Computer Vision 101

# Image is an array of numbers

You see this



| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 34 | 45 | 53 | 55 | 69 | 79 | 91 | 95 | 105 | 197 | 254 | 250 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 0 | 11 | 20 | 39 | 59 | 58 | 62 | 73 | 67 | 92 | 213 | 255 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 5 | 5 | 0 | 11 | 30 | 16 | 39 | 87 | 67 | 27 | 167 | 255 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 0 | 0 | 10 | 12 | 8 | 5 | 73 | 172 | 172 | 140 | 204 | 255 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 5 | 0 | 17 | 0 | 0 | 20 | 123 | 237 | 249 | 255 | 246 | 250 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 0 | 16 | 9 | 0 | 0 | 48 | 200 | 255 | 242 | 255 | 255 | 255 | 255 | 255 | 254 | 252 | 251 | 252 | 253 | 254 |
| 7 | 0 | 0 | 5 | 23 | 175 | 234 | 250 | 243 | 250 | 253 | 254 | 251 | 251 | 252 | 252 | 253 | 253 | 254 | 254 |
| 0 | 0 | 17 | 0 | 17 | 198 | 255 | 248 | 250 | 246 | 255 | 245 | 254 | 255 | 255 | 255 | 255 | 253 | 252 | 250 |
| 0 | 16 | 2 | 14 | 69 | 125 | 247 | 255 | 255 | 247 | 255 | 249 | 253 | 253 | 253 | 254 | 253 | 253 | 252 | 251 |
| 26 | 15 | 1 | 109 | 181 | 102 | 148 | 235 | 254 | 240 | 249 | 252 | 250 | 250 | 250 | 250 | 251 | 252 | 254 | 255 |
| 0 | 0 | 44 | 203 | 249 | 169 | 69 | 208 | 255 | 255 | 248 | 255 | 255 | 255 | 255 | 255 | 255 | 253 | 251 | 250 |
| 4 | 47 | 156 | 232 | 255 | 245 | 115 | 166 | 244 | 253 | 249 | 245 | 244 | 247 | 252 | 255 | 255 | 254 | 251 | 249 |
| 114 | 193 | 251 | 253 | 247 | 255 | 191 | 88 | 153 | 185 | 207 | 182 | 200 | 209 | 224 | 240 | 251 | 255 | 255 | 255 |
| 193 | 255 | 255 | 213 | 147 | 131 | 97 | 63 | 59 | 77 | 86 | 81 | 88 | 110 | 123 | 112 | 156 | 199 | 250 | 245 |
| 228 | 178 | 151 | 113 | 9 | 4 | 17 | 40 | 43 | 32 | 42 | 68 | 65 | 53 | 36 | 74 | 70 | 75 | 121 | 215 |
| 171 | 52 | 33 | 0 | 13 | 0 | 0 | 31 | 44 | 29 | 32 | 55 | 61 | 72 | 71 | 107 | 91 | 55 | 62 | 165 |
| 47 | 0 | 17 | 11 | 40 | 28 | 22 | 33 | 52 | 68 | 76 | 80 | 78 | 101 | 119 | 110 | 124 | 63 | 74 | 170 |
| 21 | 22 | 19 | 30 | 26 | 45 | 59 | 60 | 64 | 77 | 85 | 84 | 93 | 101 | 125 | 120 | 117 | 30 | 56 | 213 |
| 35 | 40 | 27 | 51 | 52 | 51 | 57 | 66 | 66 | 55 | 48 | 49 | 92 | 108 | 108 | 101 | 52 | 0 | 18 | 195 |
| 27 | 19 | 52 | 89 | 56 | 31 | 19 | 34 | 45 | 41 | 40 | 47 | 67 | 66 | 39 | 15 | 18 | 45 | 51 | 159 |

but a computer sees this

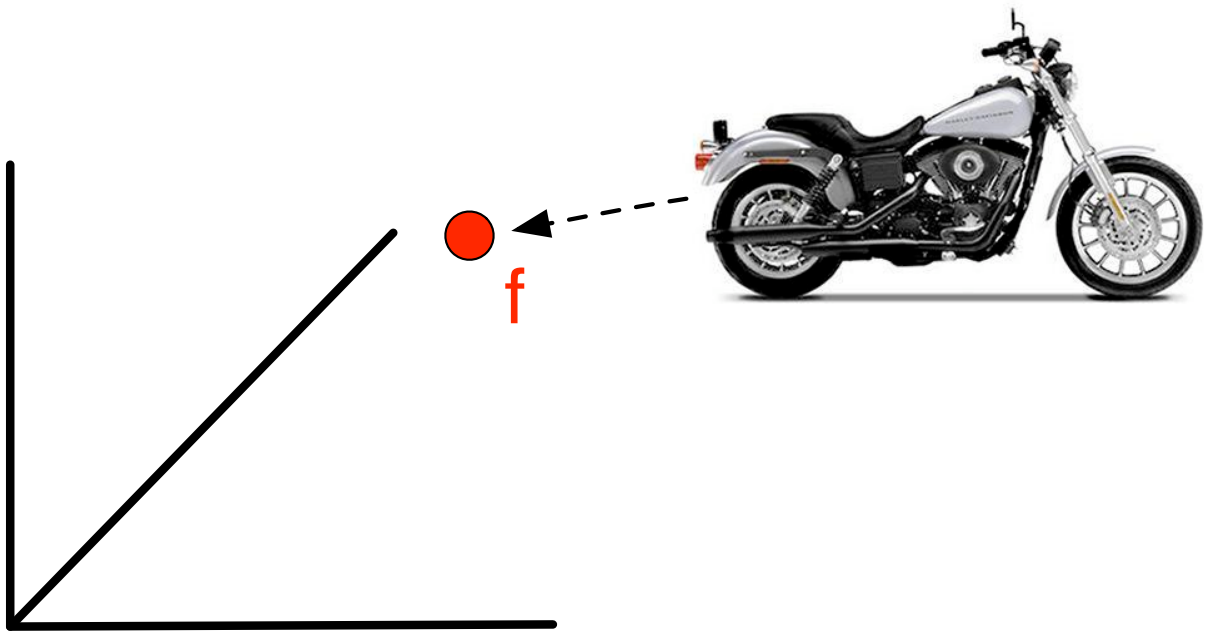
# Feature Extraction

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 34 | 45 | 53 | 55 | 69 | 79 | 91 | 95 | 105 | 197 | 254 | 250 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 0 | 11 | 20 | 39 | 59 | 58 | 62 | 73 | 67 | 92 | 213 | 255 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 5 | 5 | 0 | 11 | 30 | 16 | 39 | 87 | 67 | 27 | 167 | 255 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 0 | 0 | 10 | 12 | 8 | 5 | 73 | 172 | 172 | 140 | 204 | 255 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 5 | 0 | 17 | 0 | 0 | 20 | 123 | 237 | 249 | 255 | 246 | 250 | 254 | 254 | 254 | 254 | 254 | 254 | 254 | 254 |
| 0 | 16 | 9 | 0 | 0 | 48 | 200 | 255 | 242 | 255 | 255 | 255 | 255 | 255 | 254 | 252 | 251 | 252 | 253 | 254 |
| 7 | 0 | 0 | 5 | 23 | 175 | 234 | 250 | 243 | 250 | 253 | 254 | 251 | 251 | 252 | 252 | 253 | 253 | 254 | 254 |
| 0 | 0 | 17 | 0 | 17 | 198 | 255 | 248 | 250 | 246 | 255 | 245 | 254 | 255 | 255 | 255 | 255 | 253 | 252 | 250 |
| 0 | 16 | 2 | 14 | 69 | 125 | 247 | 255 | 255 | 247 | 255 | 249 | 253 | 253 | 253 | 254 | 253 | 253 | 252 | 251 |
| 26 | 15 | 1 | 109 | 181 | 102 | 148 | 235 | 254 | 240 | 249 | 252 | 250 | 250 | 250 | 250 | 251 | 252 | 254 | 255 |
| 0 | 0 | 44 | 203 | 249 | 169 | 69 | 208 | 255 | 255 | 248 | 255 | 255 | 255 | 255 | 255 | 255 | 253 | 251 | 250 |
| 4 | 47 | 156 | 232 | 255 | 245 | 115 | 166 | 244 | 253 | 249 | 245 | 244 | 247 | 252 | 255 | 255 | 254 | 251 | 249 |
| 114 | 193 | 251 | 253 | 247 | 255 | 191 | 88 | 153 | 185 | 207 | 182 | 200 | 209 | 224 | 240 | 251 | 255 | 255 | 255 |
| 193 | 255 | 255 | 213 | 147 | 131 | 97 | 63 | 59 | 77 | 86 | 81 | 88 | 110 | 123 | 112 | 156 | 199 | 250 | 245 |
| 228 | 178 | 151 | 113 | 9 | 4 | 17 | 40 | 43 | 32 | 42 | 68 | 65 | 53 | 36 | 74 | 70 | 75 | 121 | 215 |
| 171 | 52 | 33 | 0 | 13 | 0 | 0 | 31 | 44 | 29 | 32 | 55 | 61 | 72 | 71 | 107 | 91 | 55 | 62 | 165 |
| 47 | 0 | 17 | 11 | 40 | 28 | 22 | 33 | 52 | 68 | 76 | 80 | 78 | 101 | 119 | 110 | 124 | 63 | 74 | 170 |
| 21 | 22 | 19 | 30 | 26 | 45 | 59 | 60 | 64 | 77 | 85 | 84 | 93 | 101 | 125 | 120 | 117 | 30 | 56 | 213 |
| 35 | 40 | 27 | 51 | 52 | 51 | 57 | 66 | 66 | 55 | 48 | 49 | 92 | 108 | 108 | 101 | 52 | 0 | 18 | 195 |
| 27 | 19 | 52 | 89 | 56 | 31 | 19 | 34 | 45 | 41 | 40 | 47 | 67 | 66 | 39 | 15 | 18 | 45 | 51 | 159 |

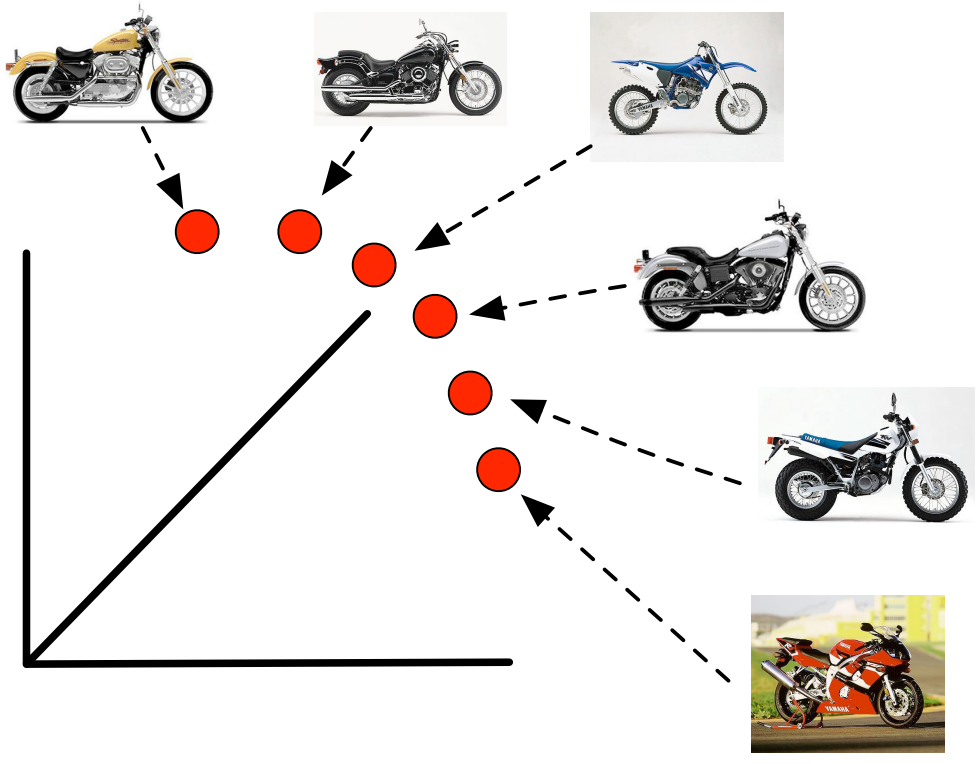
$$\Rightarrow \mathbf{f} = (f_1, f_2, \cdots)$$

**Convert pixel data to a feature vector.**

# Feature extraction turns image into a point

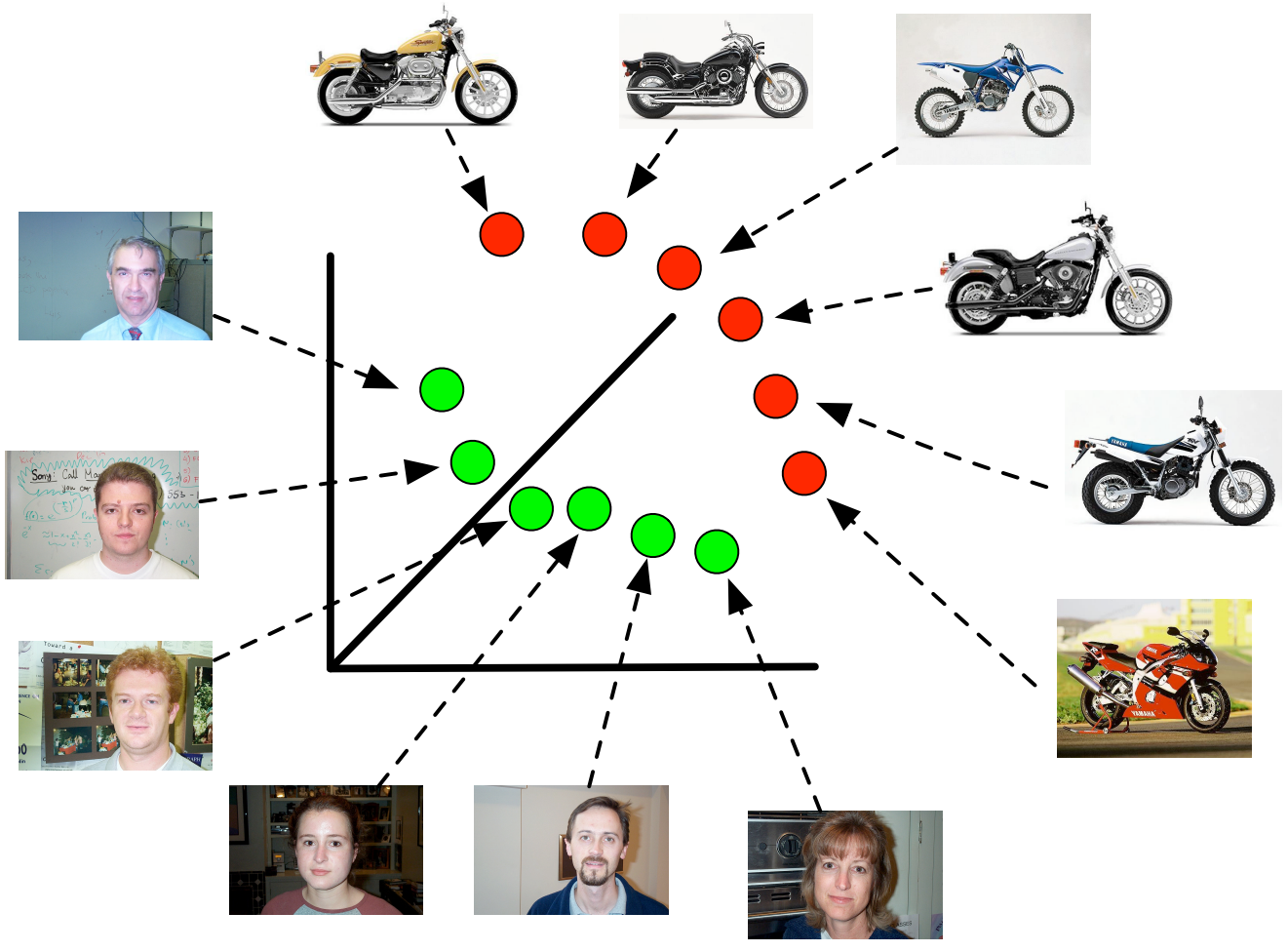


# Learning from examples



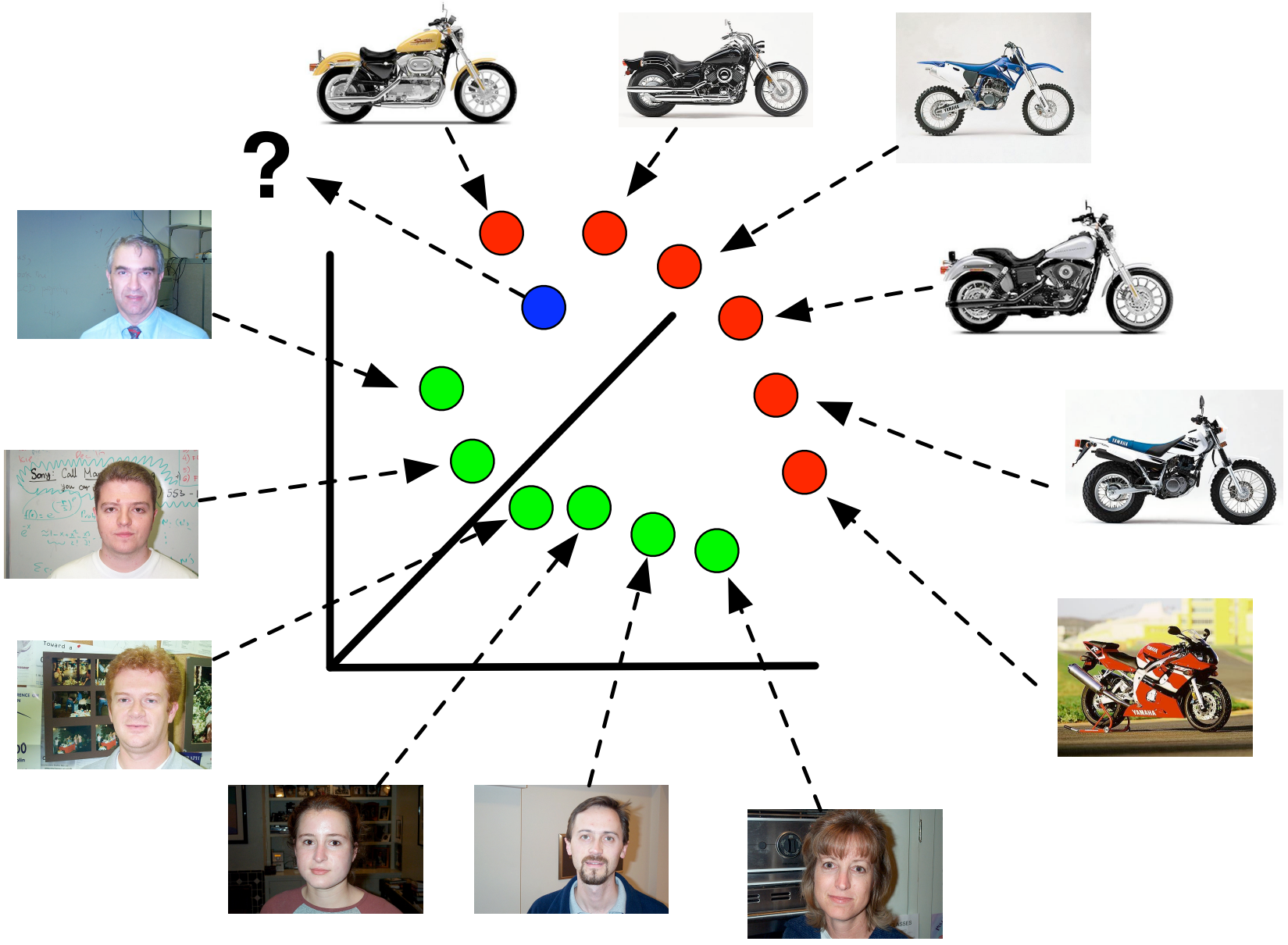
**Many feature vectors from a category**

# Learning from examples

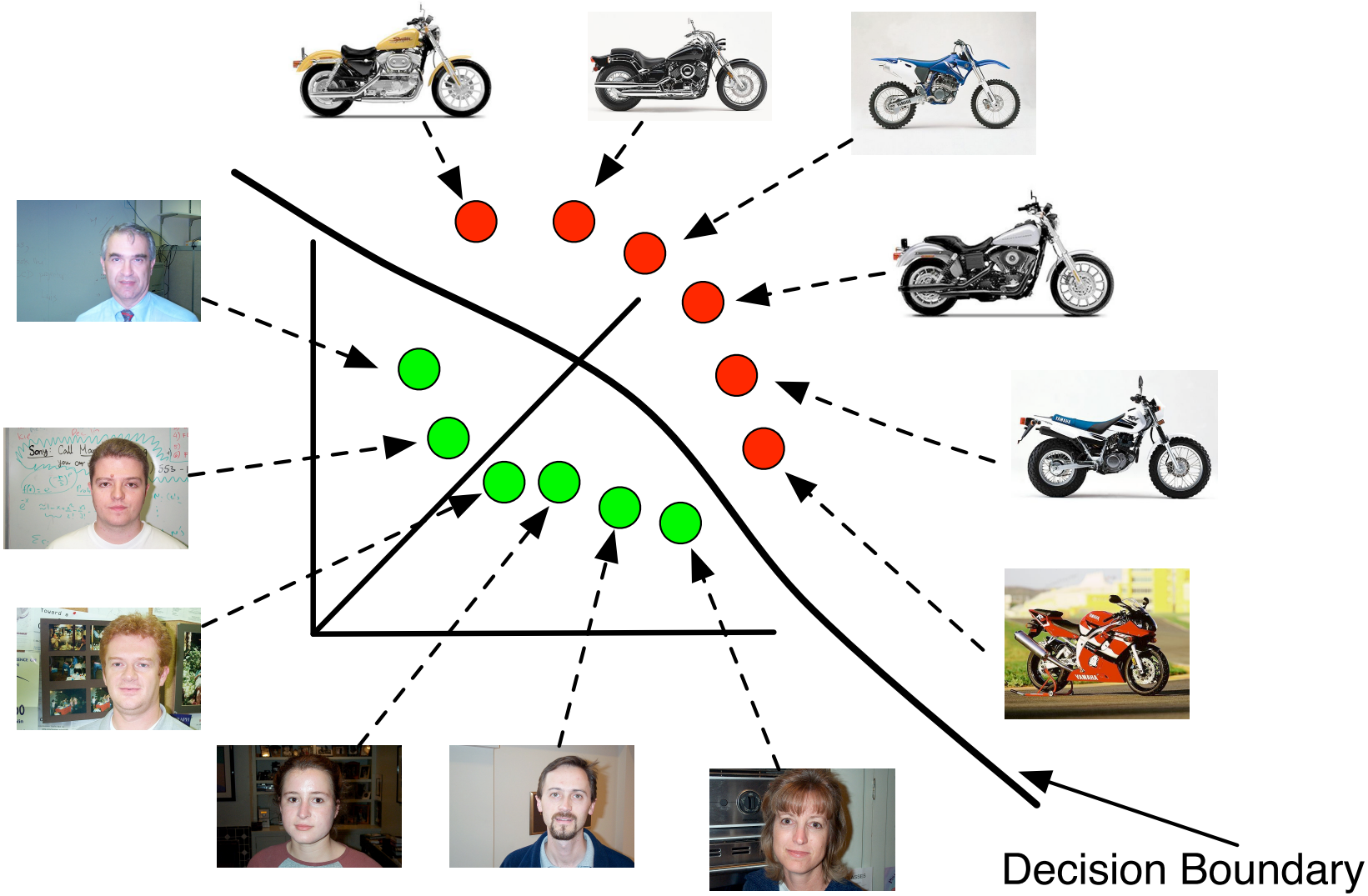


**Want different categories to occupy different volumes.**

# Is it a bike or a face ?



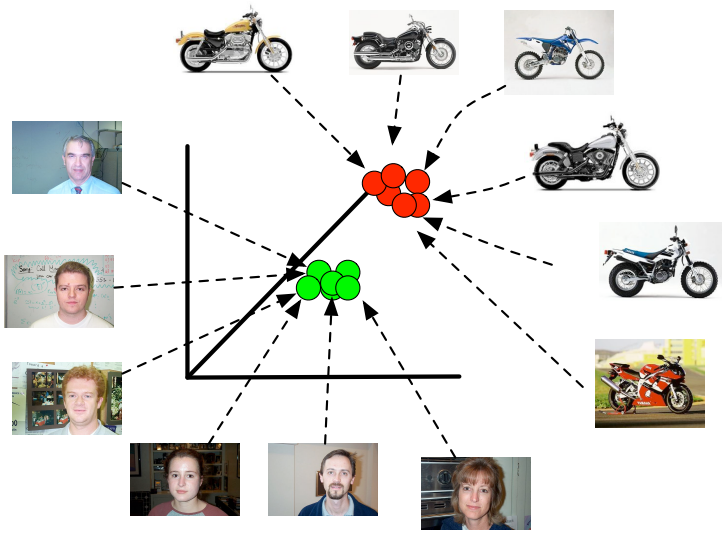
# Construct a decision boundary



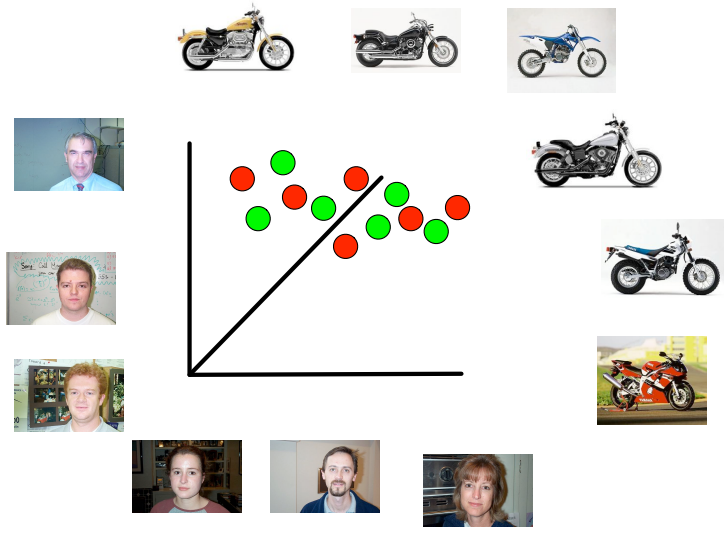
1. **Training Phase**
   - Gather labelled training data.
   - Extract a feature representation for each training example.
   - Construct a decision boundary.
2. **Test Phase**
   - Extract feature representation from the test example.
   - Compare to the learnt decision boundary.

**This is supervised learning.**

# Success depends mainly on quality of feature extraction



**Ideal features**



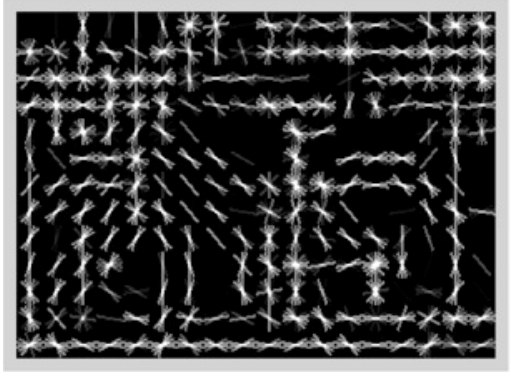
**Far from ideal**

# Naughties focused on hand-crafted features

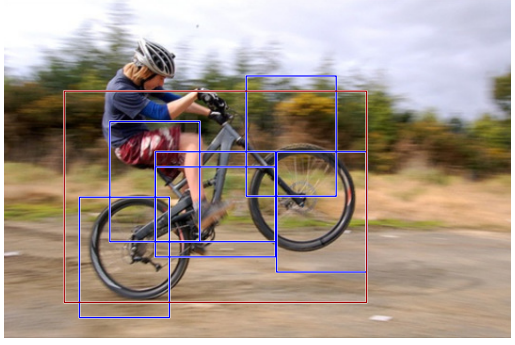
- Engineer/researcher designing and constructing ingenious features.
- Let machine learning do feature selection and refining of features.

**Popular features**

intensity template

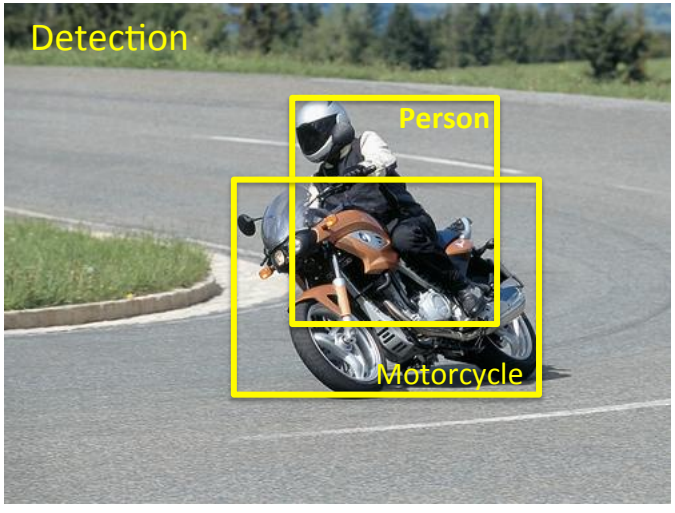


HOG

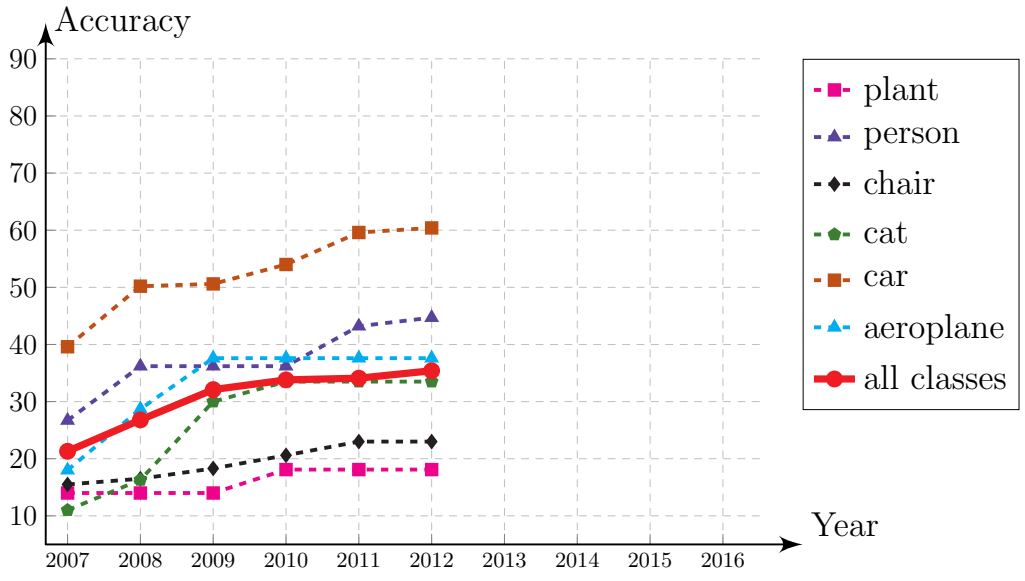


deformable part models

# Progress at first but then stagnation



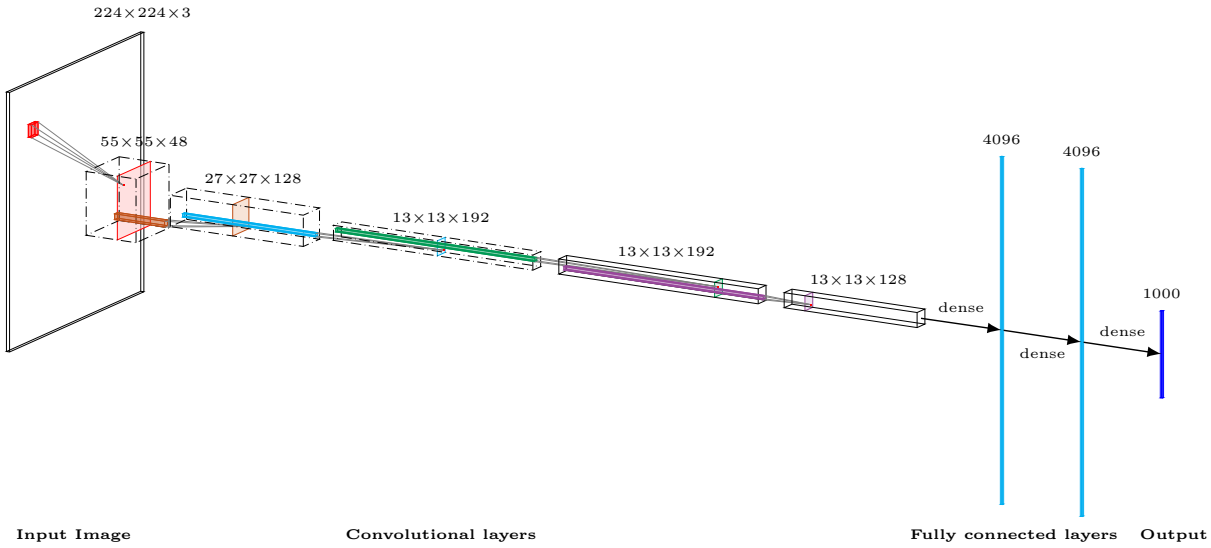

**Task of object detection**



But then ...

# ImageNet 2012: Most exciting CV workshop ever

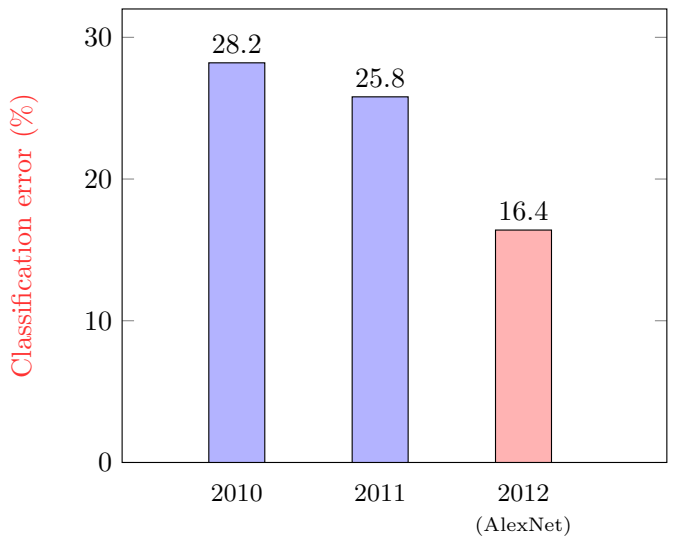
- Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton from University of Toronto present **AlexNet**.




- First modern deep Convolutional Network trained using Backprop to solve a hard computer vision problem.
- Outperforms all competitors by a large margin.

# Impact of AlexNet on state-of-the-art performance

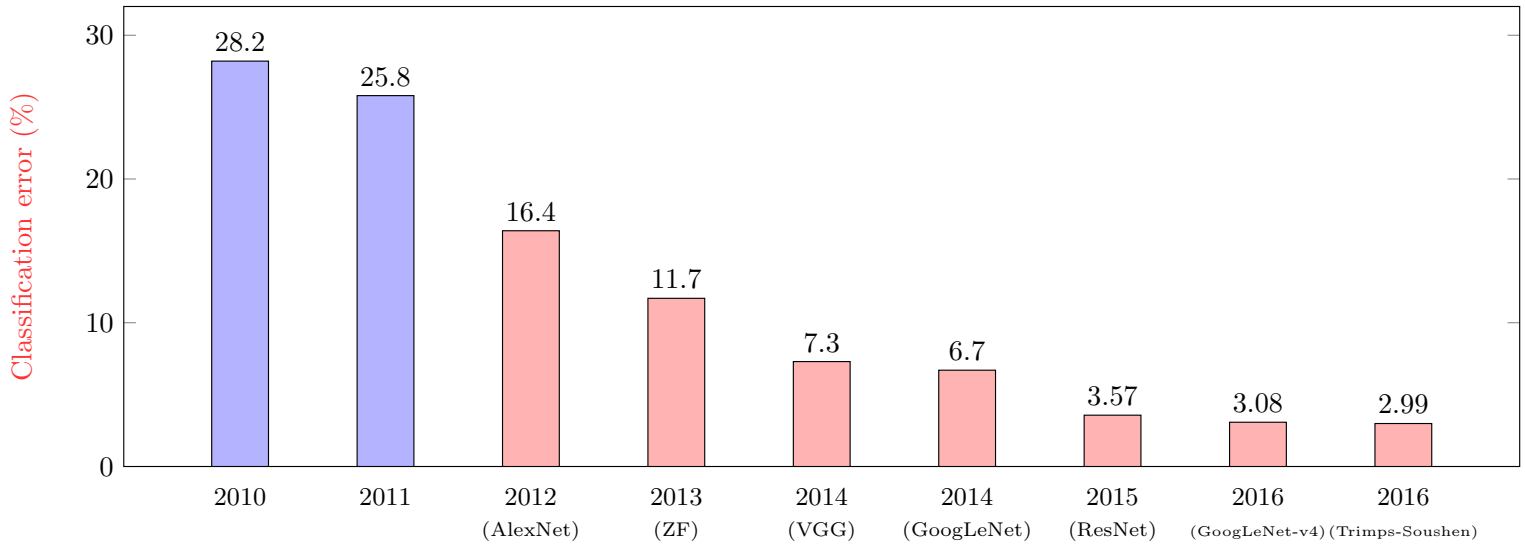
- ImageNet Large Scale Visual Recognition Challenge
- 1000 object classes, 1.4 million labelled training images




Performance of winning entry in ILSVRC competitions (2010-12).

# AlexNet was only the start

- ImageNet Large Scale Visual Recognition Challenge
- 1000 object classes, 1.4 million labelled training images



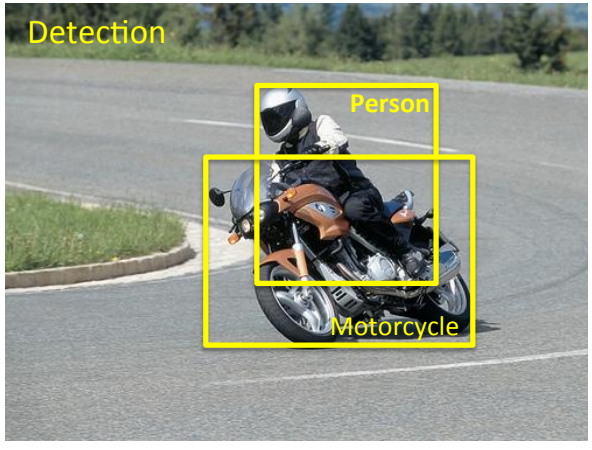

High performing systems on the ILSVRC datasets (2010-16).

Pink indicates a ConvNet based solution.

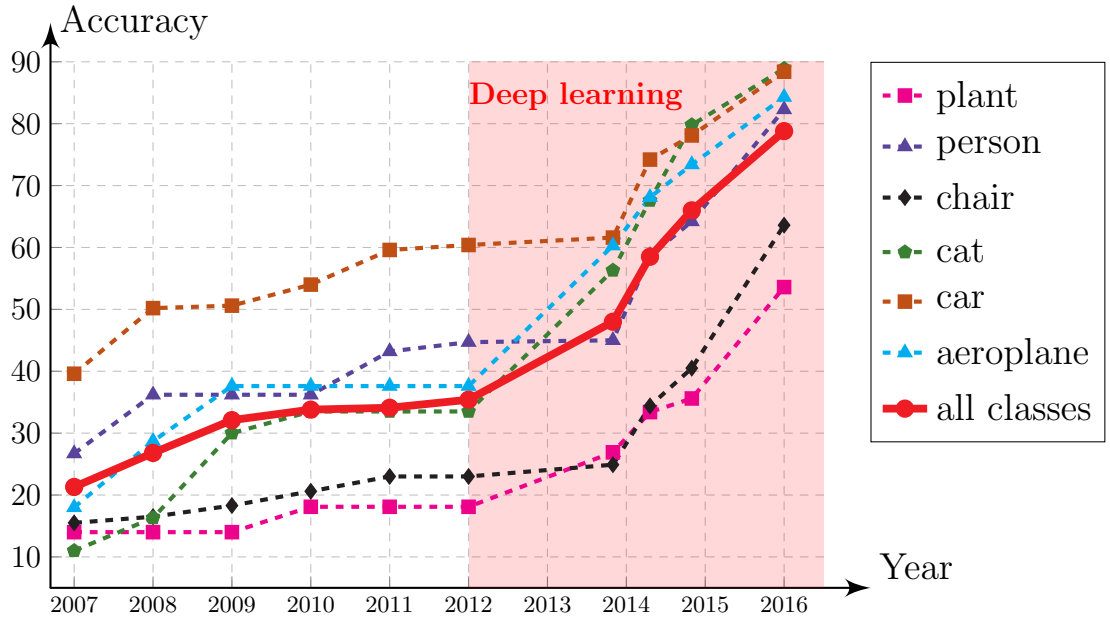
Deep ConvNets $\Longrightarrow$ great for image classification

What about Object Detection?

# Deep ConvNets → much better object detection




**Task of object detection**




Progress on the Pascal VOC 2007 challenge.

# Deep learning allows direct learning of better features

**Key properties of deep learning**

Provides a mechanism to:

- Learn a highly non-linear function.
  (Efficiently encoded in a deep structure.)
- Learn it from data.
- Build feature hierarchies
  - Distributed representations
  - Compositionality
- Perform **end-to-end** learning (no more hand-crafted features)

Exciting application domain of Deep Learning:
**Natural Language Processing**

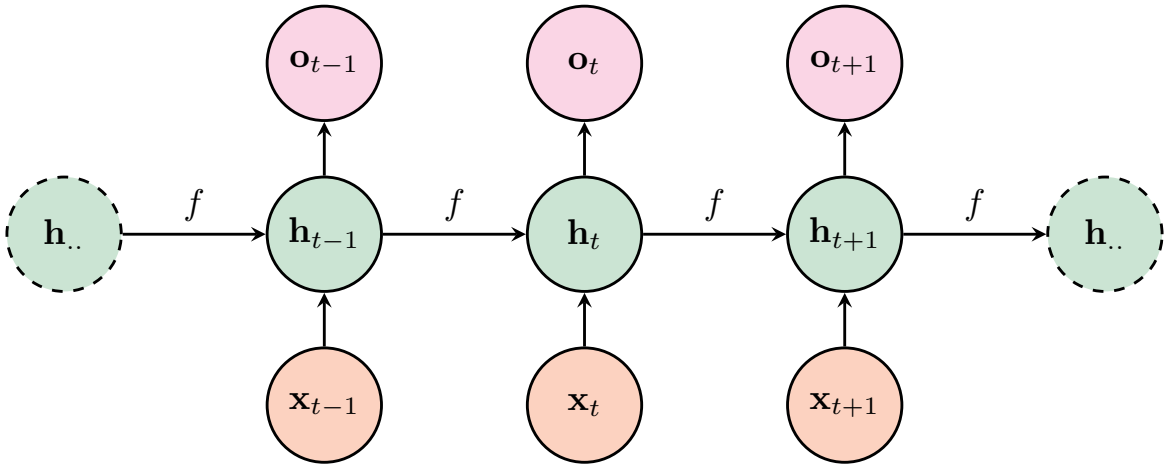
# Language Modelling - Sequence Modelling

- Need a way to computationally model sequences of words - language.

- Need ways of representing language.

**Recurrent Neural Networks (and variants) are the deep learning approach.**

# Recurrent Neural Networks are popular to model sequences



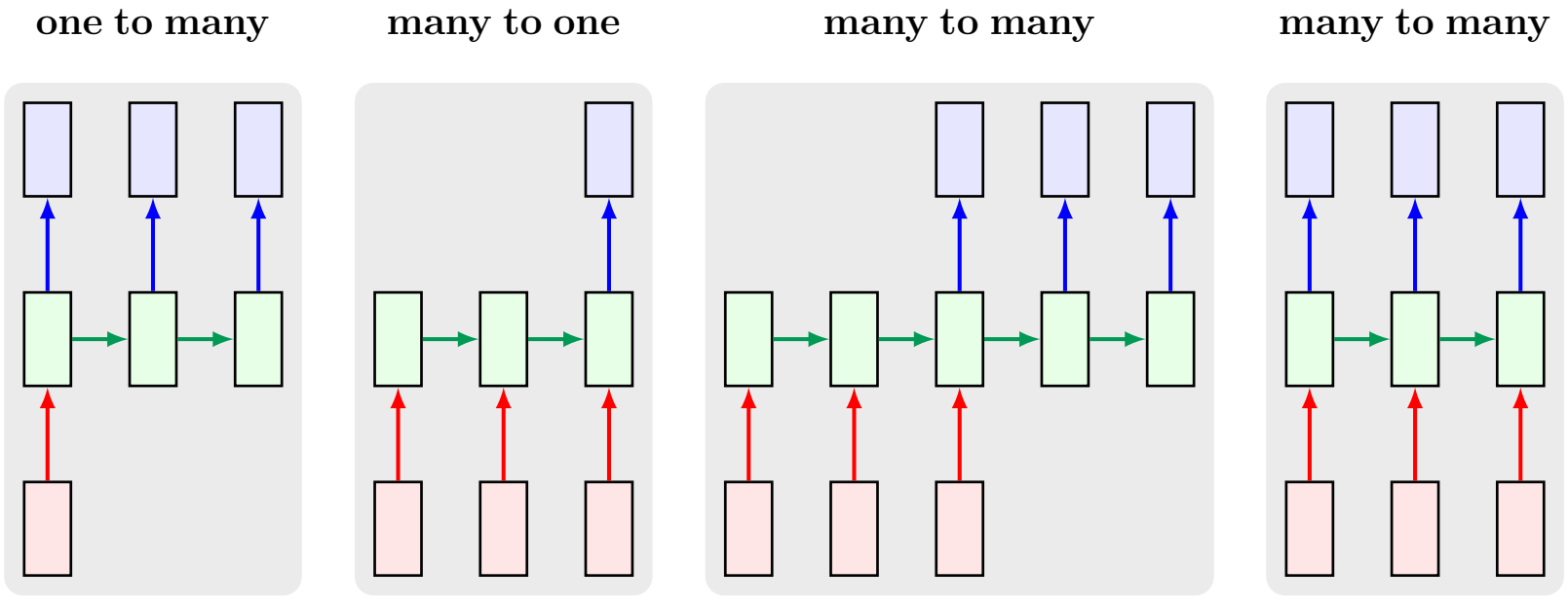

- Have a function $f$ (stays constant over time)

$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t; \boldsymbol{\theta})$$

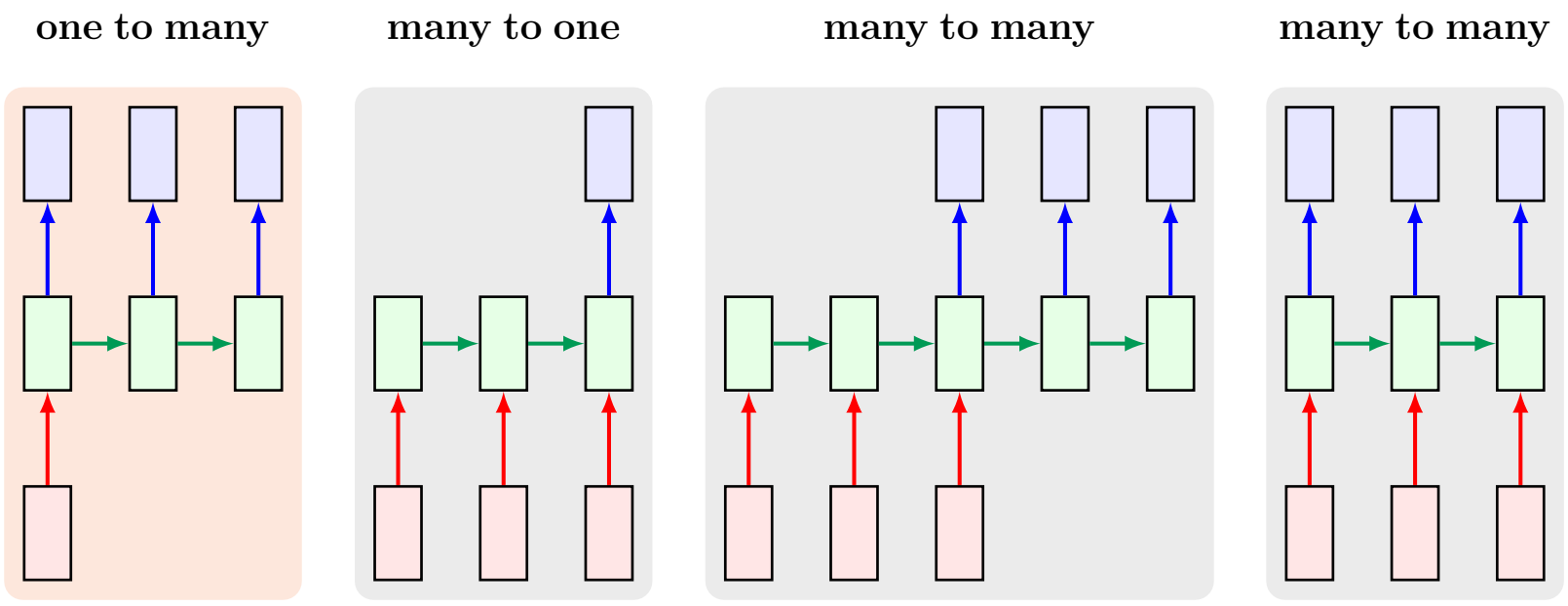
  that defines the hidden state $\mathbf{h}_t$ over time.

- Usually also predict an output vector at each time step.

# Use cases of RNNs



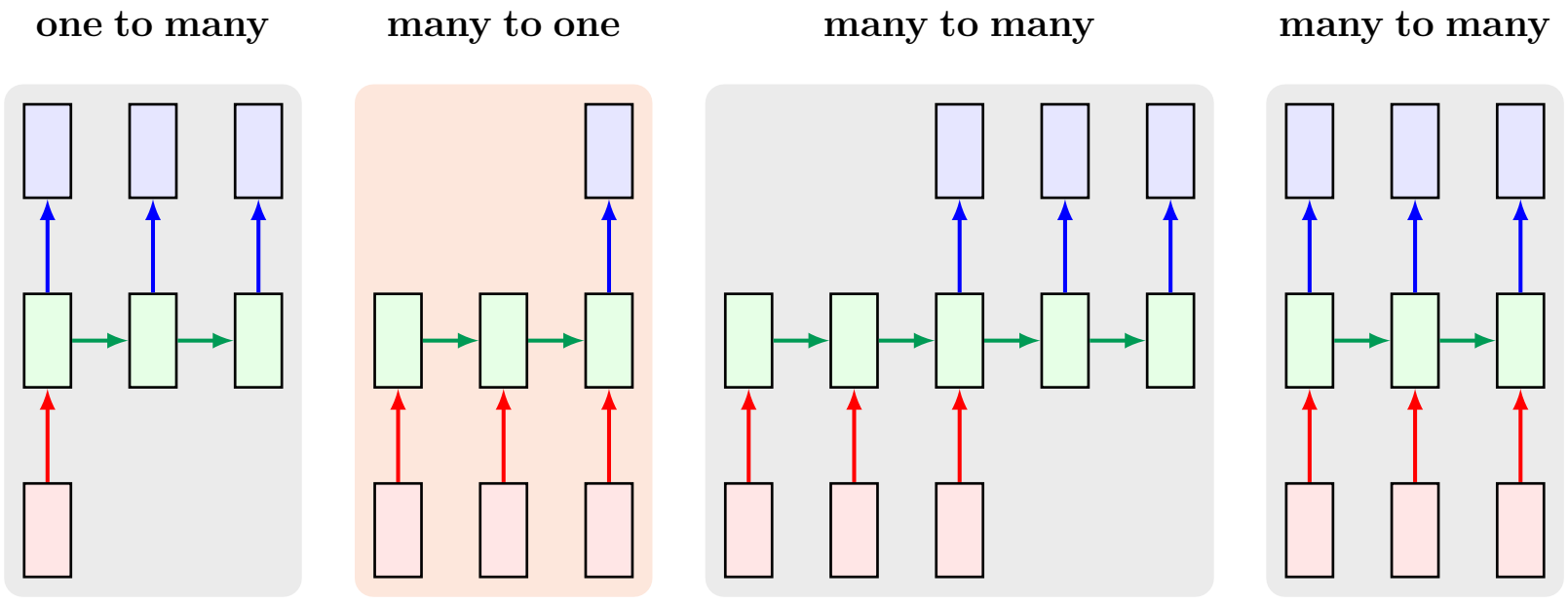
# Use cases of RNNs




**Generation Process:**

Example - **Image captioning**: image $\rightarrow$ sequence of words.

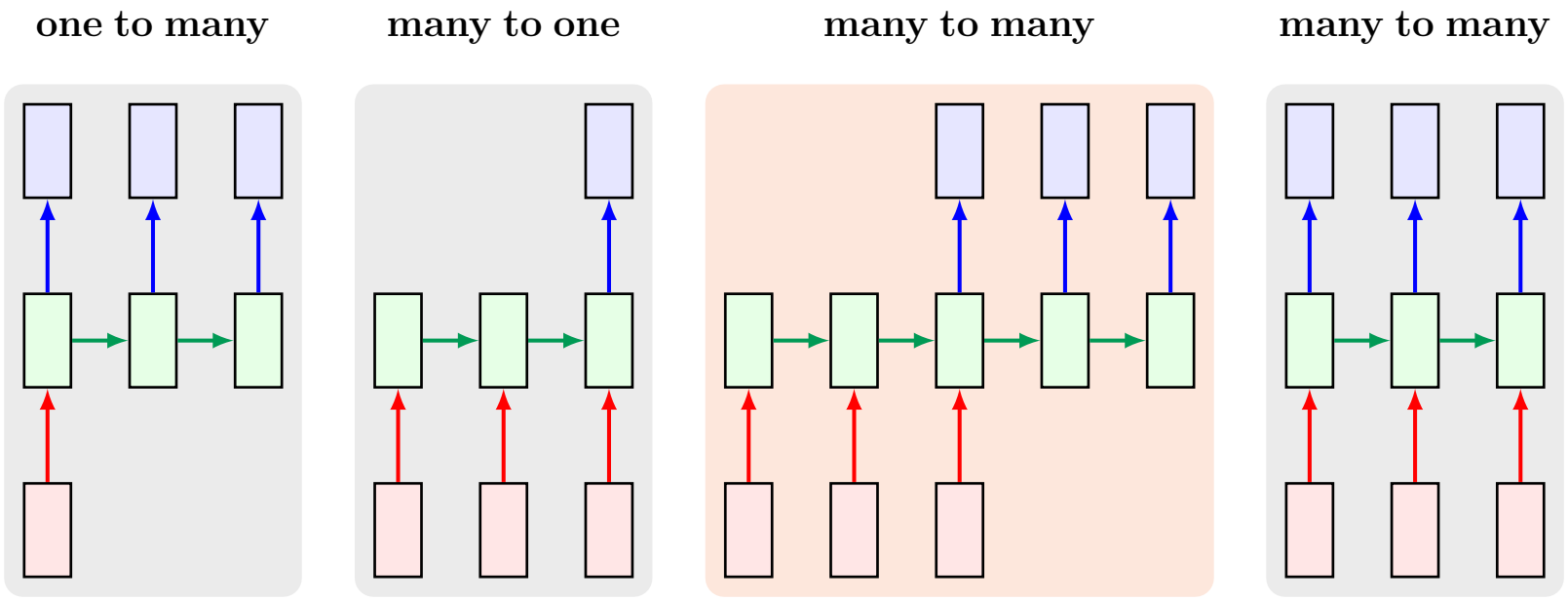
# Use cases of RNNs




**Encoding / Prediction:**

Example - **Sentiment Classification**: seq. of words $\rightarrow$ sentiment

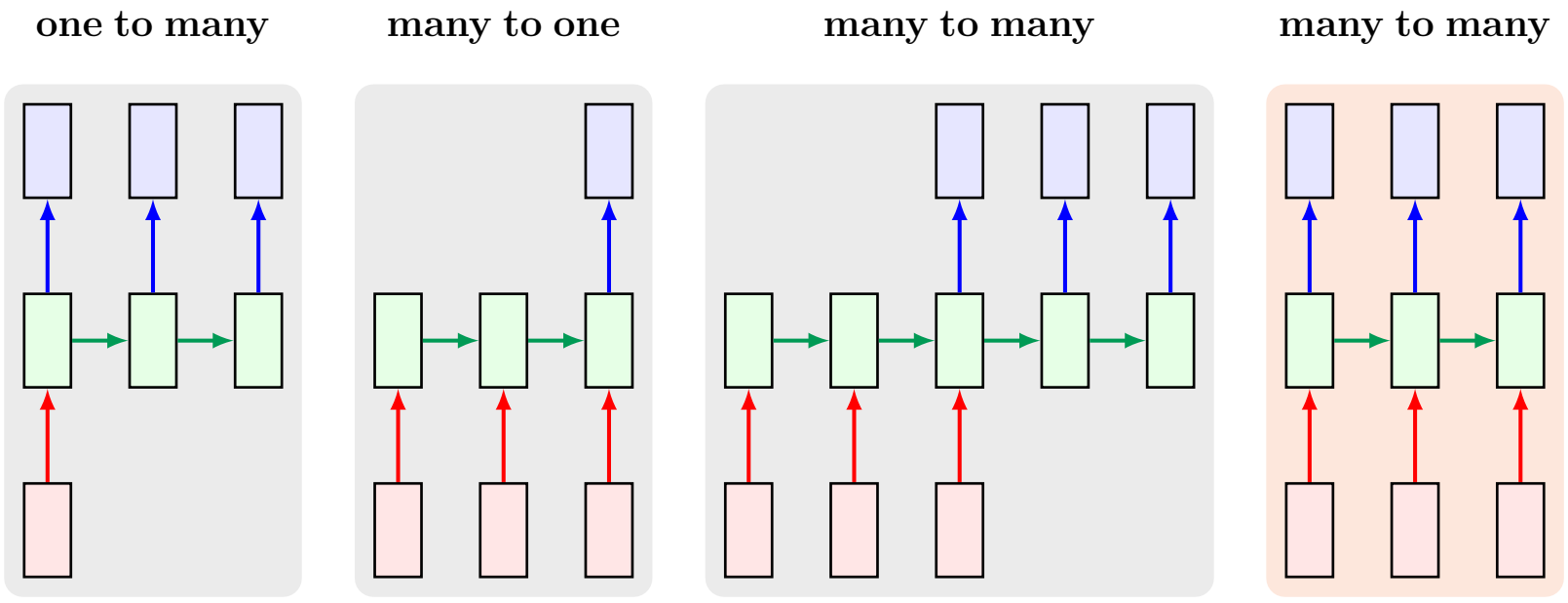
# Use cases of RNNs




**Encode then Decode:**

Example - **Machine Translation**: seq. of words $\rightarrow$ seq. of words

# Use cases of RNNs

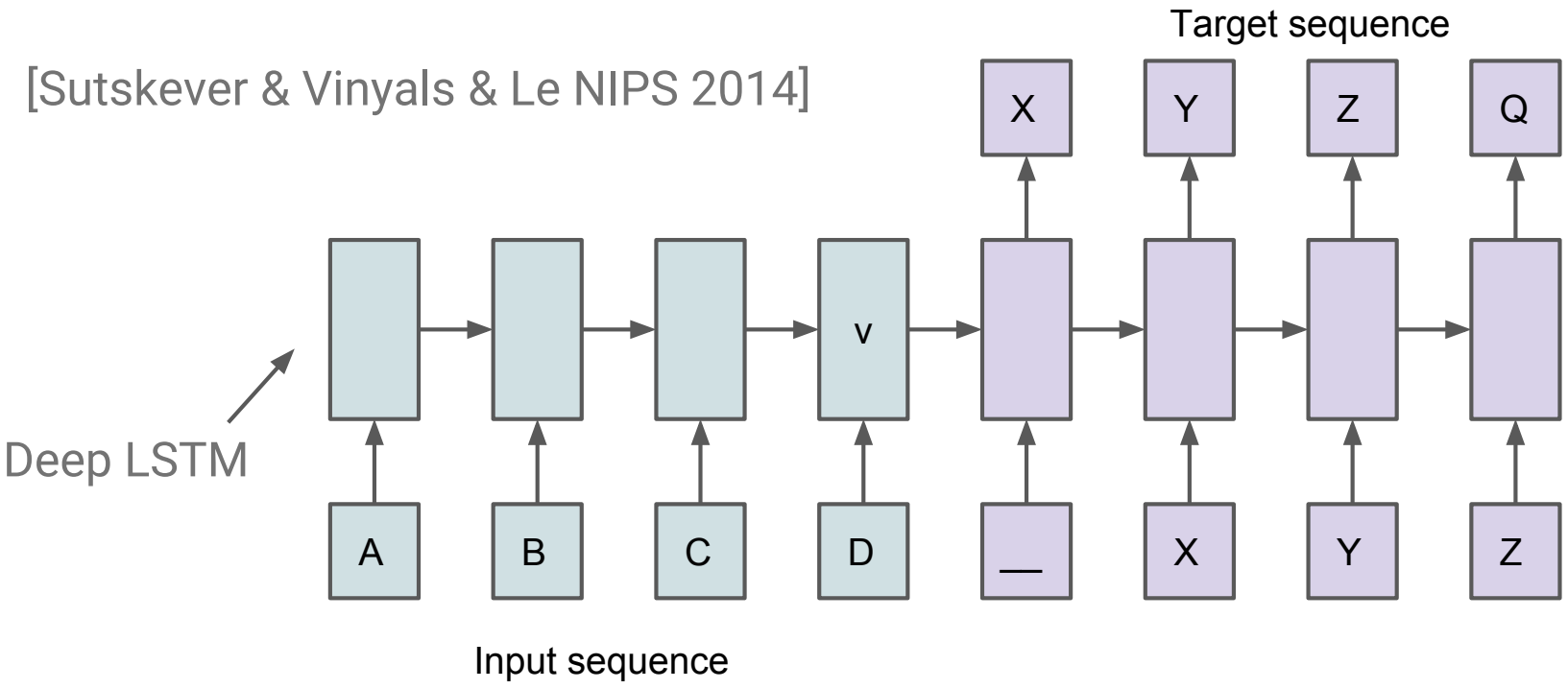



**Continuous Prediction / Encoding**:

Example - **Video classification on frame level**

# Sequence-to-Sequence Model

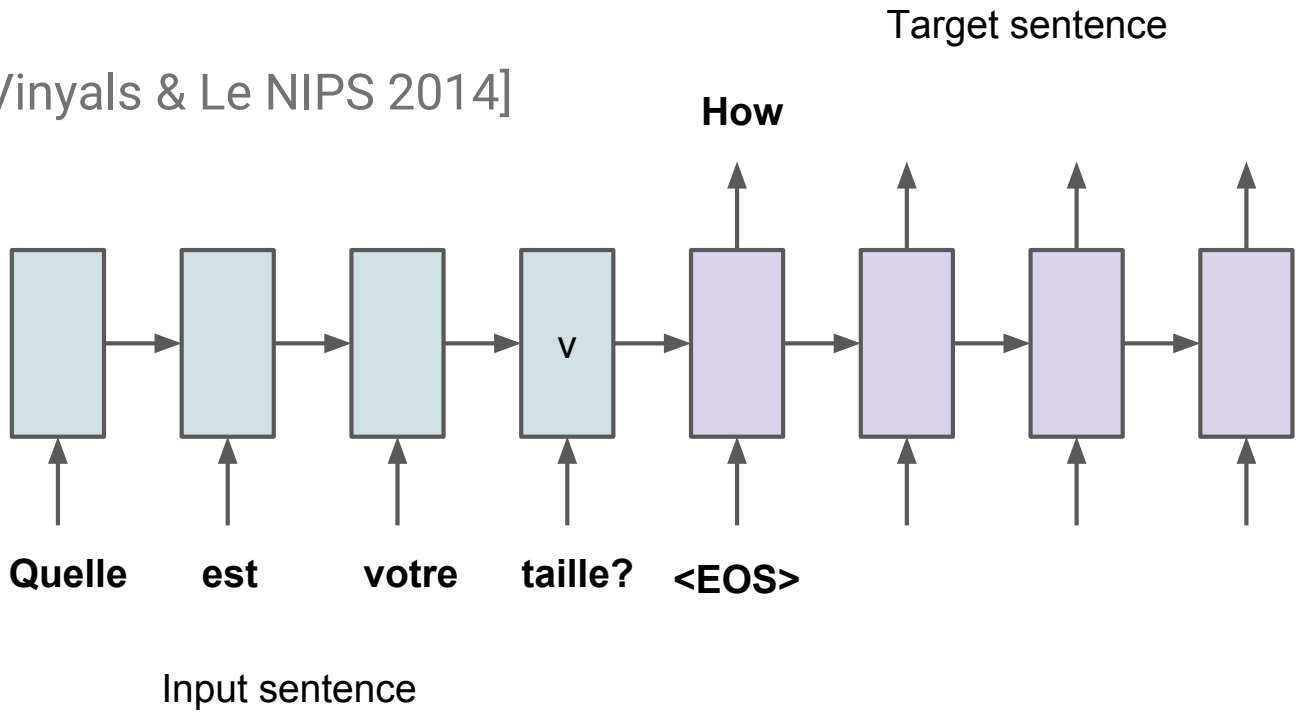
[Sutskever & Vinyals & Le NIPS 2014]




$$P(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$

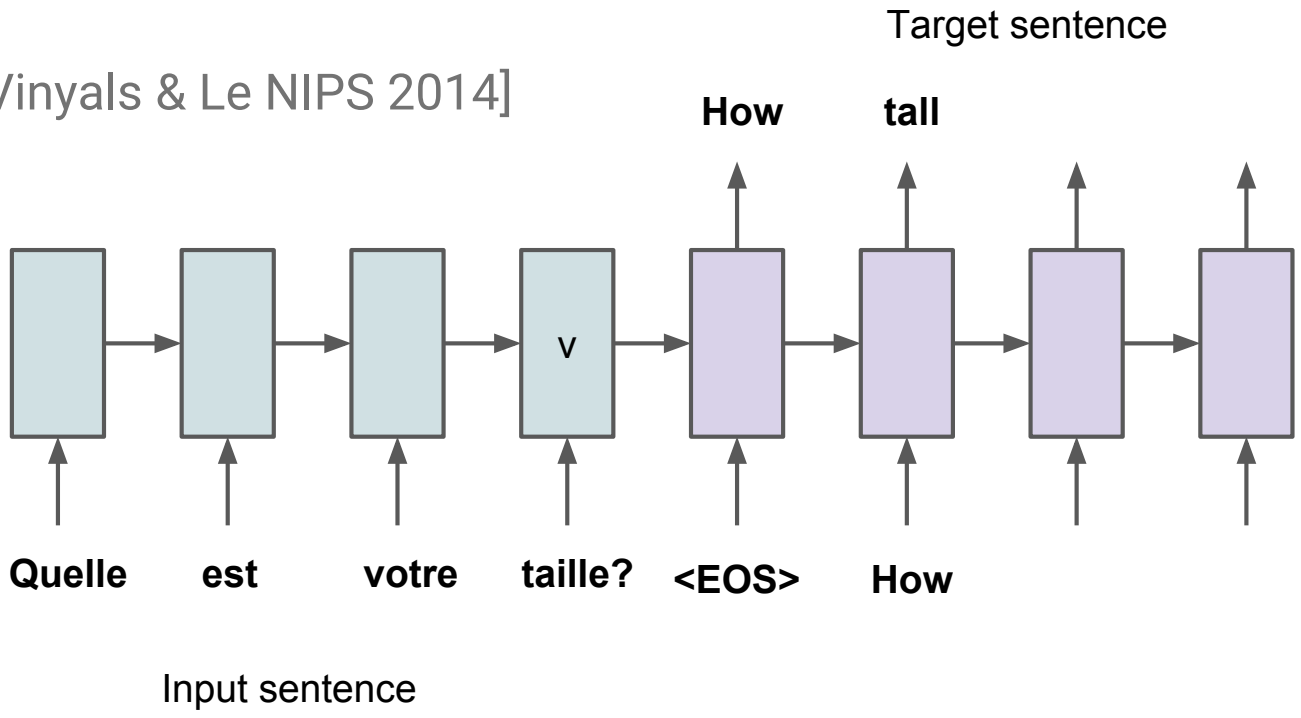
# Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]



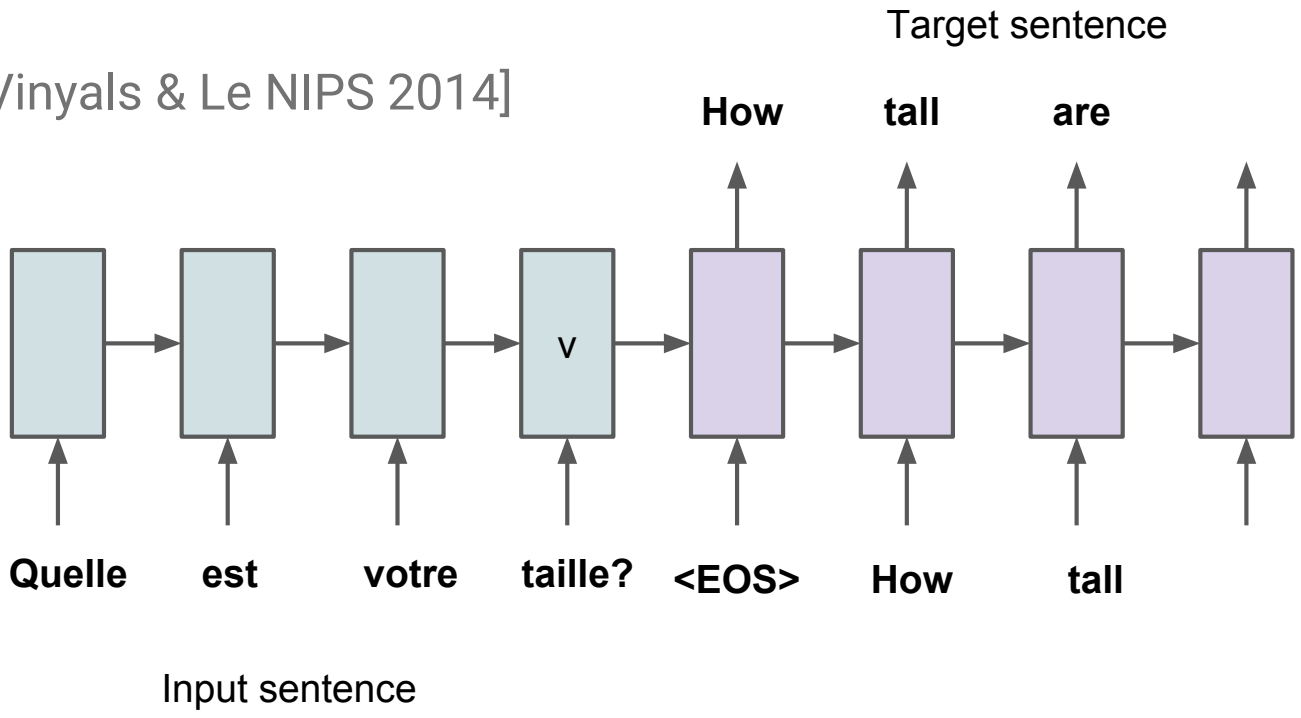
# Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]



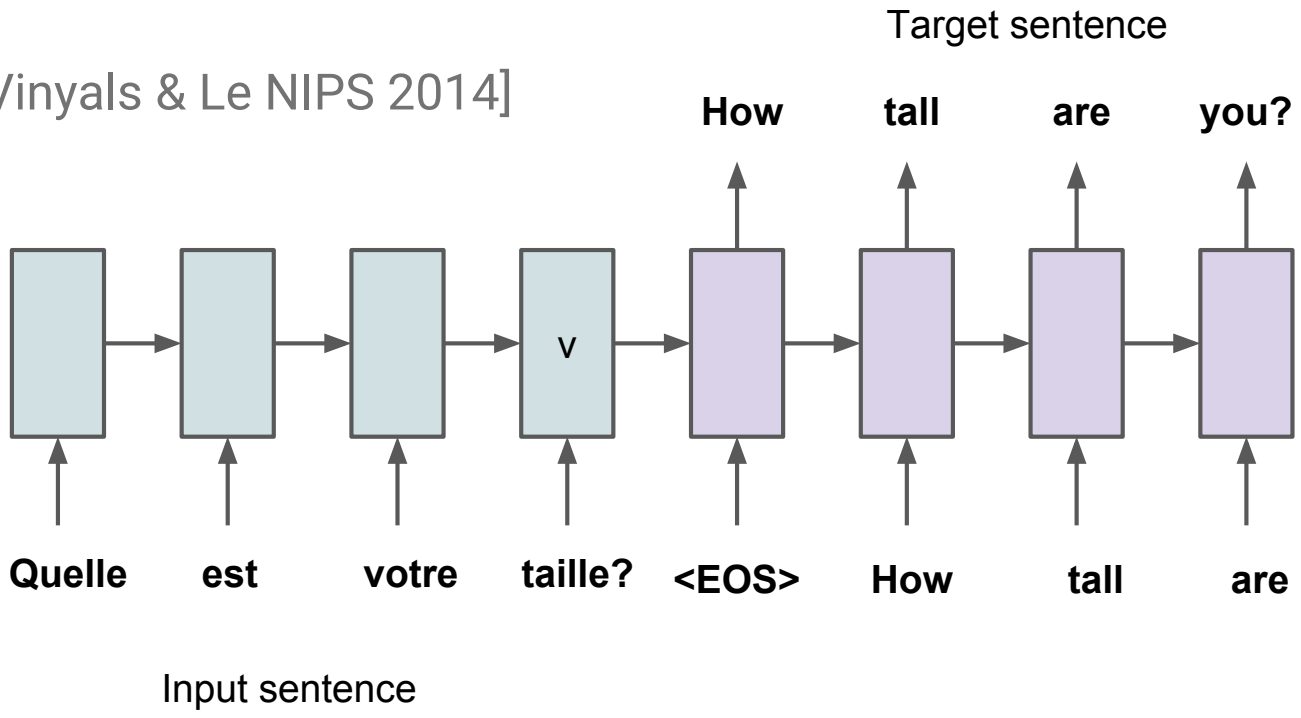
# Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]



# Sequence-to-Sequence Model: Machine Translation

[Sutskever & Vinyals & Le NIPS 2014]

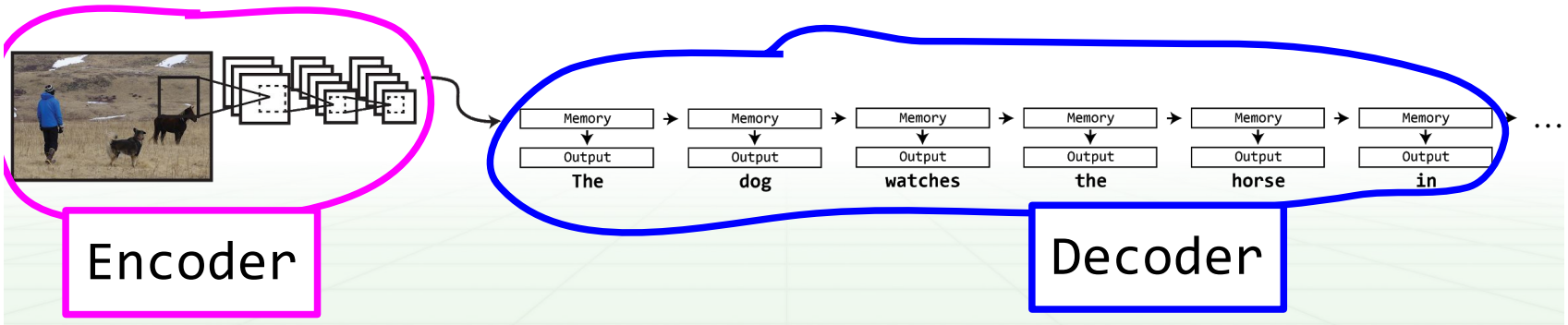


Can also have
**Multi-modal translation**

# Image Captioning with RNNs

Given an image:

- Extract features using CNN
- Feed into RNN
- Generate sentences!



# The Good

a group of people standing around a room with remotes
logprob: -9.17

a young boy is holding a baseball bat
logprob: -7.61

a cow is standing in the middle of a street
logprob: -8.84

a cat is sitting on a toilet seat
logprob: -7.79

a display case filled with lots of different types of donuts
logprob: -7.78

a group of people sitting at a table with wine glasses
logprob: -6.71

# The Bad

a man standing next to a clock on a wall
logprob: -10.08

a young boy is holding a baseball bat
logprob: -7.65

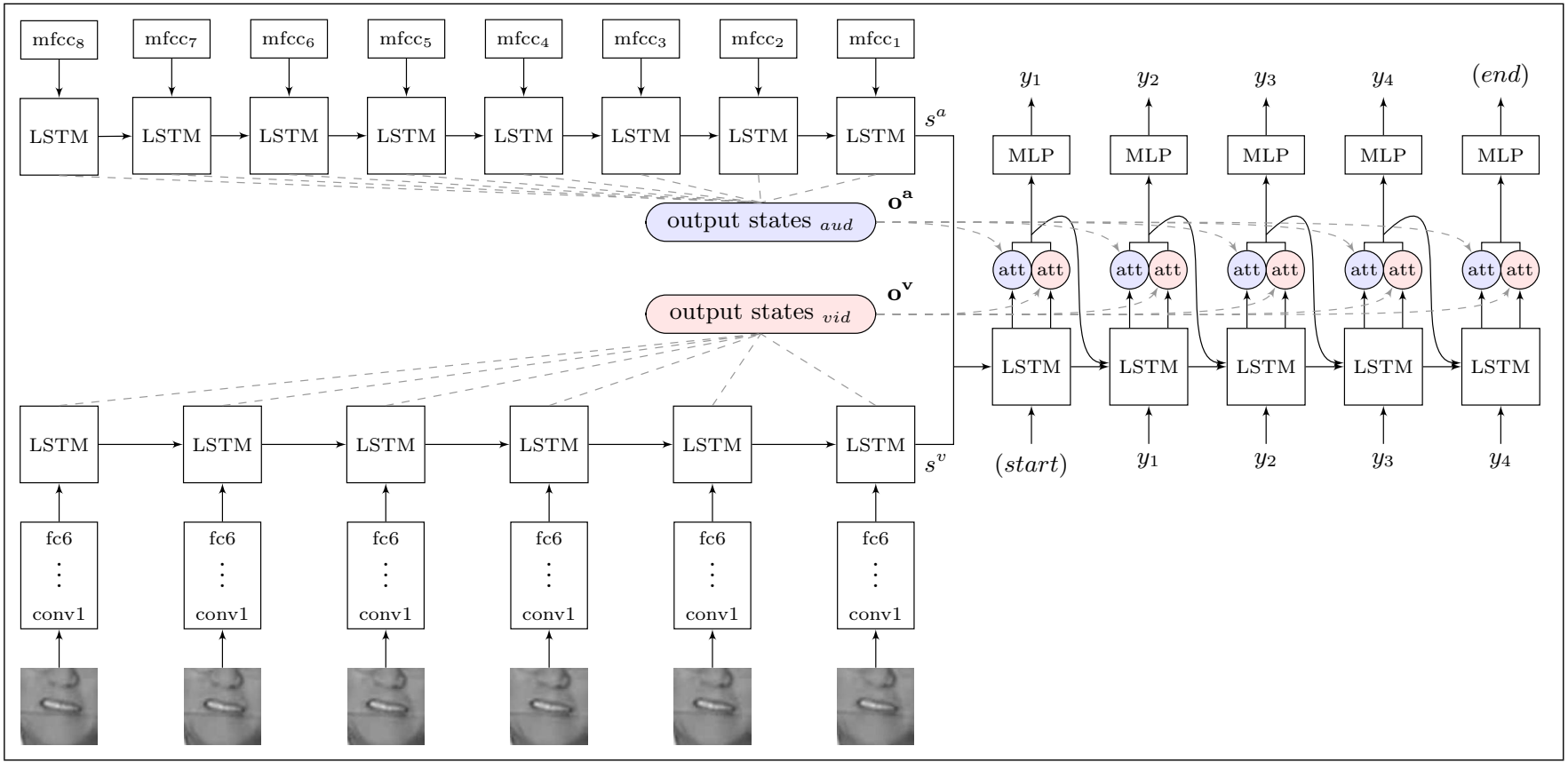
a cat is sitting on a couch with a remote control
logprob: -12.45

a baby laying on a bed with a stuffed bear
logprob: -8.66

a table with a plate of food and a cup of coffee
logprob: -9.93

a young boy is playing frisbee in the park
logprob: -9.52

# Translation from Speech & Video → Text



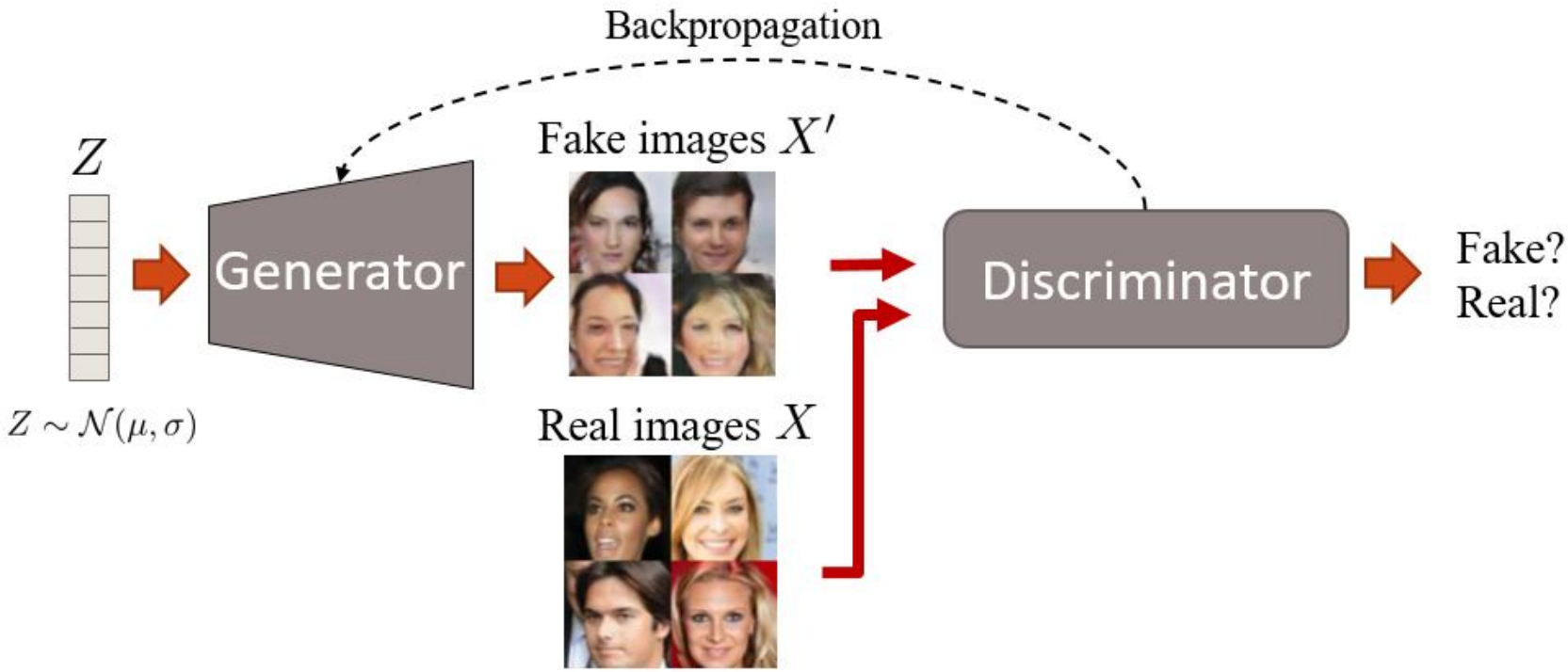

**Lip Reading Sentences in the Wild** Chung, Senior, Vinyals, Zisserman, arXiv Nov 2016

Cool next wave of ideas

**Generative networks**

# Generative Adversarial Network

- Idea originally proposed by Ian Goodfellow in NIPS '14.
- Have two networks: a **generator** and a **discriminator**. Both trained simultaneously.
- Want generator to fool discriminator by creating realistic images and want discriminator not to be fooled by generator.




**Overview of GAN training**

**Picture credit**: Blog post - *Fantastic GANs and where to find them* by Guim Perarnau.

# Improved implementation - DCGANs

**Bedroom images generated by convolutional Generator Network.**

**Figure credit**: *Unsupervised representation learning with deep convolutional generative adversarial networks.* A. Radford, L. Metz, S. Chintala, ICLR 2016.

# Amazing recent GAN results

**Faces automatically created by a Generator Network.**

**Figure credit**: *Progressive Growing of GANs For Improved Quality, Stability, and Variation.* T. Karras, T. Aila, S. Laine, J. Lehtinen, ICLR 2018.

# Creative variations of GANs

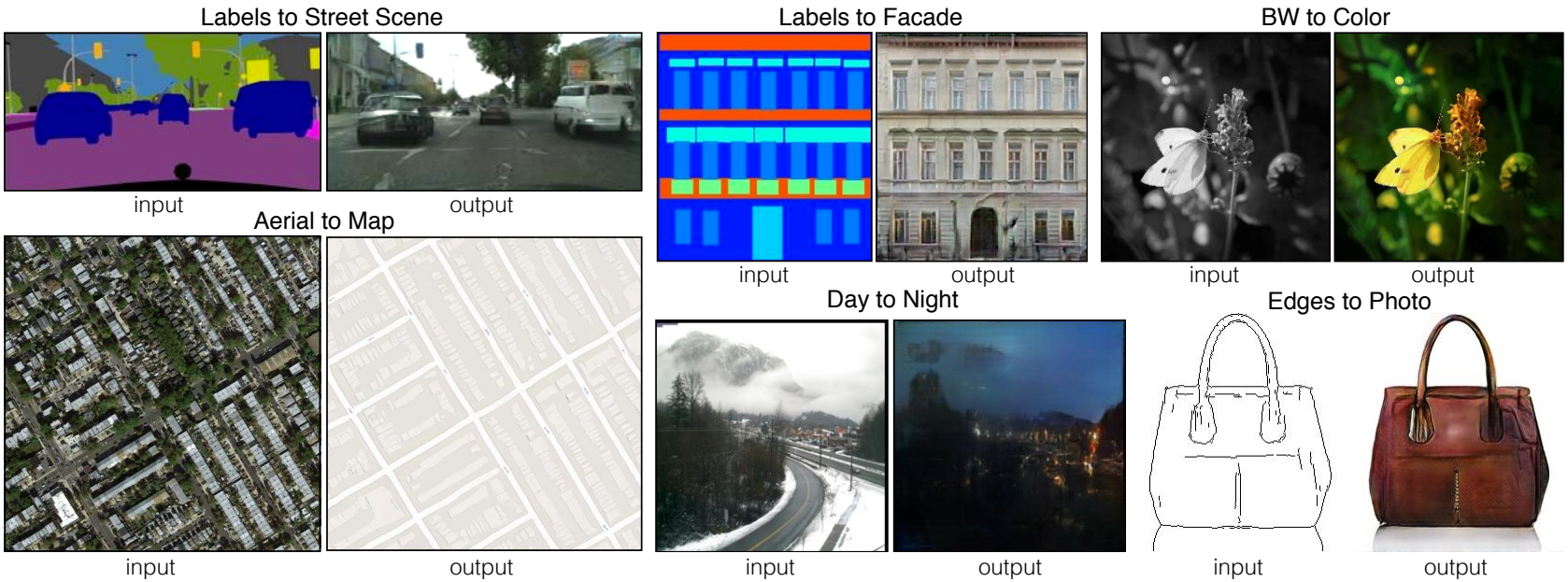
this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

**Translate descriptive text to an image.**

**Figure credit**: *Generative Adversarial Text to Image Synthesis.* S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, ICML 2016.

# Creative variations of GANs




**Translate one image format to another.**

**Figure credit**: *Image-to-Image Translation with Conditional Adversarial Networks.* P. Isola, J.-Y. Zhu, T. Zhou, A. Efros, CVPR 2017.

# Creative variations of GANs - Cycle GANs

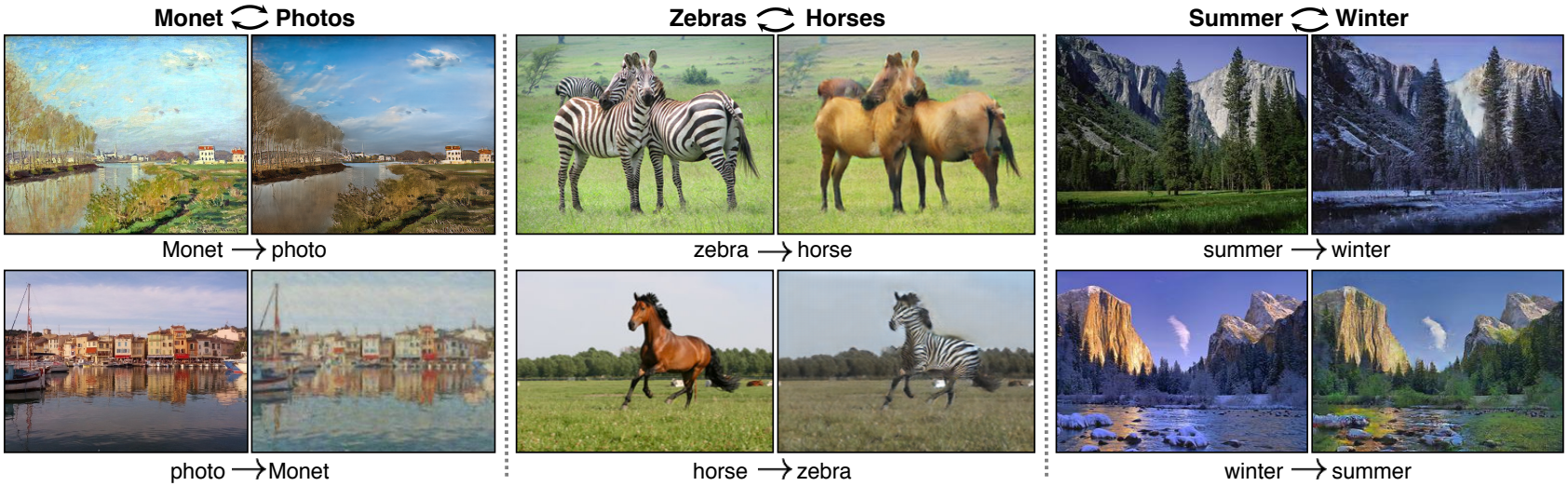



**Image-to-Image Translation.**

**Figure credit**: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, J.-Y. Zhu, T. Park, P. Isola, A. Efros, ICCV 2017.