

Lecture 25: Conclusion Open Problems

Reminder: A5

Recurrent networks, attention, Transformers

Due on **Tuesday 4/12**, 11:59pm ET

A6

Covers image generation and generative models:

Generative Models: GANs and VAEs

Network visualization: saliency maps, adversarial examples, class visualizations

Style Transfer

Due Tuesday 4/26, 11:59pm ET

YOU CANNOT USE LATE DAYS ON A6!!!!

Mini-Project Submission

Mini-project due **Monday, 4/25 11:59 ET**

Submit project here:

<https://forms.gle/CauLnF9kTuv6JGZA9>

Today:
Course Recap
What's next?

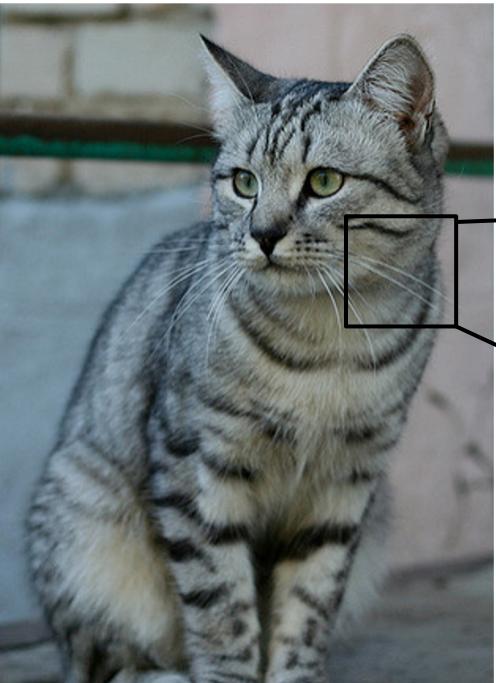
This Course: Deep Learning for Computer Vision

Deep Learning for Computer Vision

Building artificial systems that process,
perceive, and reason about visual data

Problem: Semantic Gap

What you see



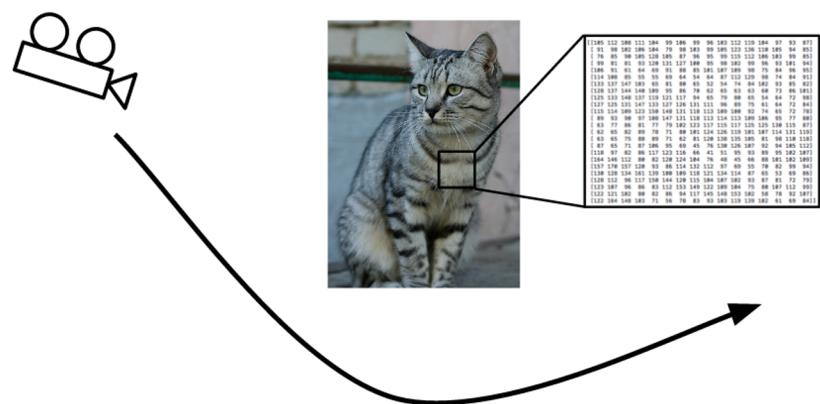
What computer sees

[105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
[91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
[76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
[99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
[106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
[114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
[133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
[128 137 144 140 109 95 86 70 62 65 63 63 60 73 86 101]
[125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
[127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
[115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
[89 93 90 97 108 147 131 118 113 114 113 109 106 95 77 80]
[63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
[62 65 82 89 78 71 80 101 124 126 119 101 107 114 131 119]
[63 65 75 88 89 71 62 81 120 138 135 105 81 98 110 118]
[87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
[118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
[164 146 112 80 82 120 124 104 76 48 45 66 88 101 102 109]
[157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
[130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
[128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
[123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
[122 121 102 80 82 86 94 117 145 148 153 102 58 78 92 107]
[122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]

This image by [Nikita](#) is
licensed under [CC-BY 2.0](#)

Problem: Visual Data is Complex!

Viewpoint

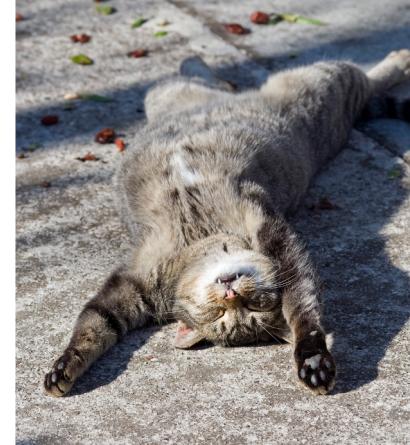


Illumination



[This image is CC0 1.0 public domain](#)

Deformation



[This image by Umberto Salvagnin is licensed under CC-BY 2.0](#)

Occlusion



[This image by jonsson is licensed under CC-BY 2.0](#)

Clutter



[This image is CC0 1.0 public domain](#)

Intraclass Variation



[This image is CC0 1.0 public domain](#)

Machine Learning: Data-Driven Approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

Example training set

```
def train(images, labels):  
    # Machine learning!  
    return model
```

```
def predict(model, test_images):  
    # Use model to predict labels  
    return test_labels
```

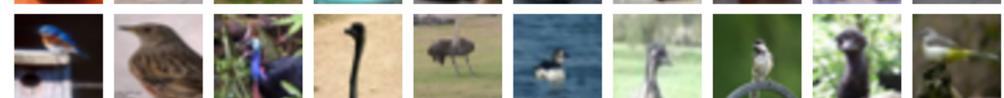
airplane



automobile



bird



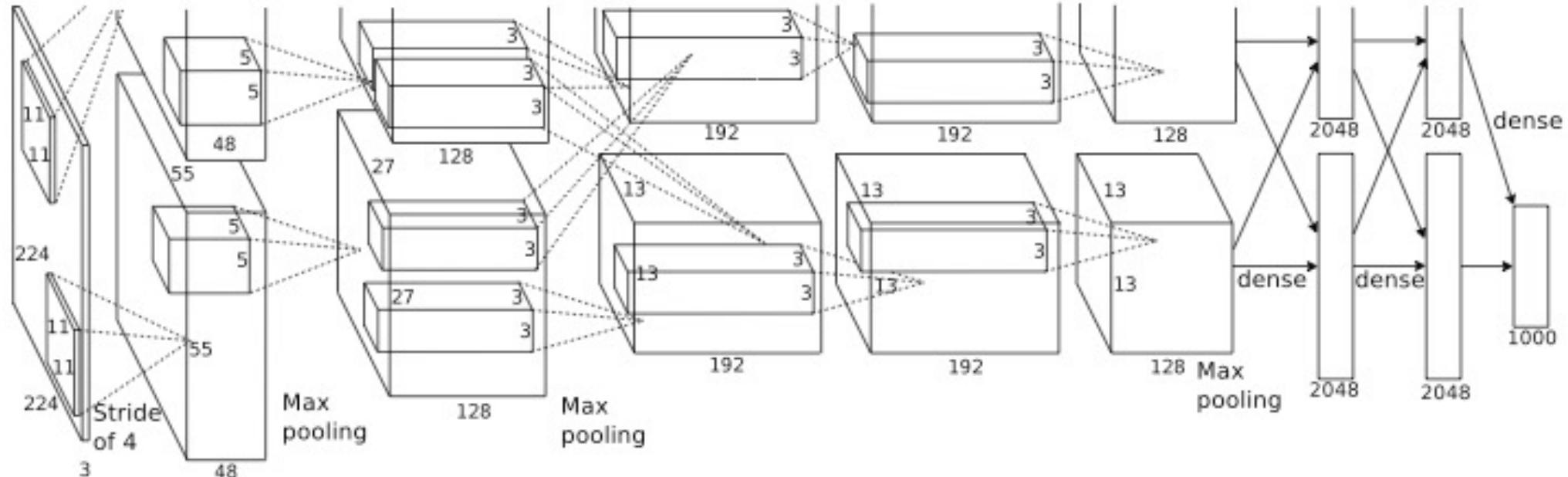
cat



deer



Model: Deep Convolutional Networks



Krizhevsky, Sutskever, and Hinton, NeurIPS 2012

IMAGENET Large Scale Visual Recognition Challenge

The Image Classification Challenge:
1,000 object classes
1,431,167 images

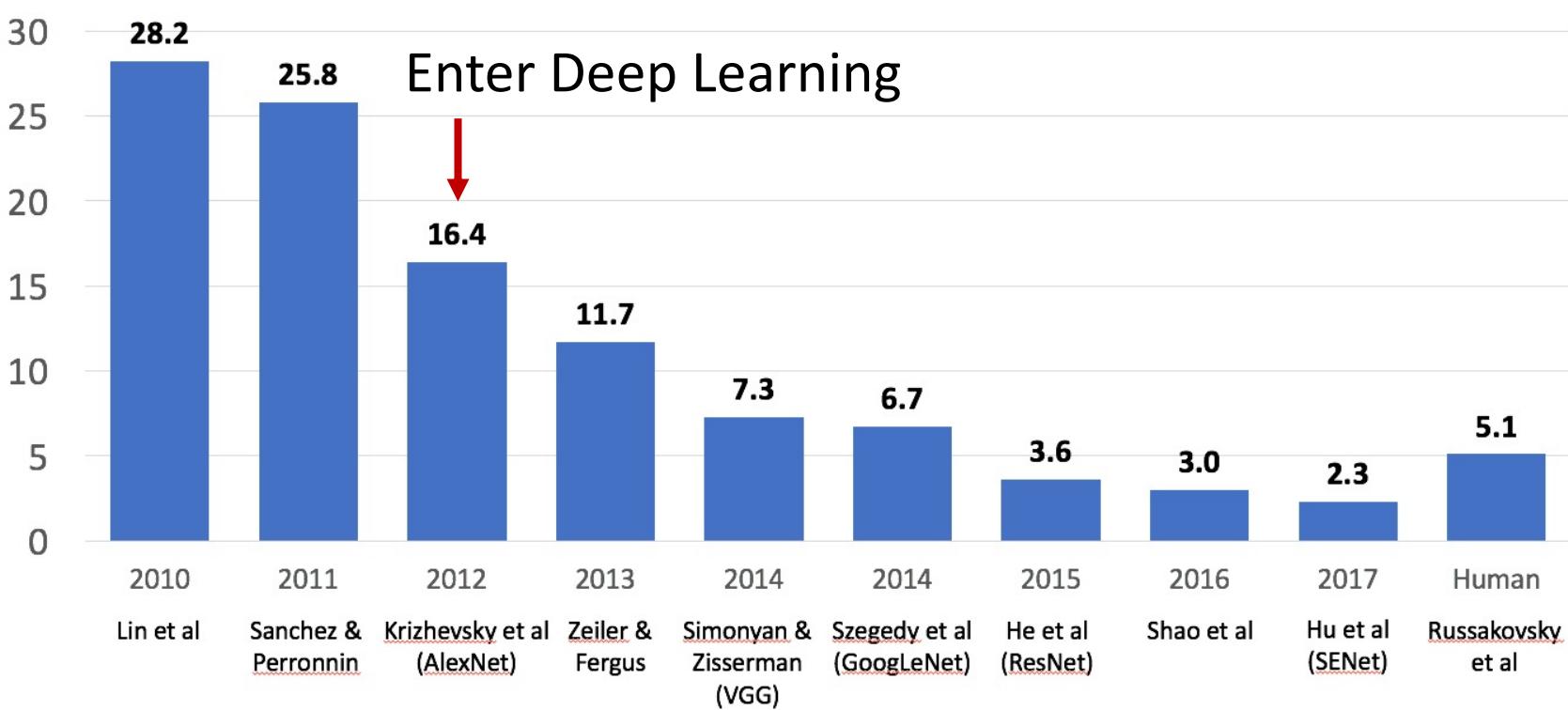


Output:
Scale
T-shirt
Steel drum
Drumstick
Mud turtle

Deng et al, 2009
Russakovsky et al. IJCV 2015

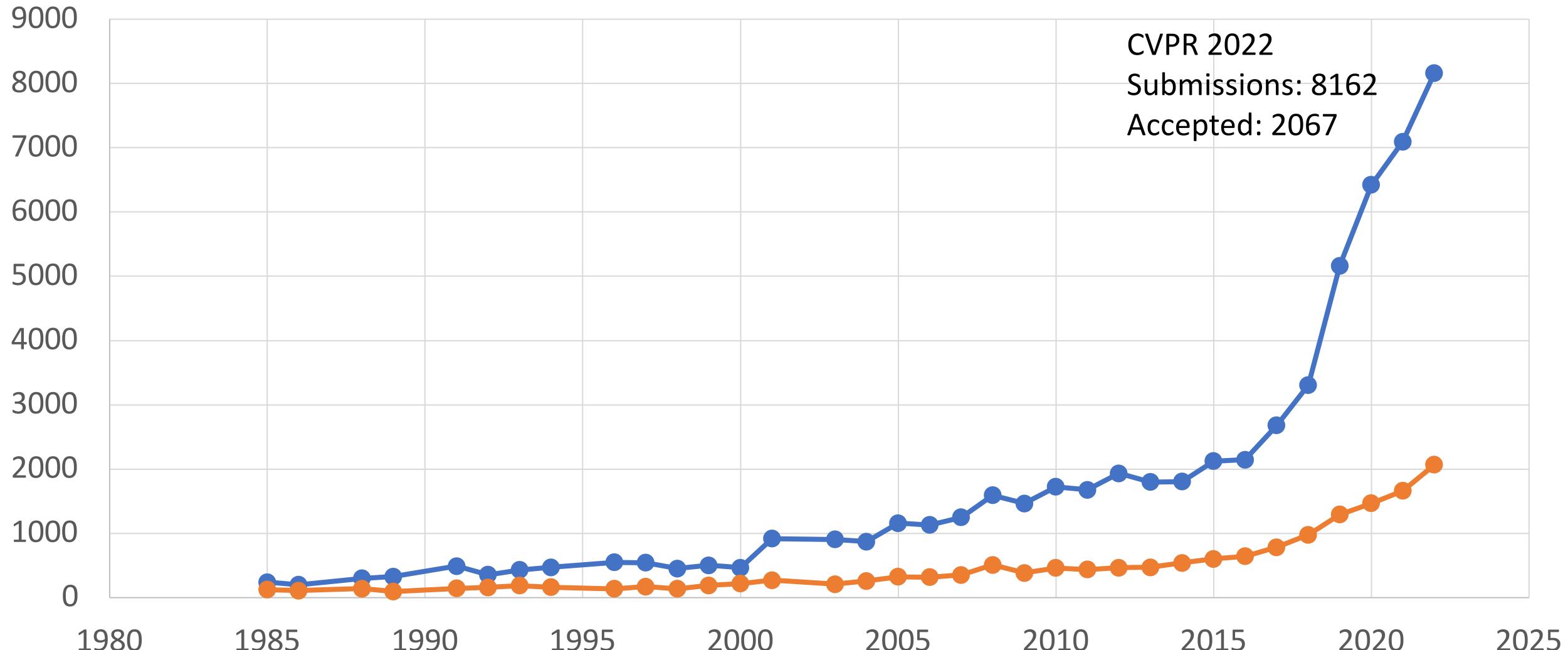


Large Scale Visual Recognition Challenge

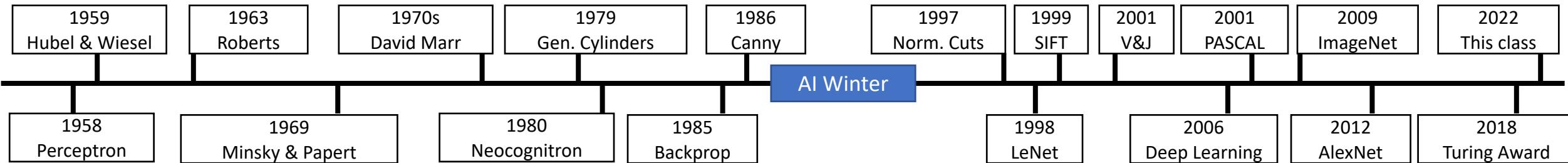


CVPR Papers

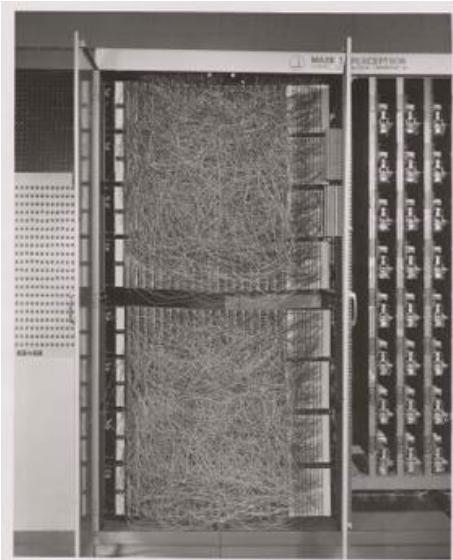
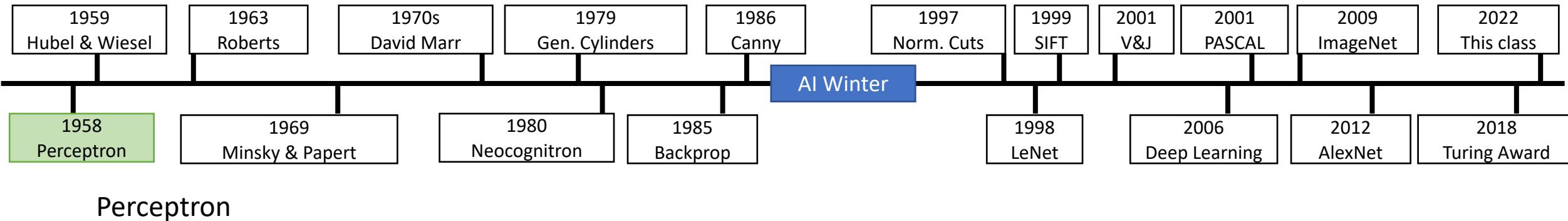
Submitted Accepted



Deep Learning wasn't invented overnight!

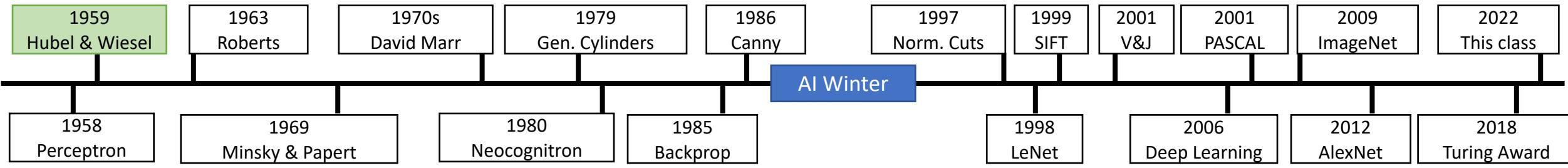


Deep Learning wasn't invented overnight!

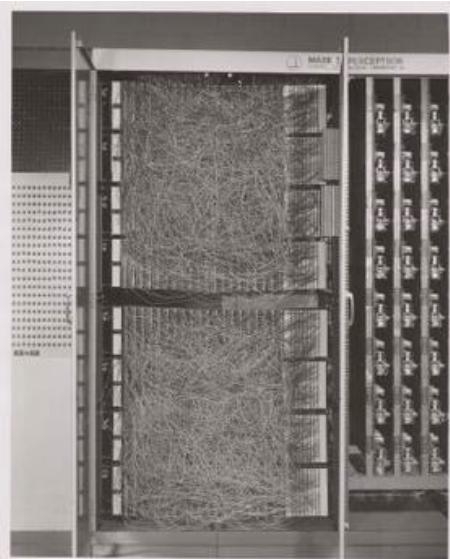


Frank Rosenblatt, ~1957

Deep Learning wasn't invented overnight!

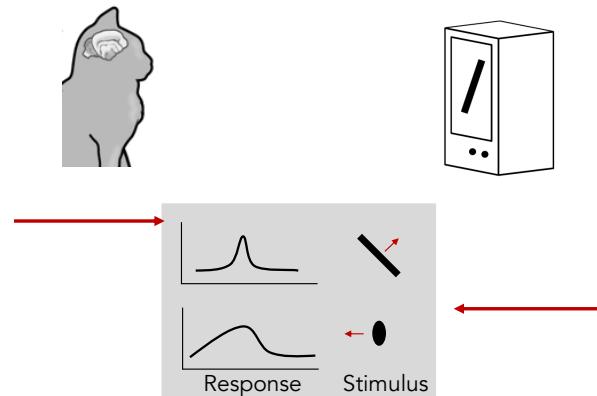


Perceptron



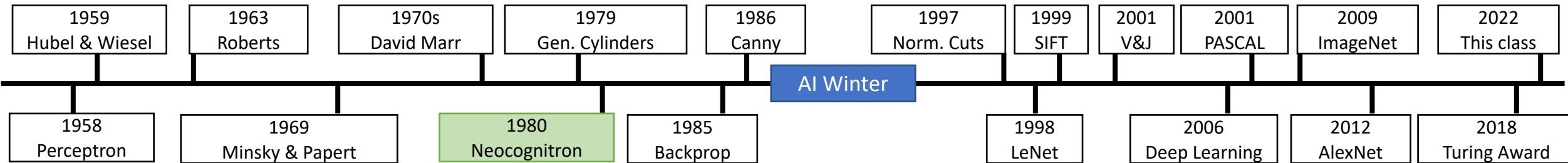
Frank Rosenblatt, ~1957

Simple and Complex cells

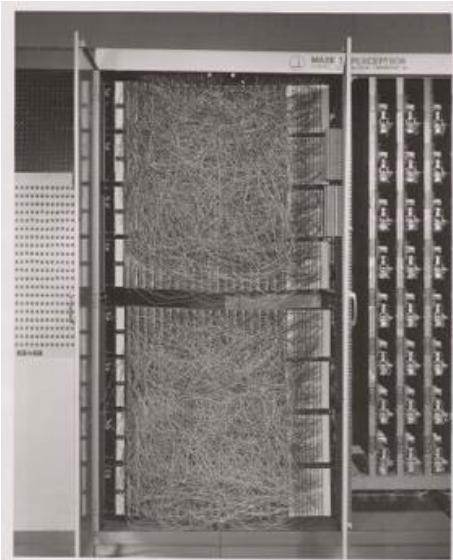


Hubel and Wiesel, 1959

Deep Learning wasn't invented overnight!

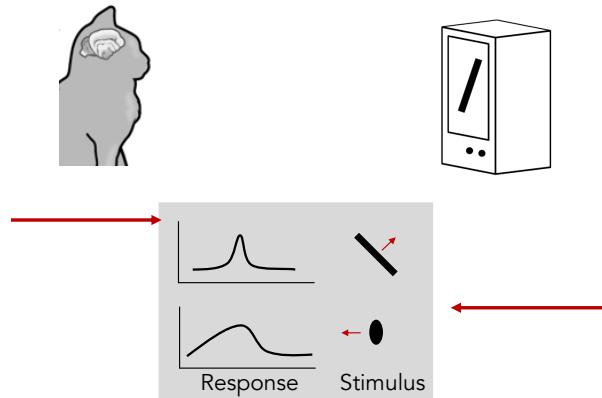


Perceptron



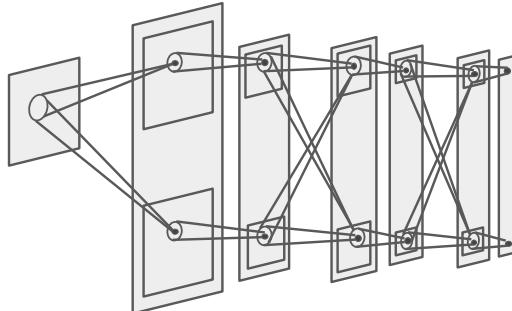
Frank Rosenblatt, ~1957

Simple and Complex cells



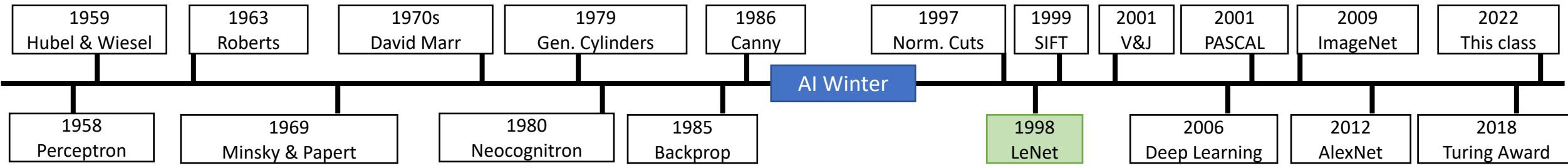
Hubel and Wiesel, 1959

Neocognitron

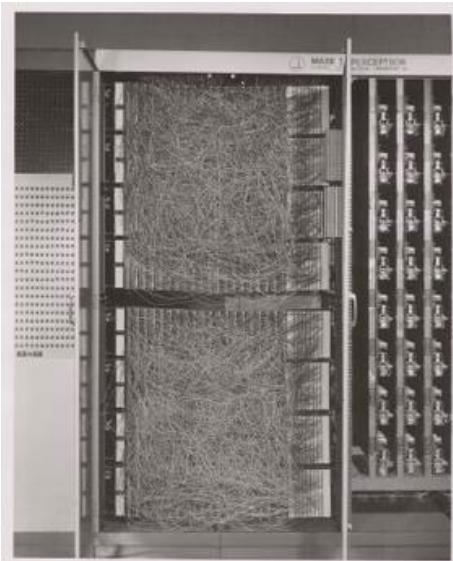


Fukushima, 1980

Deep Learning wasn't invented overnight!

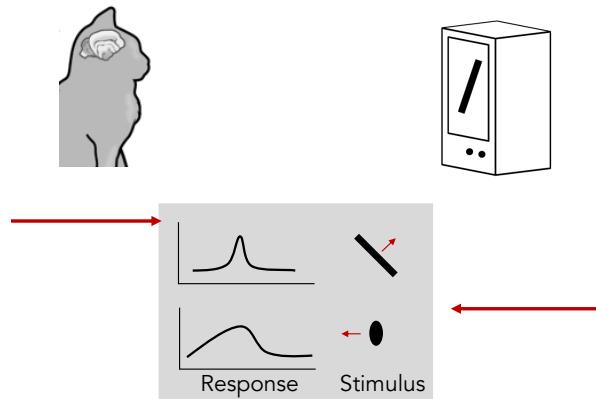


Perceptron



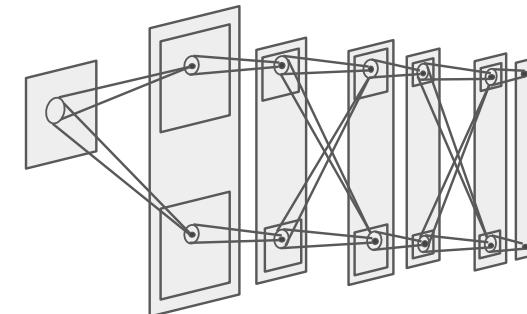
Frank Rosenblatt, ~1957

Simple and Complex cells



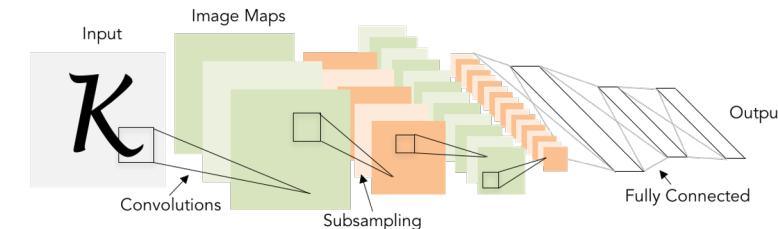
Hubel and Wiesel, 1959

Neocognitron



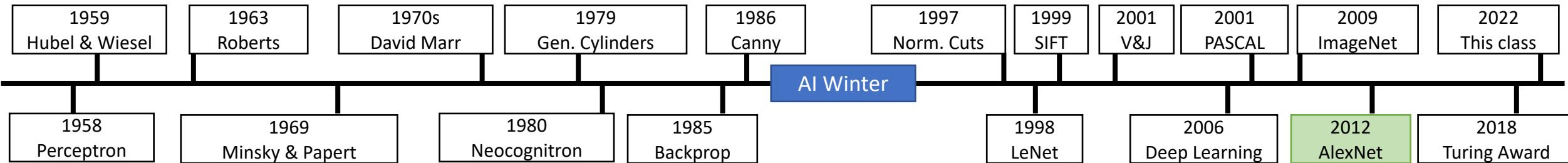
Fukushima, 1980

Convolutional Networks

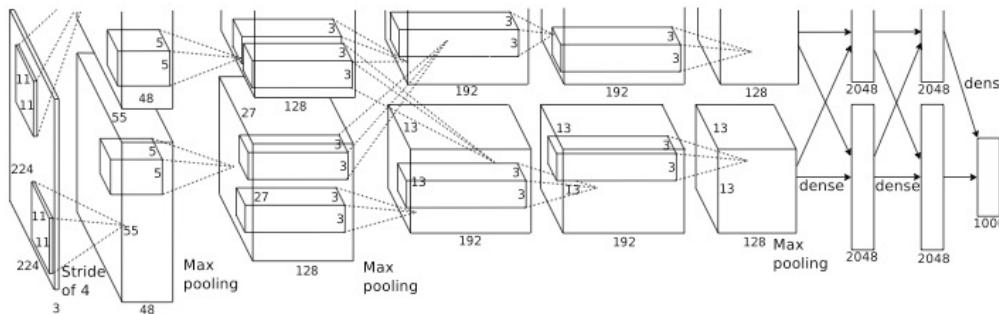


LeCun et al, 1998

Deep Learning wasn't invented overnight!

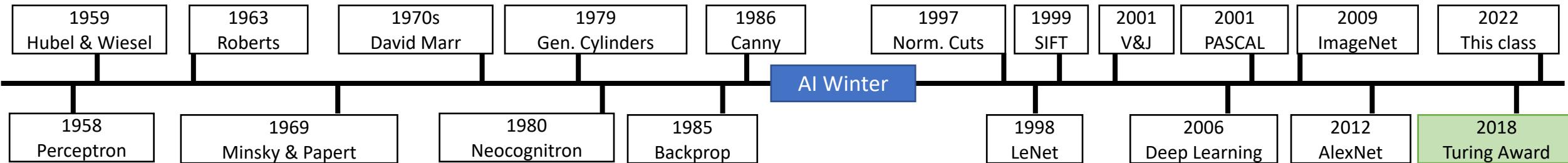


AlexNet



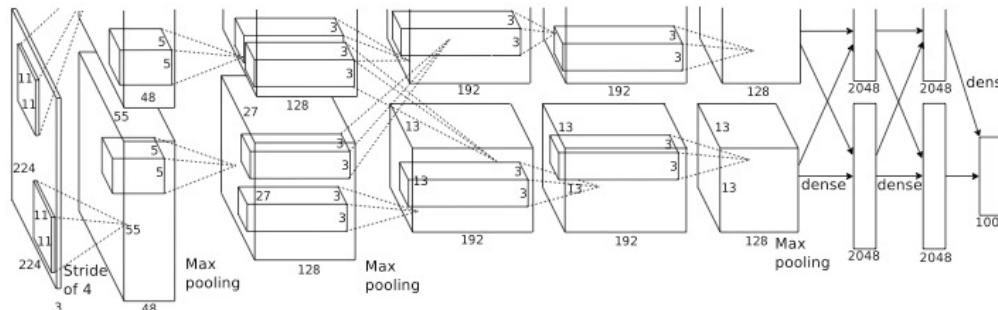
Krizhevsky, Sutskever, and Hinton, 2012

Deep Learning wasn't invented overnight!



2018 Turing Award

AlexNet



Krizhevsky, Sutskever, and Hinton, 2012



Yoshua
Bengio

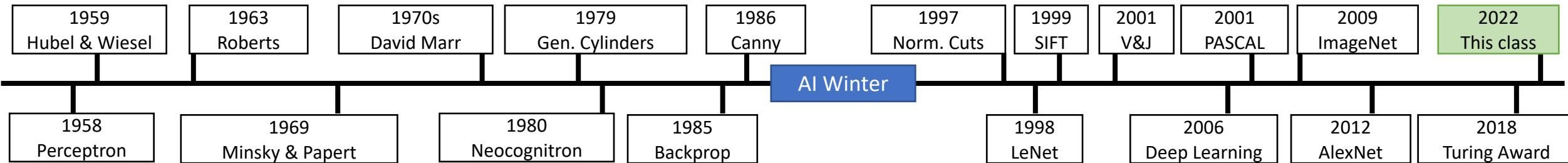


Geoffrey
Hinton



Yann
LeCun

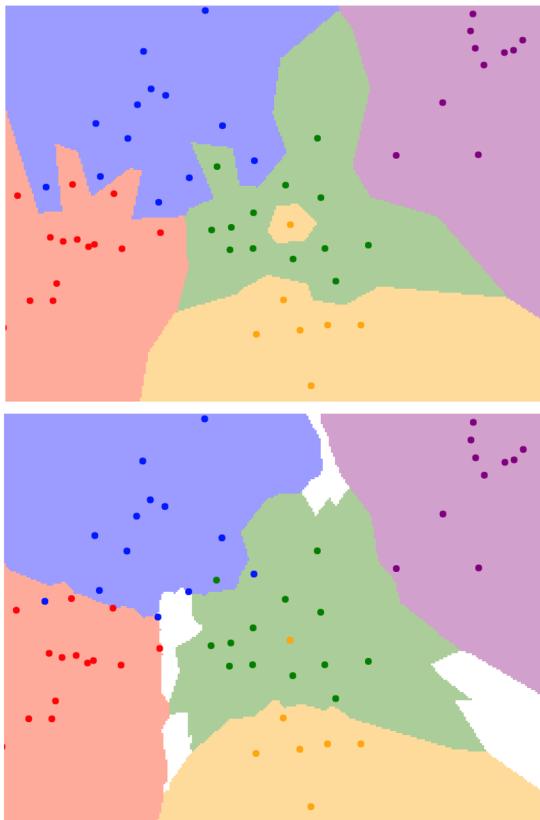
Deep Learning wasn't invented overnight!



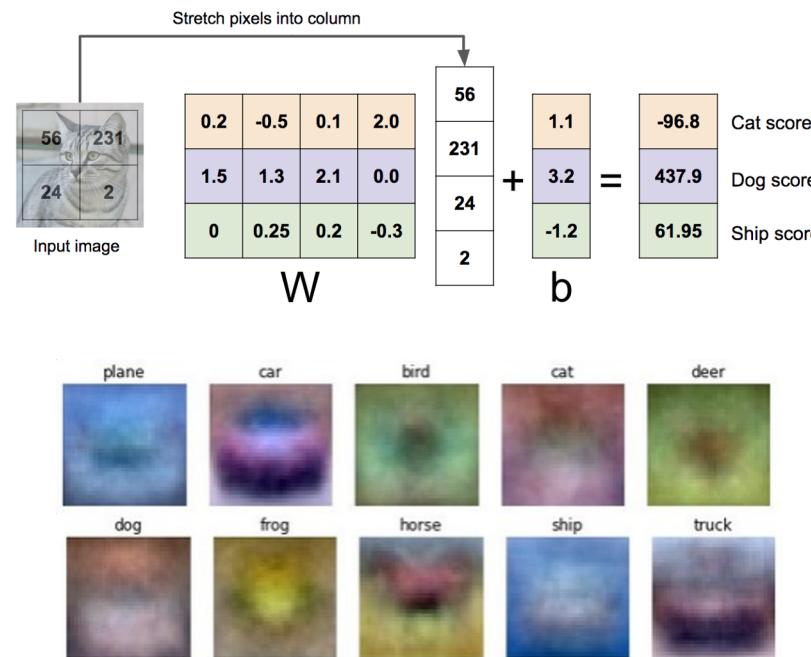
Winter 2022: This class

Simple Classifiers: kNN and Linear Classifiers

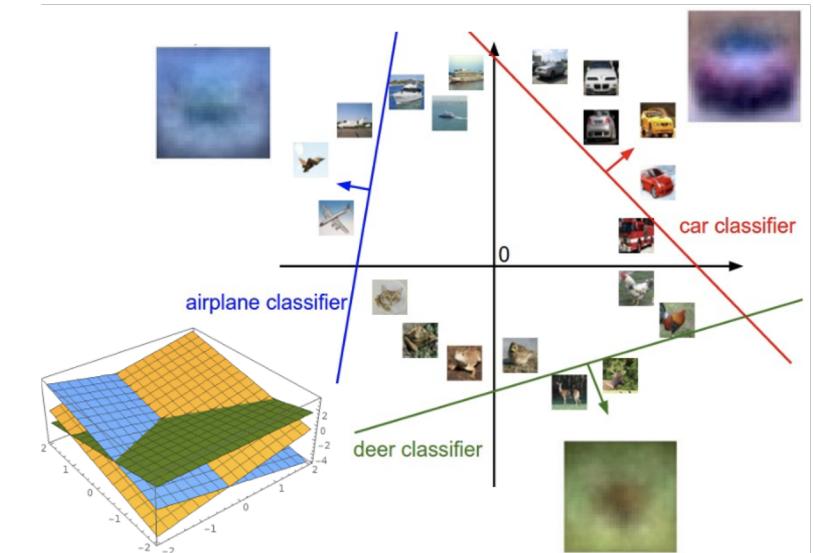
1-NN classifier



5-NN classifier



Linear Classifiers: $y = Wx + b$



Optimization with Gradient Descent

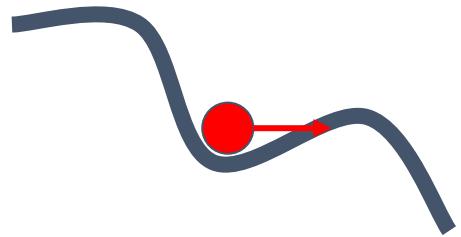


```
# Vanilla gradient descent
w = initialize_weights()
for t in range(num_steps):
    dw = compute_gradient(loss_fn, data, w)
    w -= learning_rate * dw
```

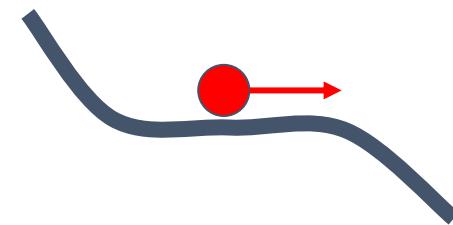
This image is CC0 1.0 public domain
Walking man image is CC0 1.0 public domain

Problems with Gradient Descent

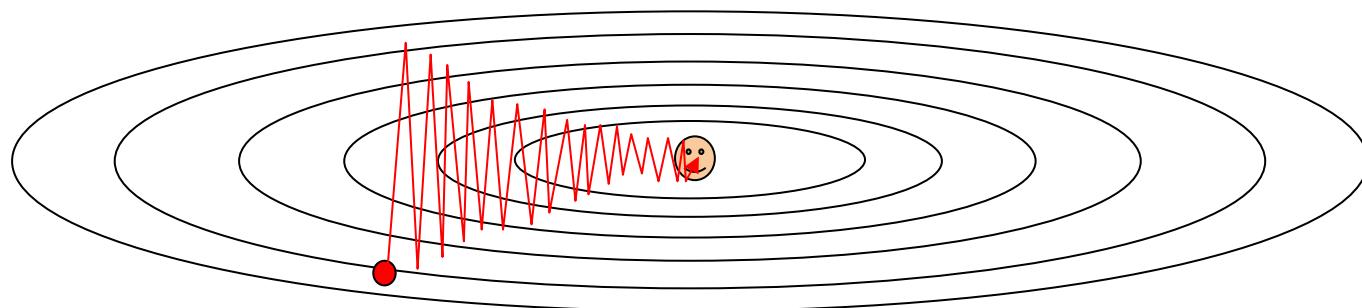
Local Minima



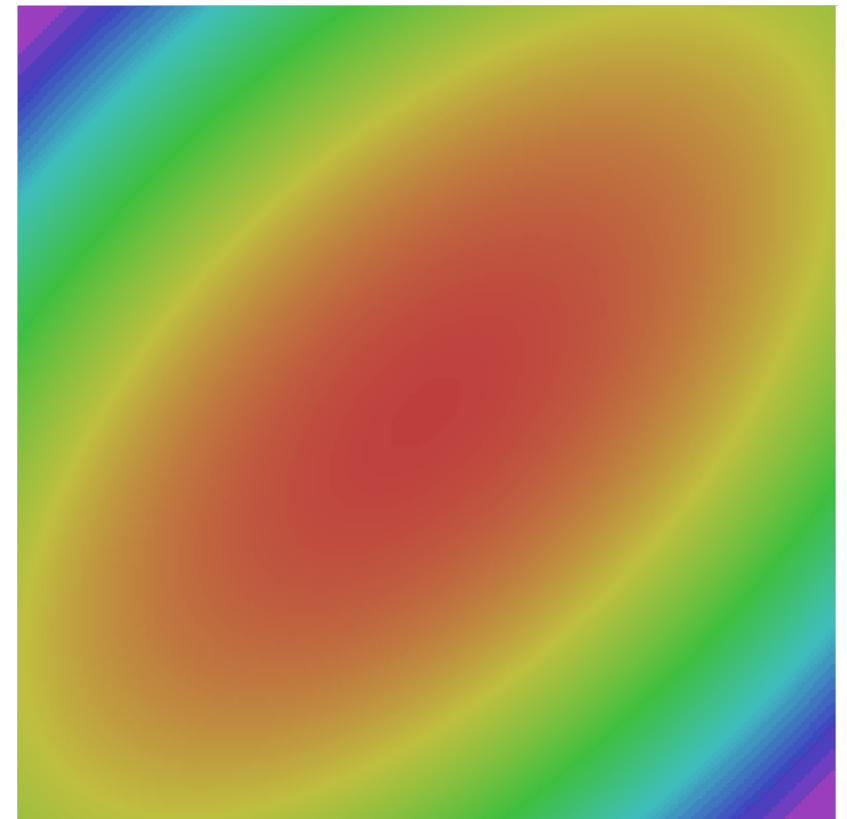
Saddle points



Poor Conditioning



Gradient Noise

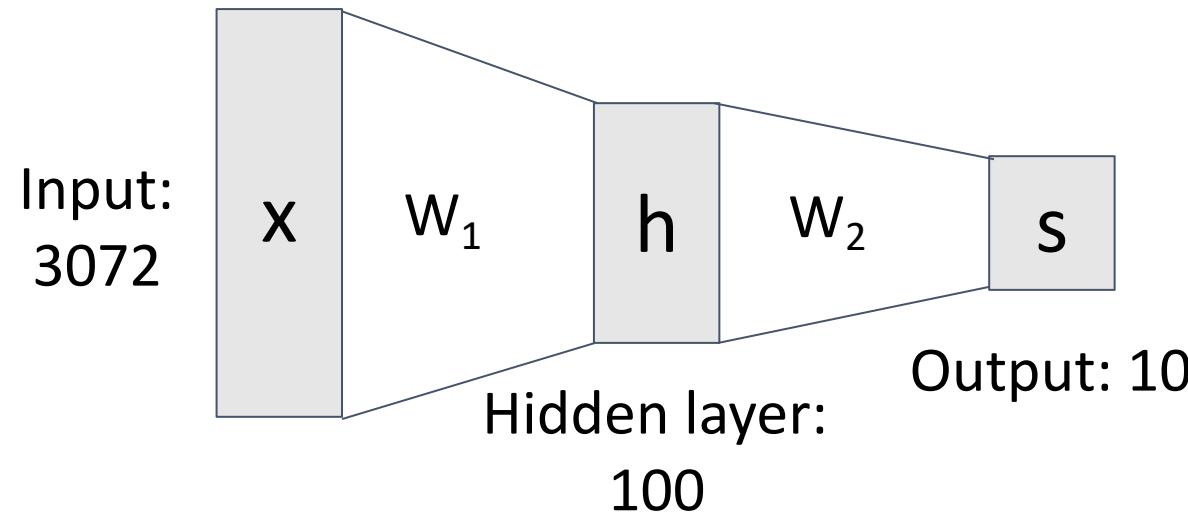


— SGD — SGD+Momentum

Gradient Descent Improvements

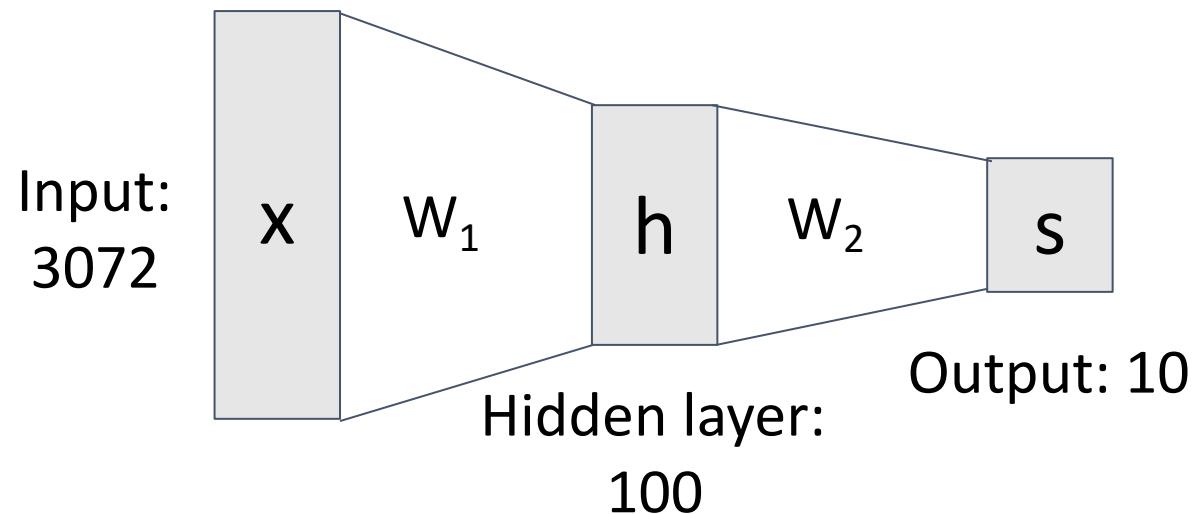
Algorithm	Tracks first moments (Momentum)	Tracks second moments (Adaptive learning rates)	Leaky second moments	Bias correction for moment estimates
SGD	✗	✗	✗	✗
SGD+Momentum	✓	✗	✗	✗
Nesterov	✓	✗	✗	✗
AdaGrad	✗	✓	✗	✗
RMSProp	✗	✓	✓	✗
Adam	✓	✓	✓	✓

More Complex Models: Neural Networks



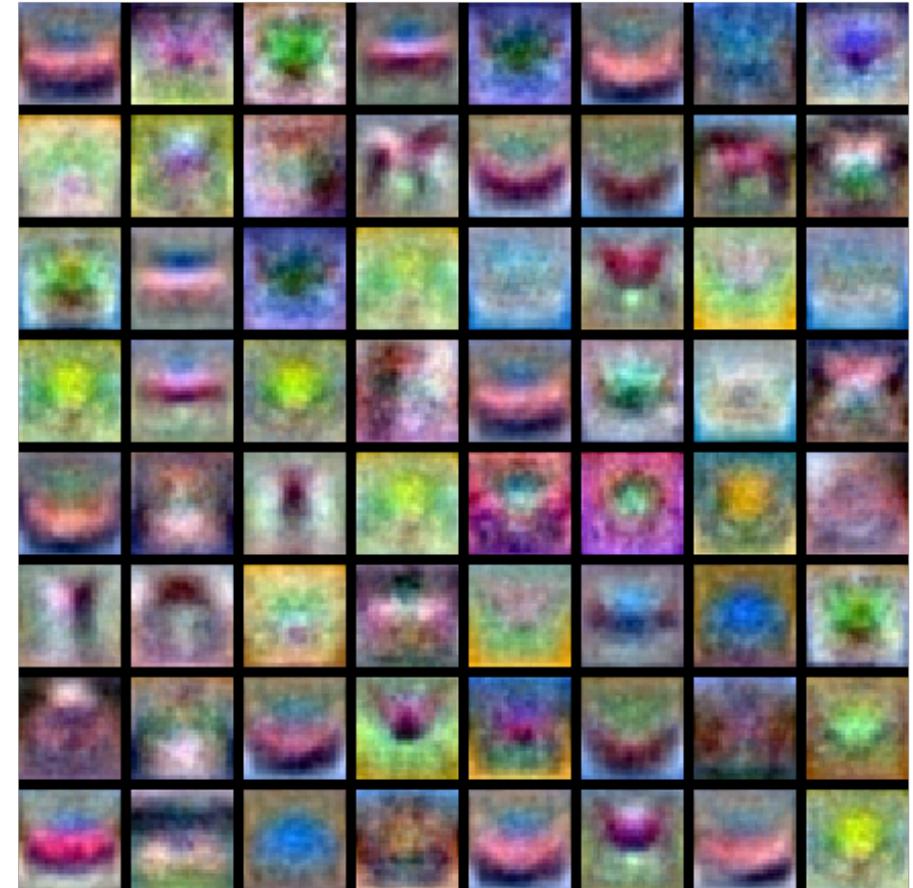
$$f = W_2 \max(0, W_1 x)$$

More Complex Models: Neural Networks



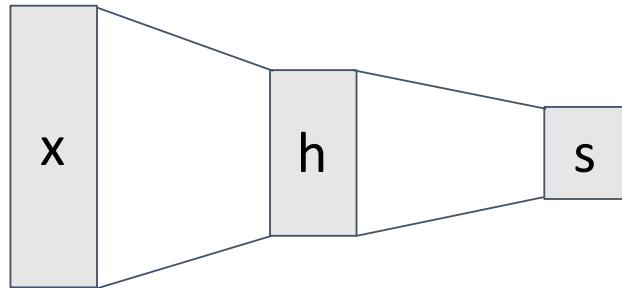
$$f = W_2 \max(0, W_1 x)$$

Learns bank of templates

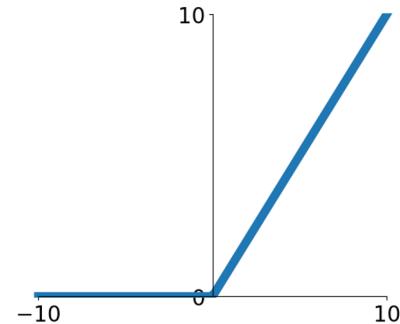


More Complex Models: Convolutional Networks

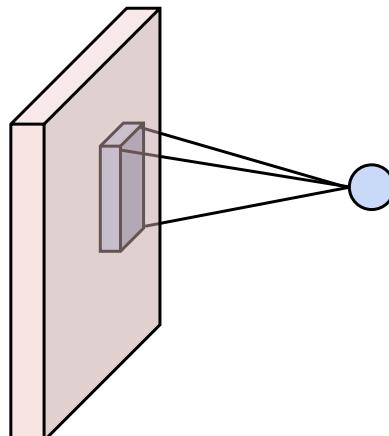
Fully-Connected Layers



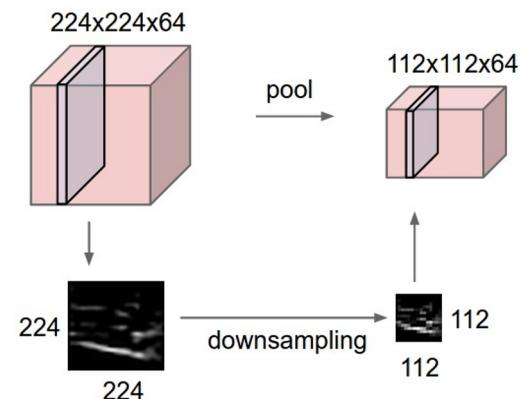
Activation Function



Convolution Layers

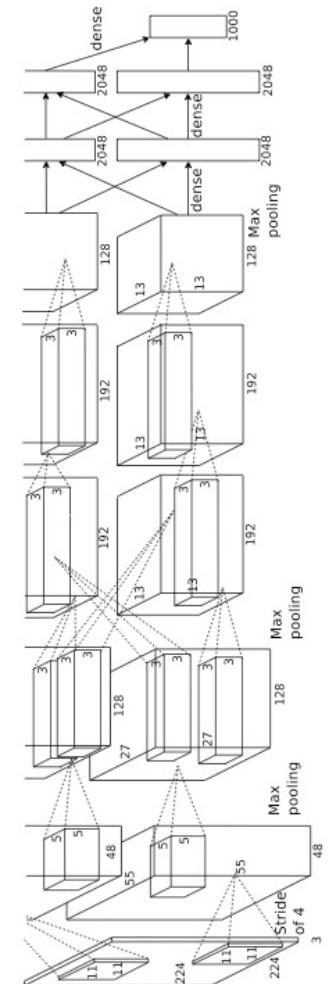


Pooling Layers

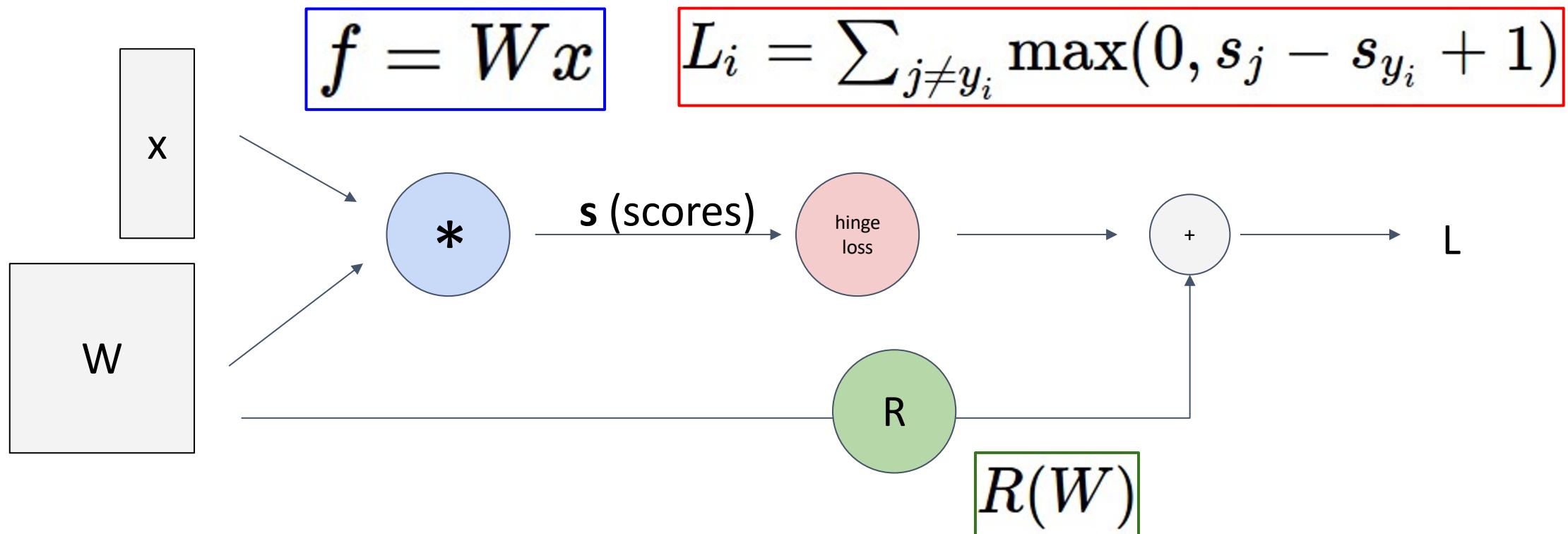


Normalization

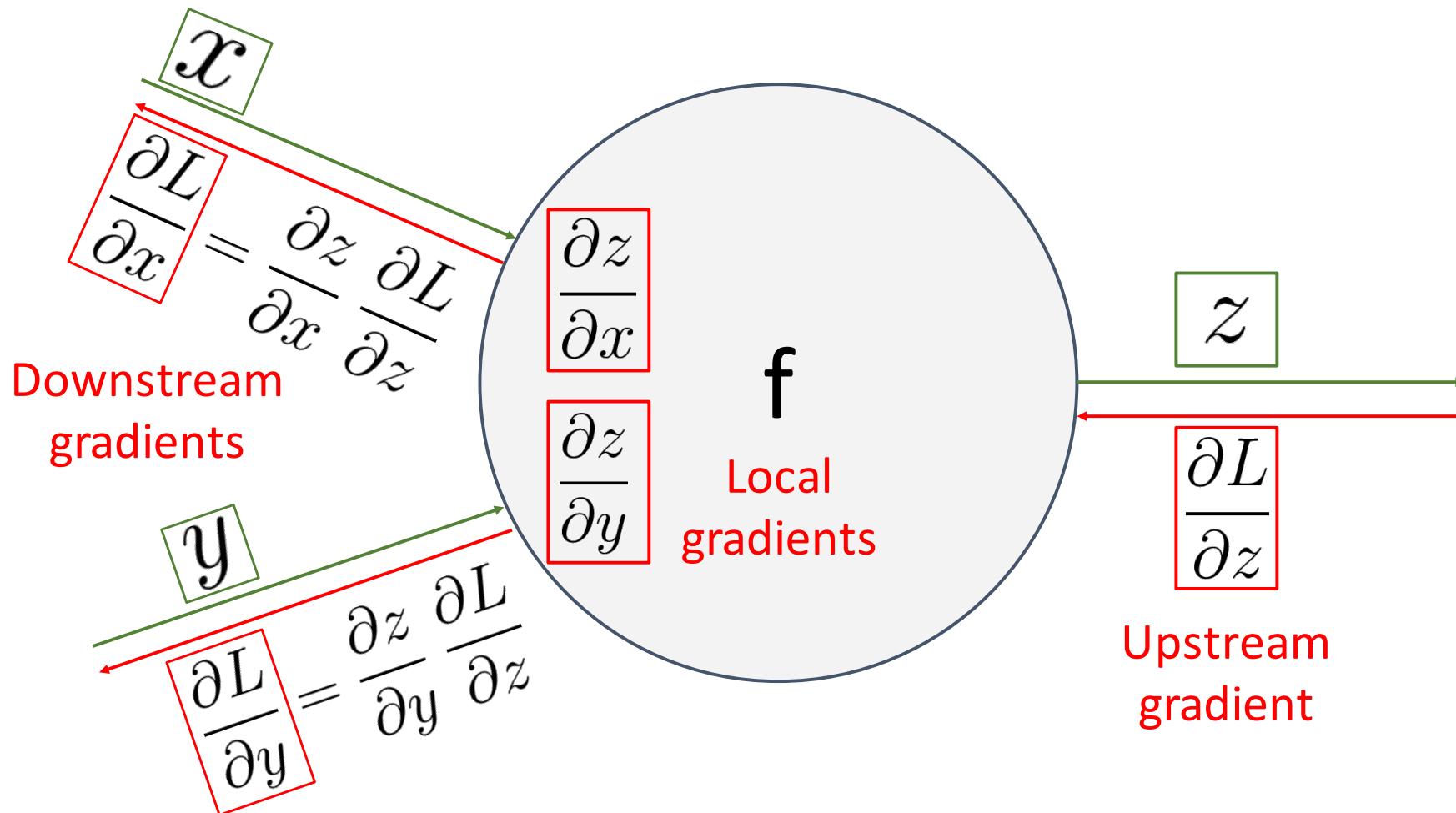
$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}}$$



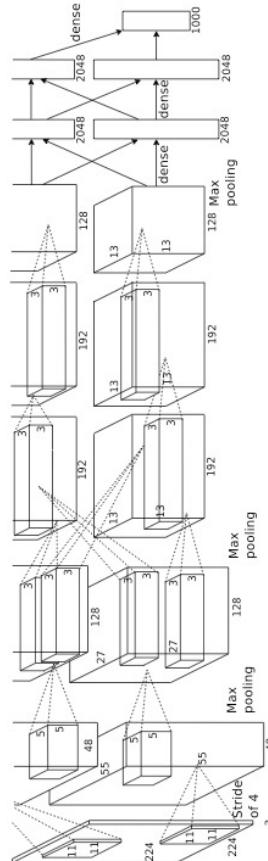
Representing Networks: Computational Graphs



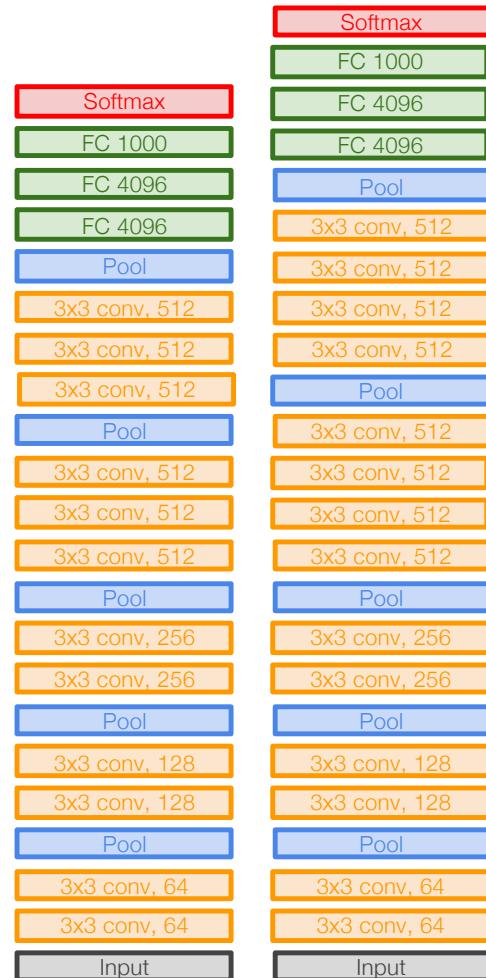
Computing Gradients: Backpropagation



CNN Architectures

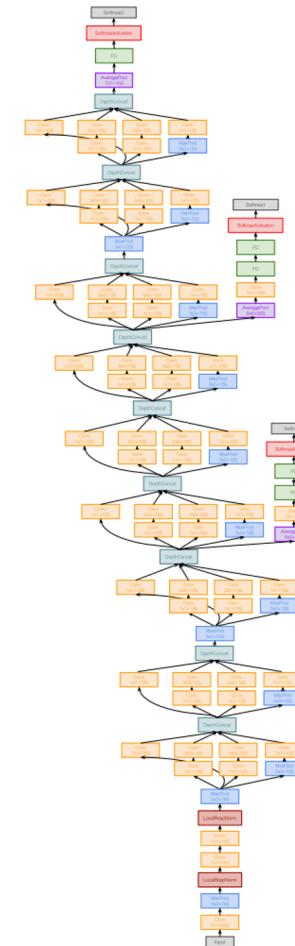


AlexNet

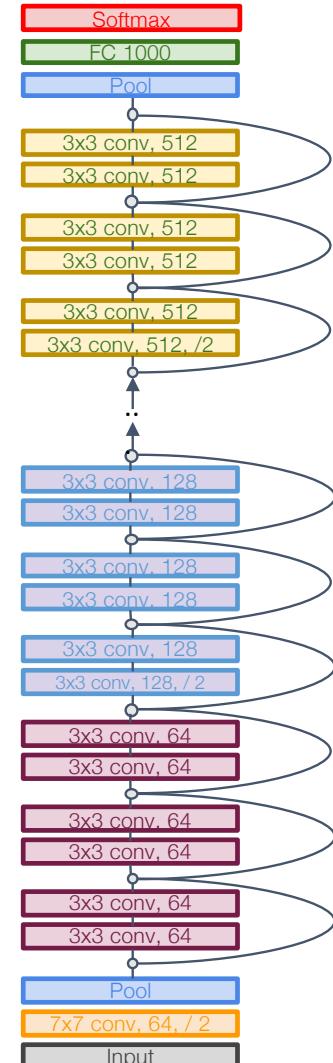


VGG16

VGG19

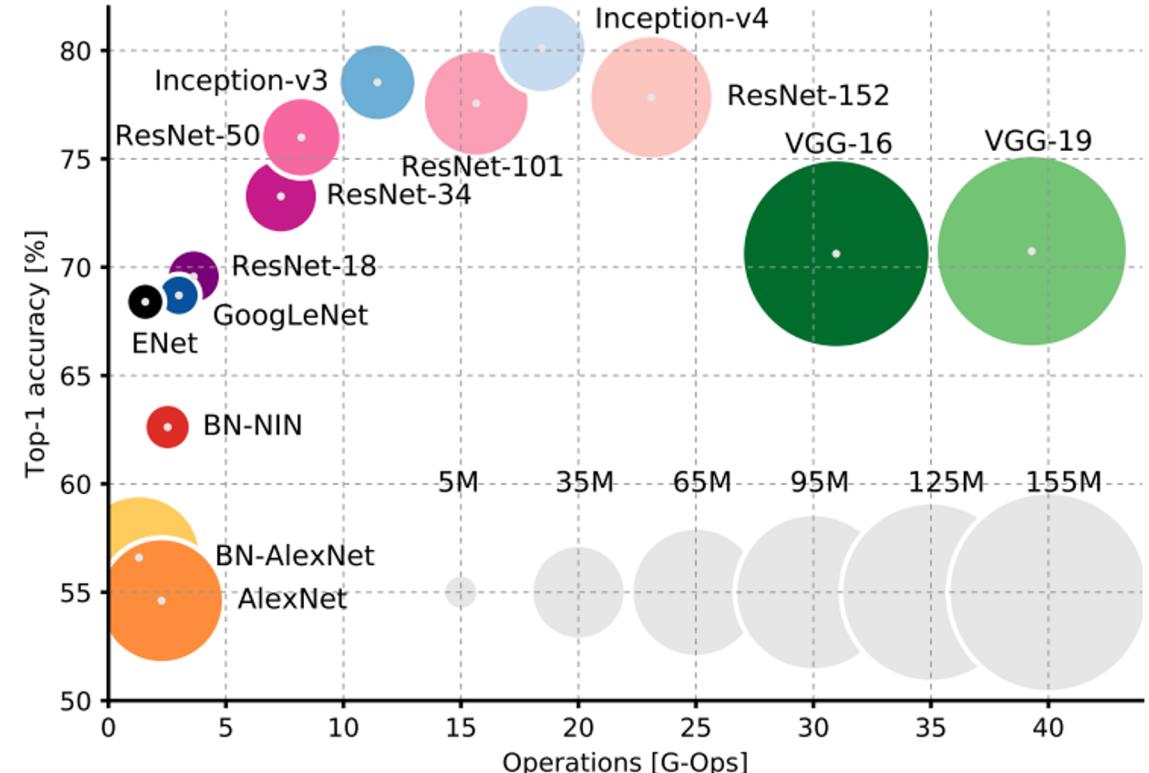
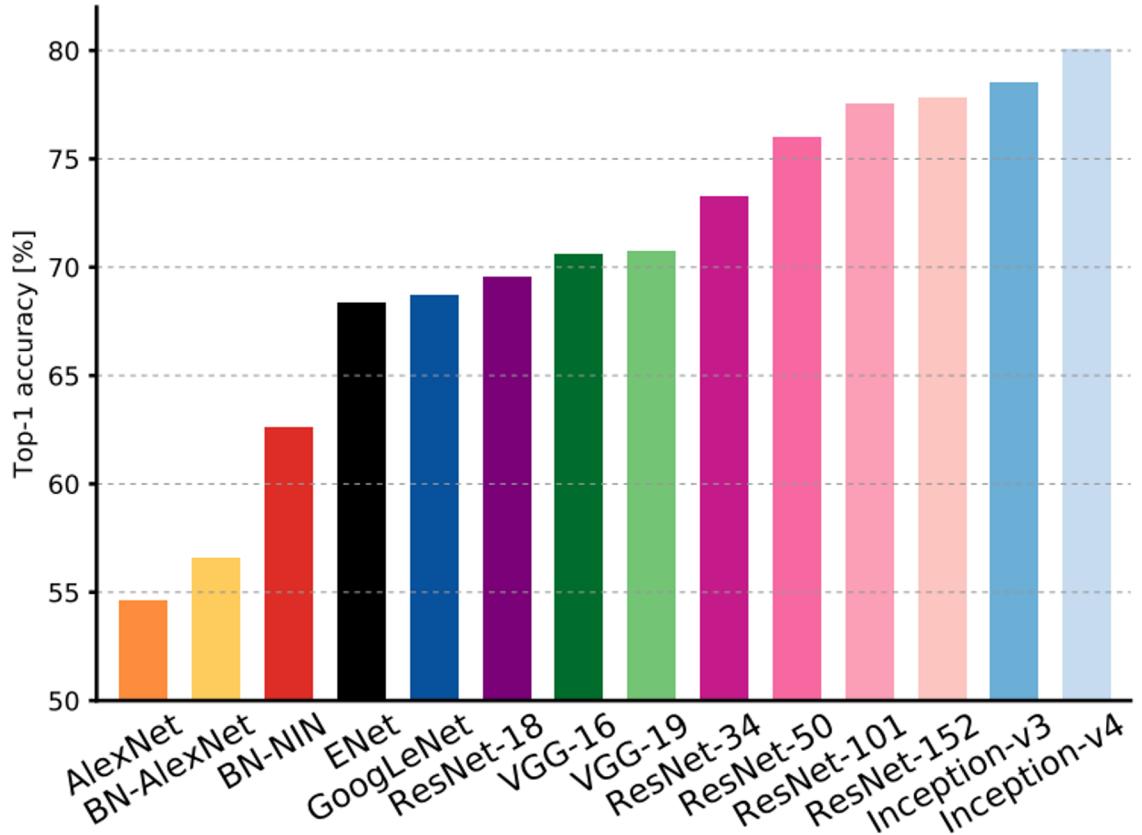


GoogLeNet



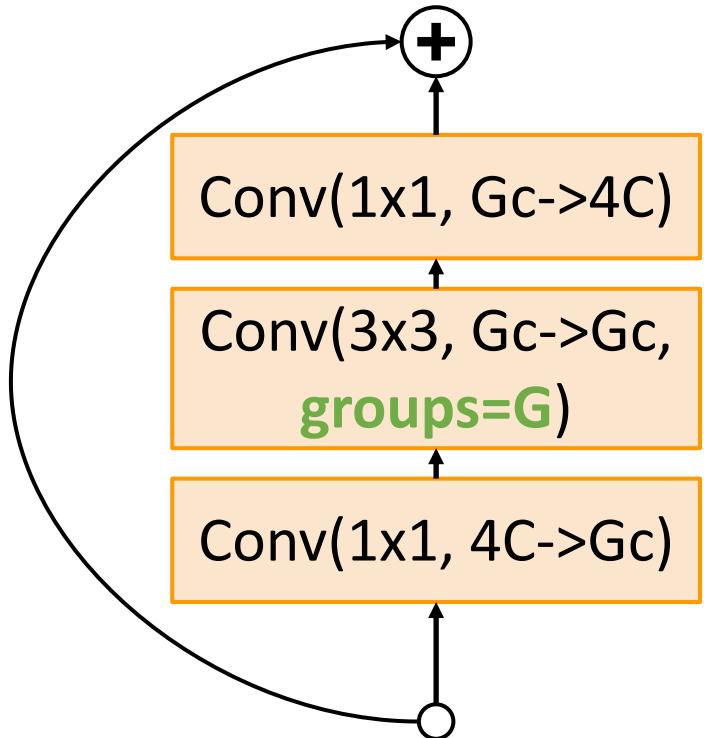
ResNet

CNN Architectures: Efficiency

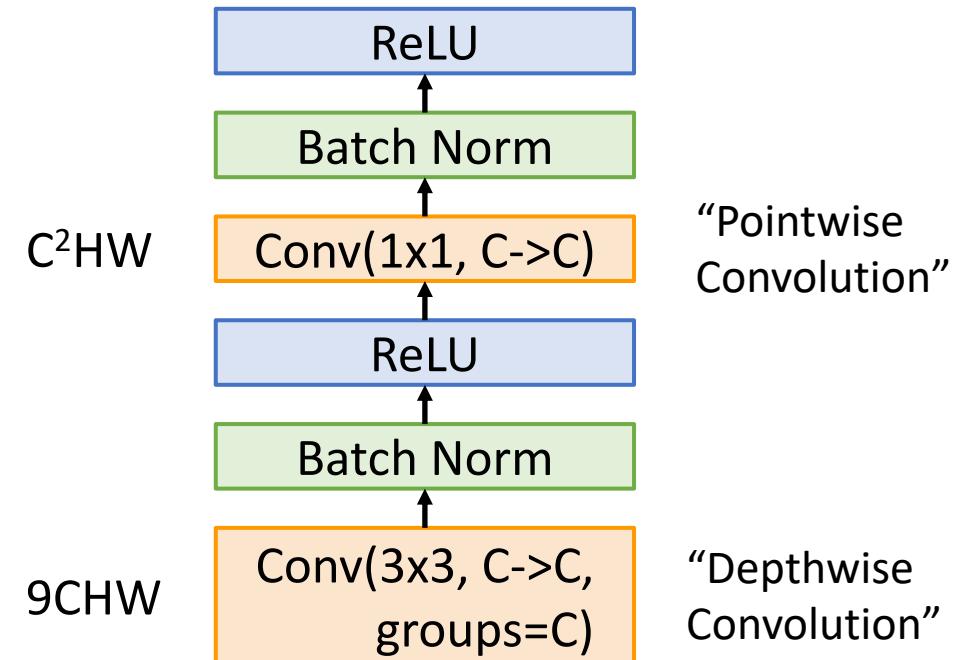


Canziani et al, "An analysis of deep neural network models for practical applications", 2017

CNN Architectures: Efficiency



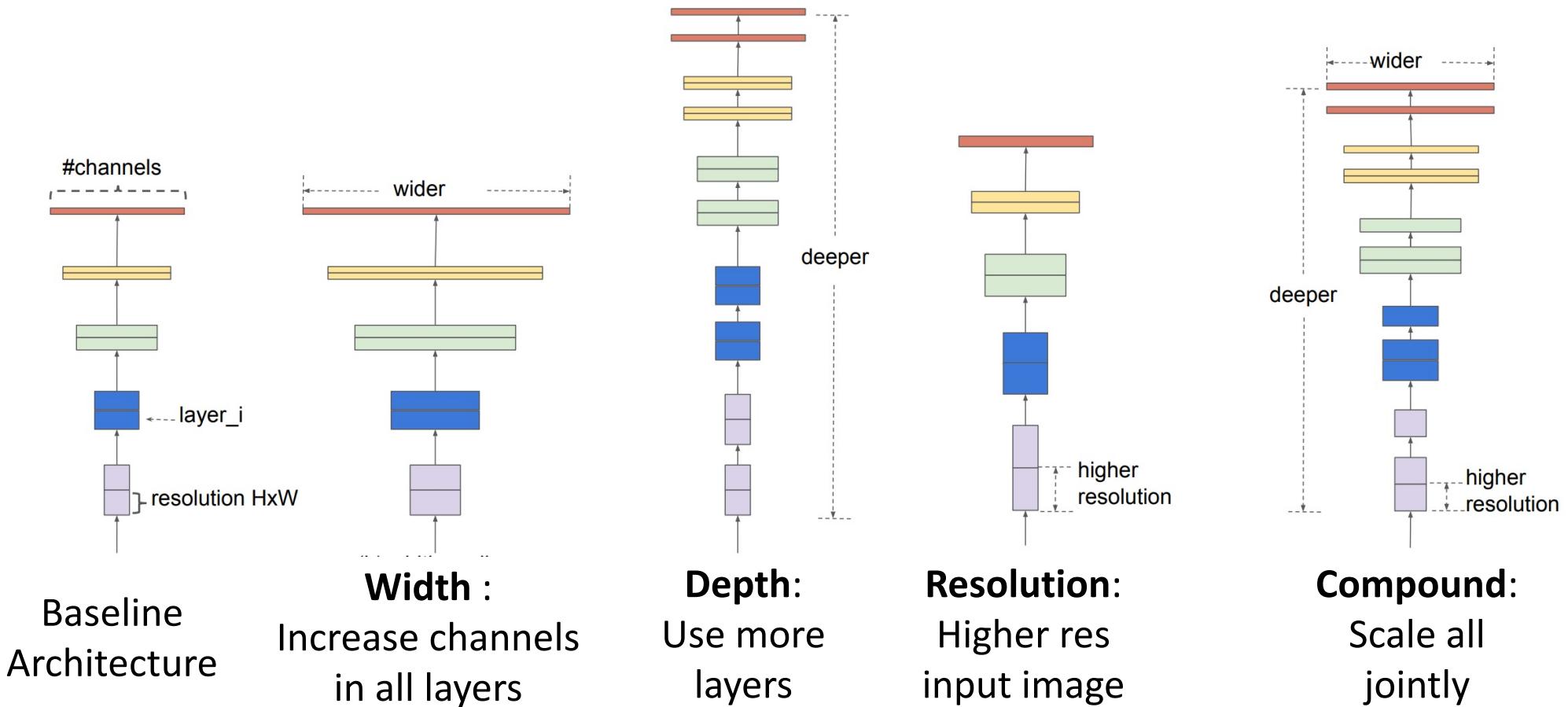
ResNeXt:
Grouped convolution



MobileNets: Depthwise /
Pointwise Convolution

Model Scaling

Starting from a given architecture, how should you **scale it up** to improve performance?

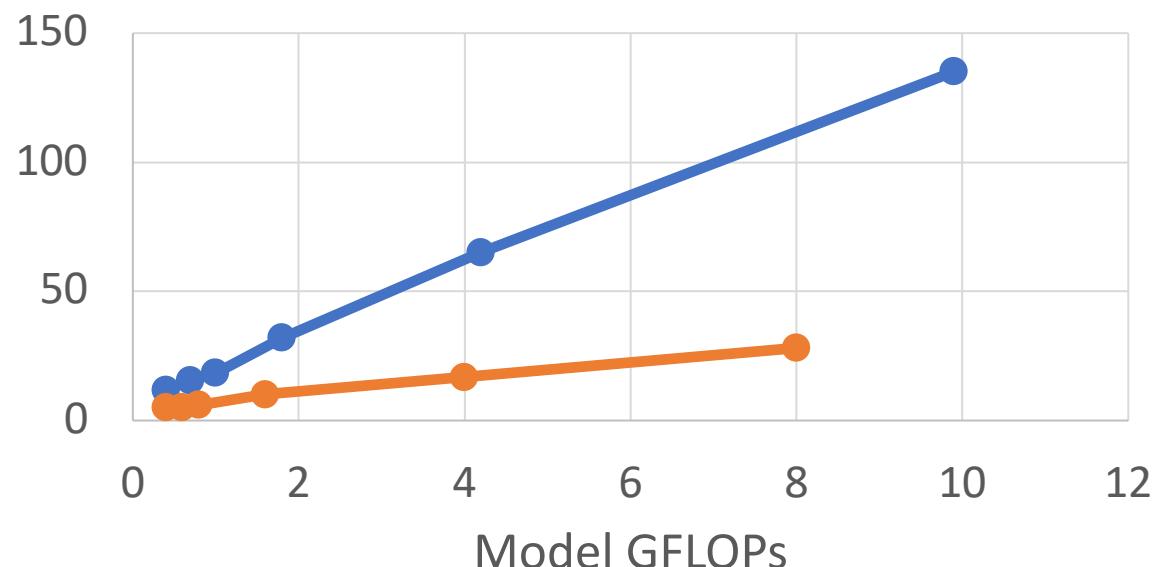


Tan and Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", ICML 2019

Model Scaling: RegNets

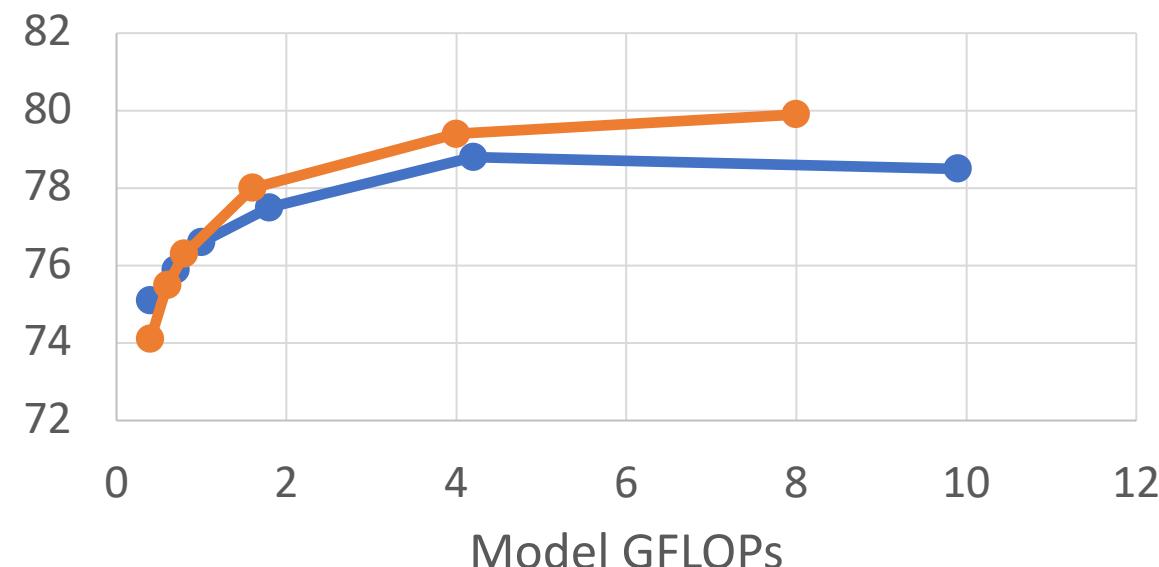
Training Time (hours)

EfficientNet RegNetY



ImageNet Top1 Accuracy

EfficientNet RegNetY



At same FLOPs, RegNet models get similar accuracy as EfficientNets
but are up to 5x faster in training (each iteration is faster)

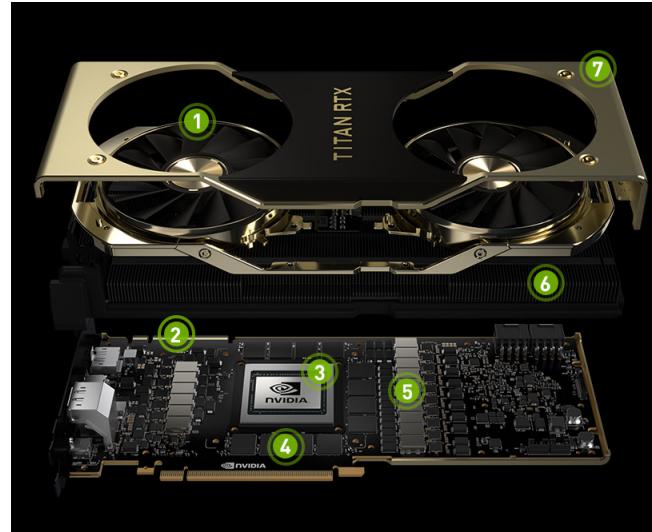
Radosavovic et al, "Designing Network Design Spaces", CVPR 2020
Dollar et al, "Fast and Accurate Model Scaling", CVPR 2021

Deep Learning Hardware and Software

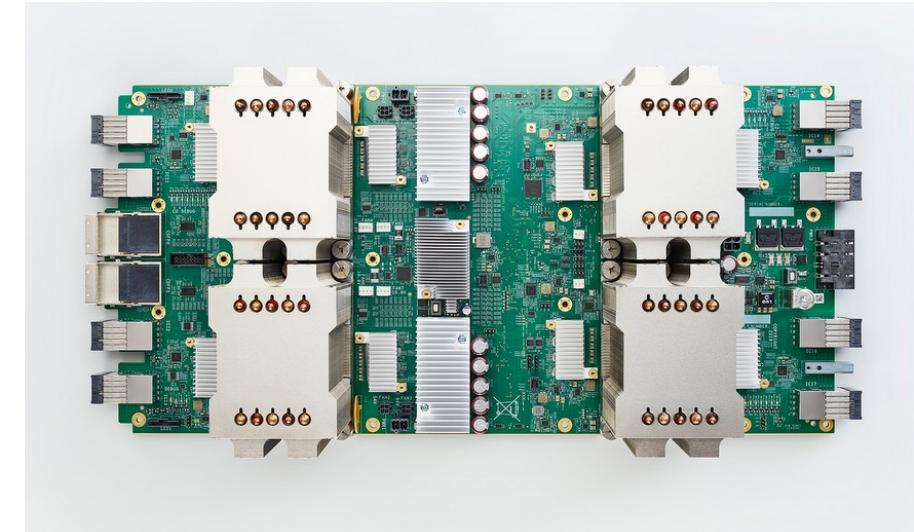
CPU



GPU



TPU



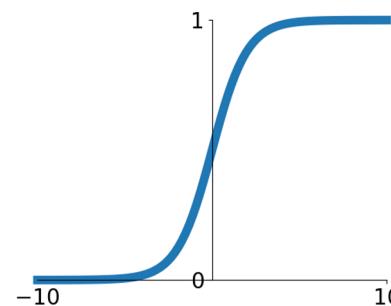
Static Graphs vs Dynamic Graphs

PyTorch vs TensorFlow

Training Networks: Activation Functions

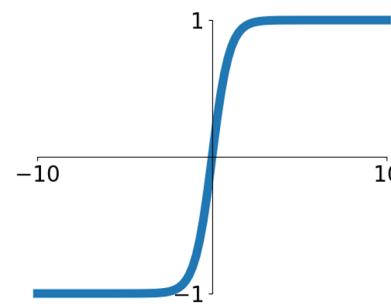
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



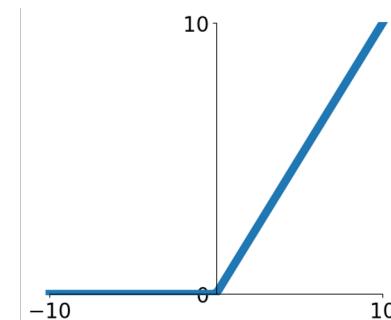
tanh

$$\tanh(x)$$



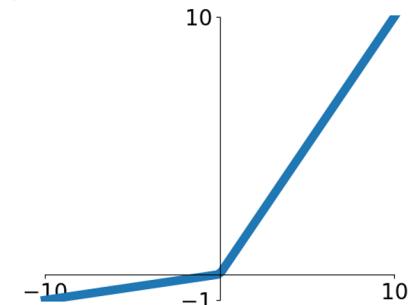
ReLU

$$\max(0, x)$$



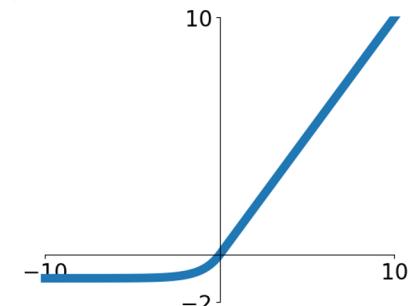
Leaky ReLU

$$\max(0.1x, x)$$



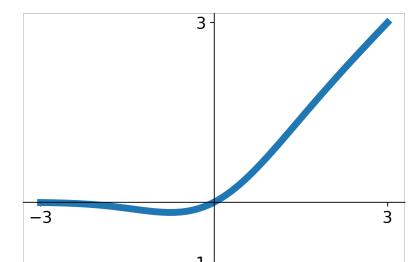
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



GELU

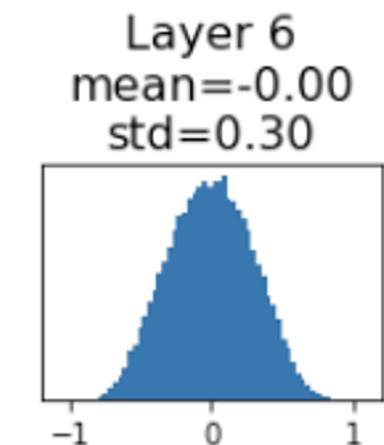
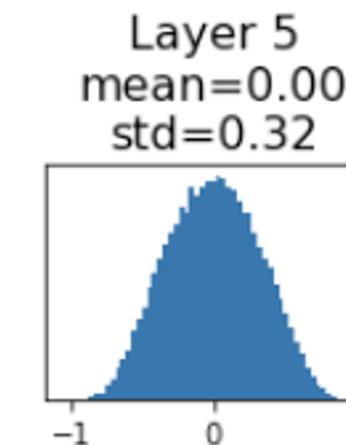
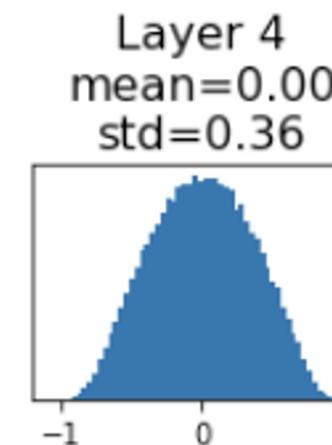
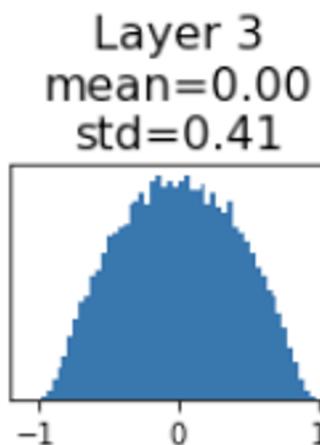
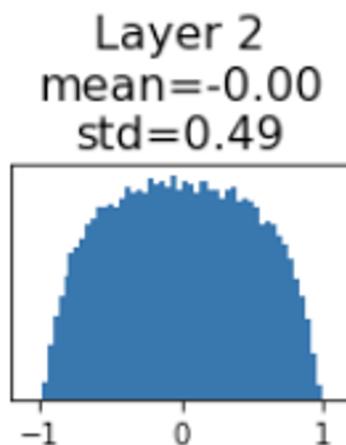
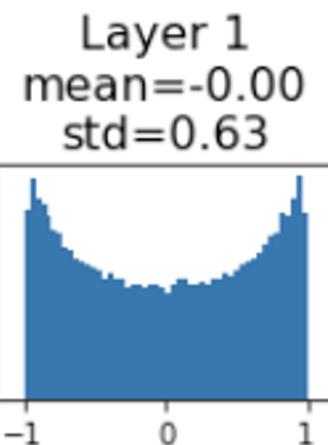
$$\approx x\sigma(1.702x)$$



Training Networks: Weight Initialization

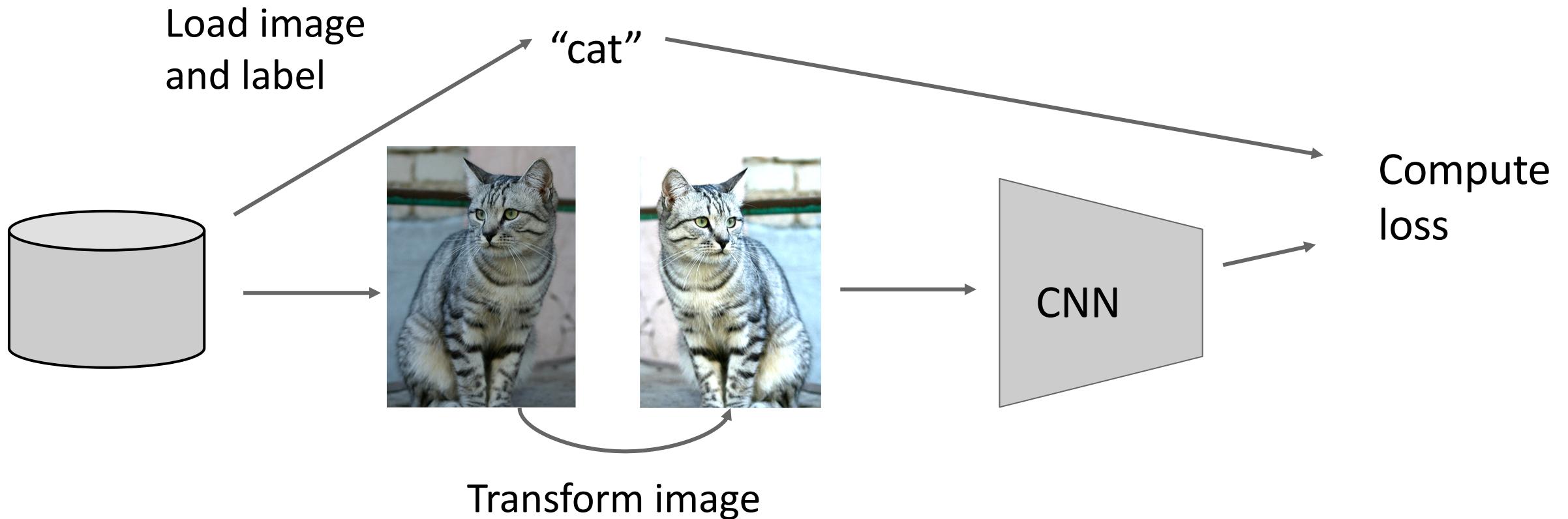
```
dims = [4096] * 7
hs = []
x = np.random.randn(16, dims[0])
for Din, Dout in zip(dims[:-1], dims[1:]):
    W = np.random.randn(Din, Dout) / np.sqrt(Din)
    x = np.tanh(x.dot(W))
    hs.append(x)
```

“Just right”: Activations are nicely scaled for all layers!



Glorot and Bengio, “Understanding the difficulty of training deep feedforward neural networks”, AISTAT 2010

Training Networks: Data Augmentation



Training Networks: Regularization

Training: Add randomness

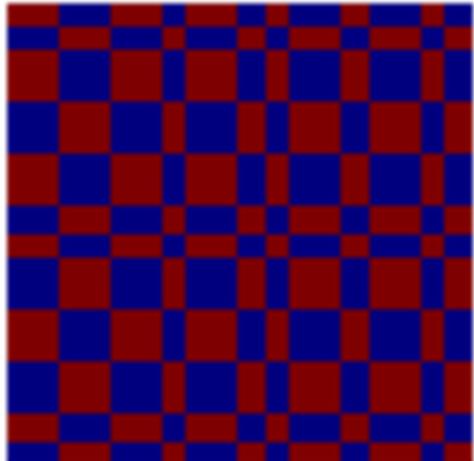
Testing: Marginalize out randomness

Examples:

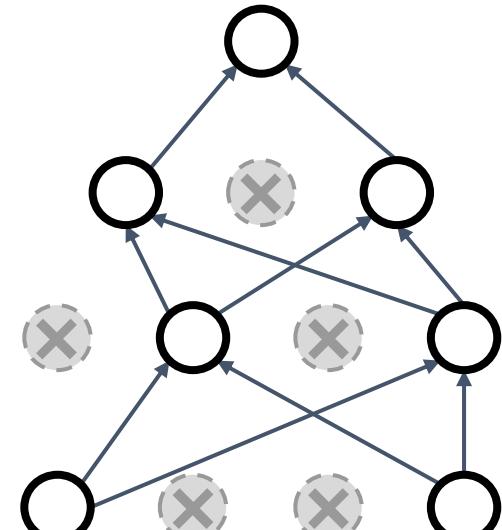
Batch Normalization

Data Augmentation

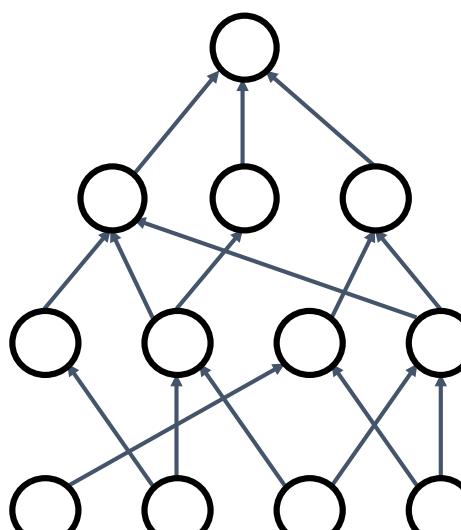
Fractional pooling



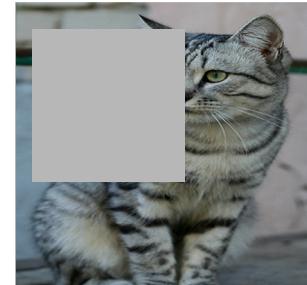
Dropout



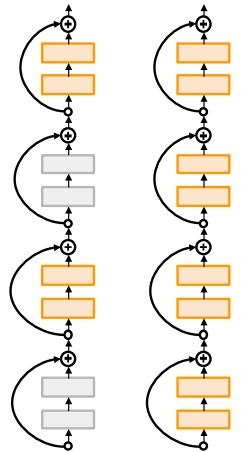
DropConnect



Cutout



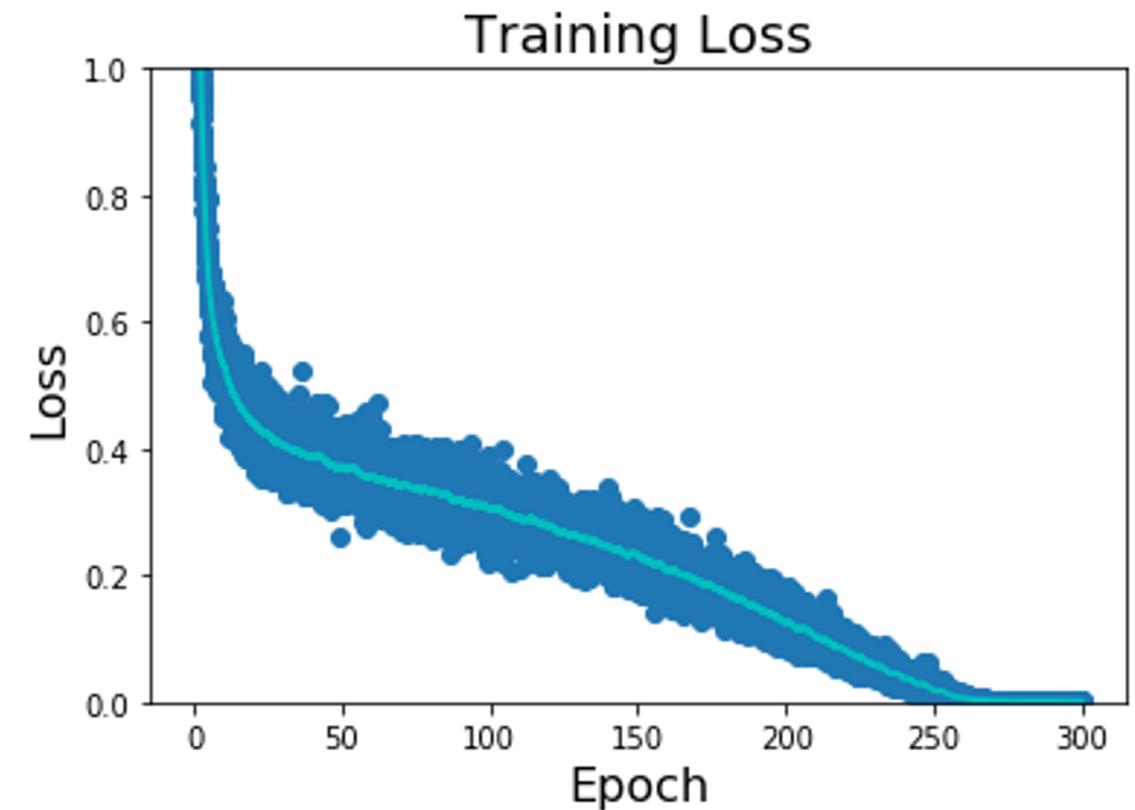
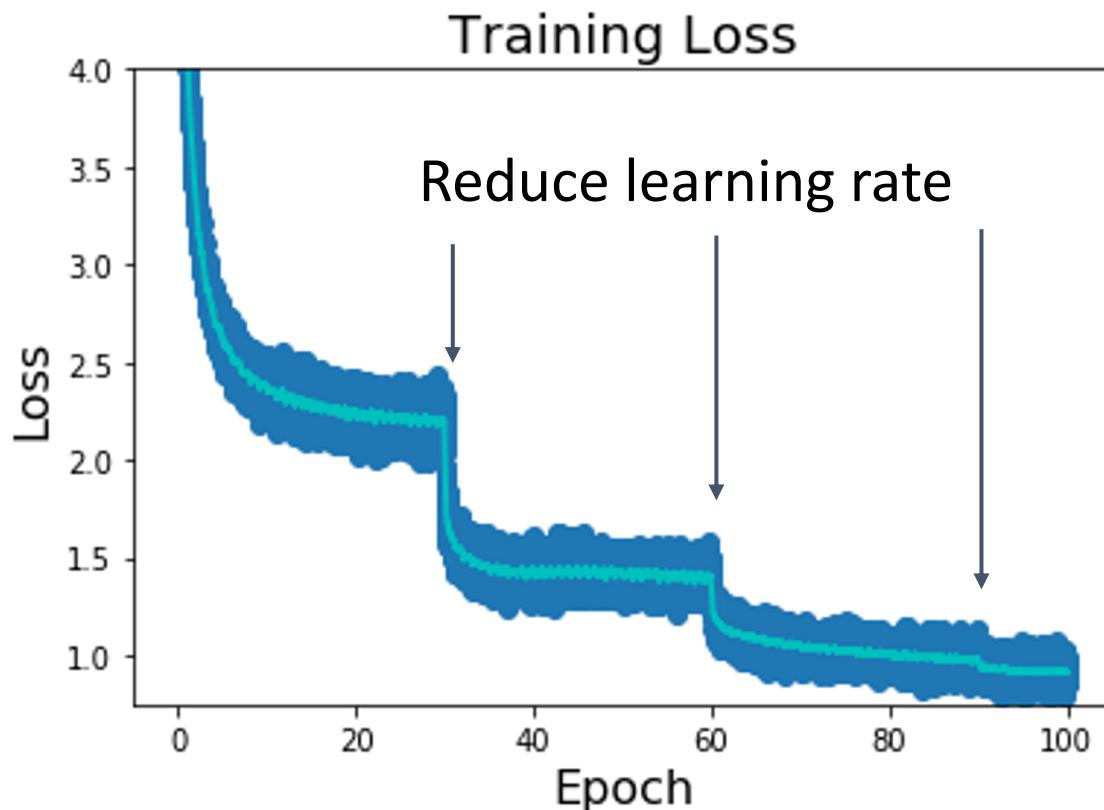
Stochastic Depth



Mixup



Training Neural Networks: Learning Rate Schedules



Computer Vision Tasks

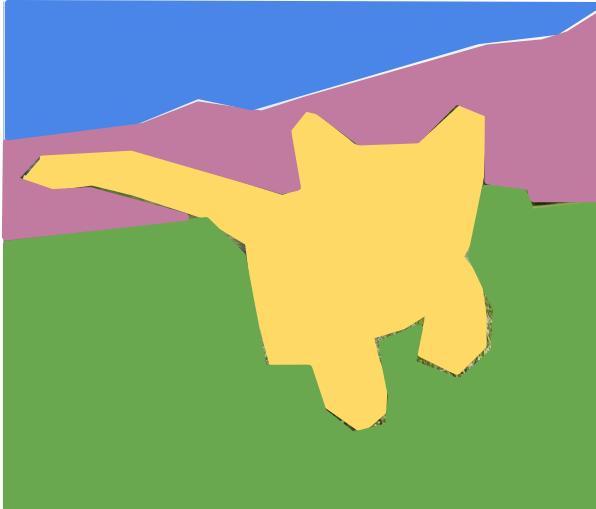
Classification



CAT

No spatial extent

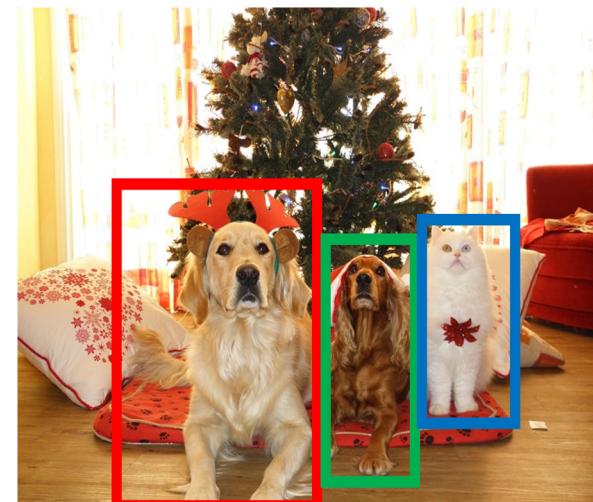
Semantic Segmentation



GRASS, CAT, TREE,
SKY

No objects, just pixels

Object Detection



DOG, DOG, CAT

Multiple Objects

Instance Segmentation

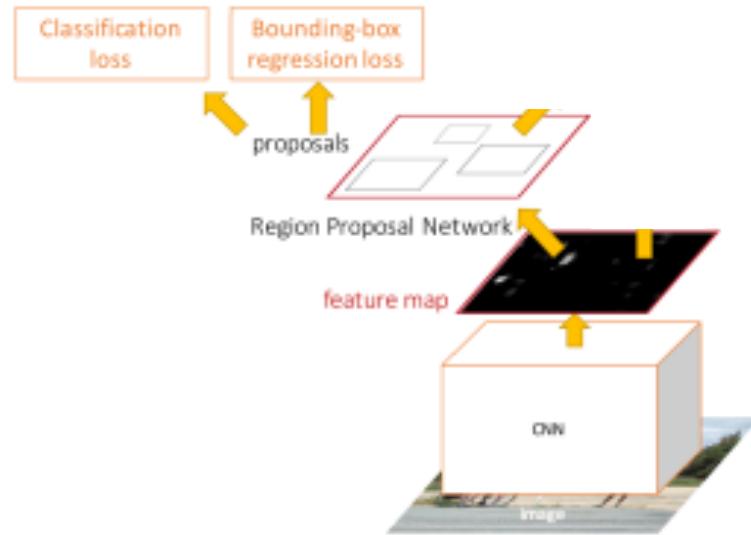


DOG, DOG, CAT

Object Detection: Single Stage vs Two Stage

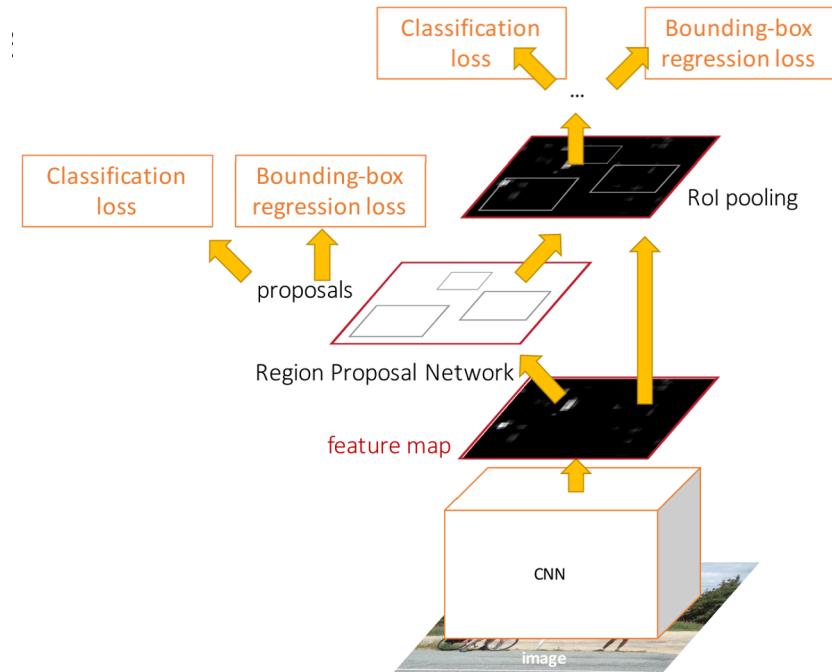
Single-Stage:

FCOS, YOLO, RetinaNet
Make all predictions
with a CNN



Two-Stage:

Faster R-CNN
Use RPN to predict proposals,
classify them with second stage



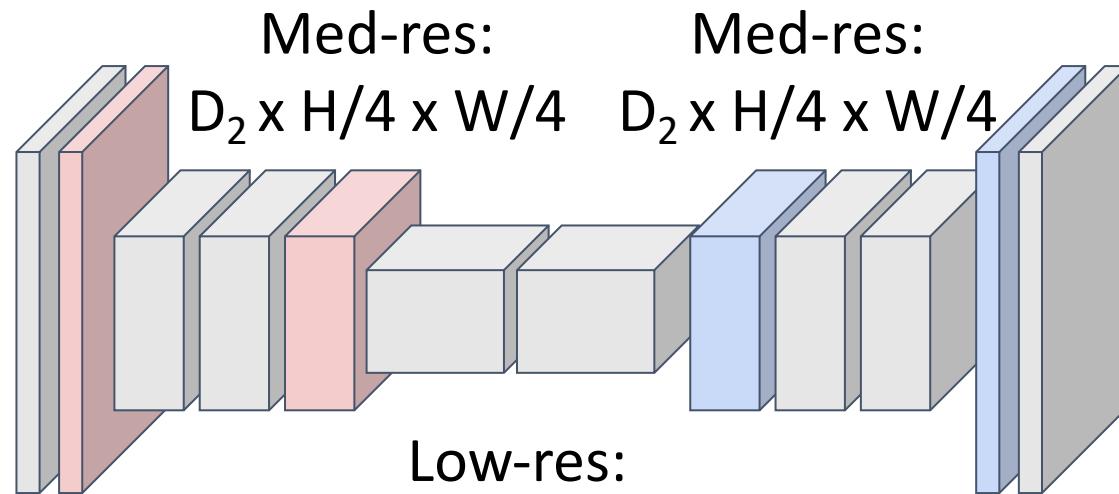
Semantic Segmentation: Fully Convolutional Network

Downsampling:
Pooling, strided
convolution



Input:
 $3 \times H \times W$

High-res:
 $D_1 \times H/2 \times W/2$



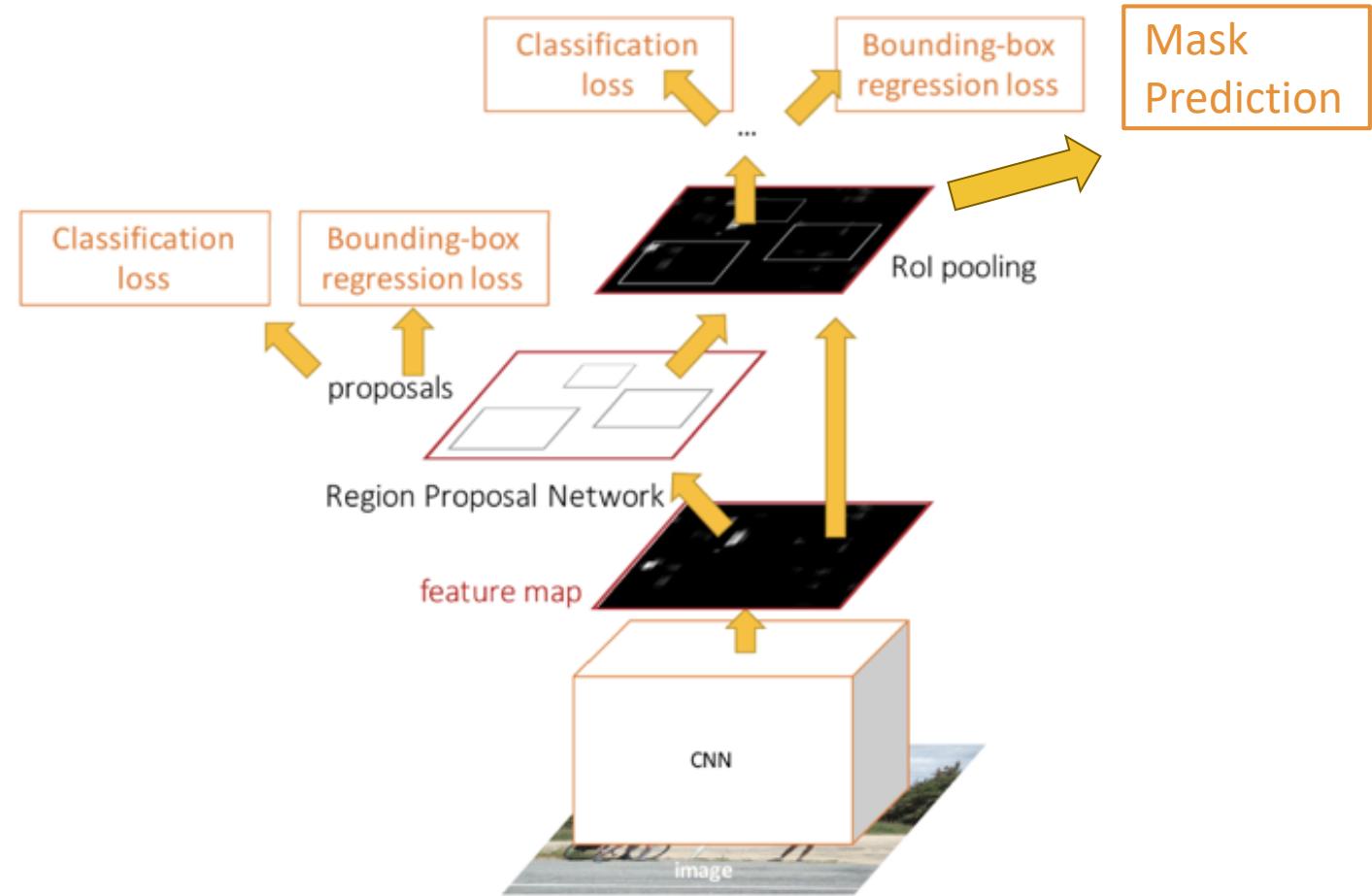
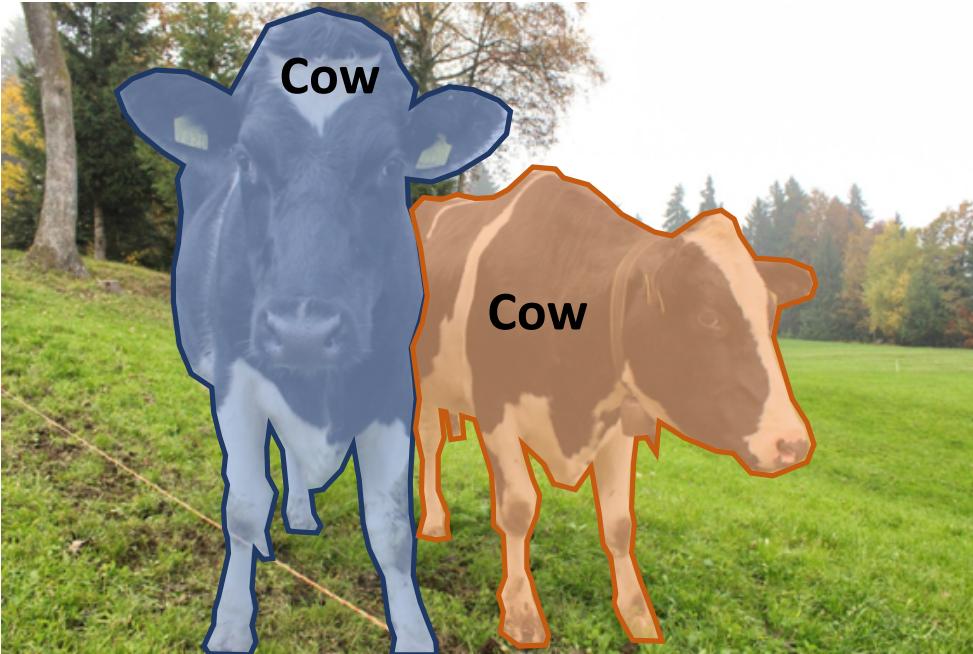
Design network as a bunch of convolutional layers, with
downsampling and **upsampling** inside the network!

Upsampling:
interpolation,
transposed conv



Loss function: Per-Pixel cross-entropy

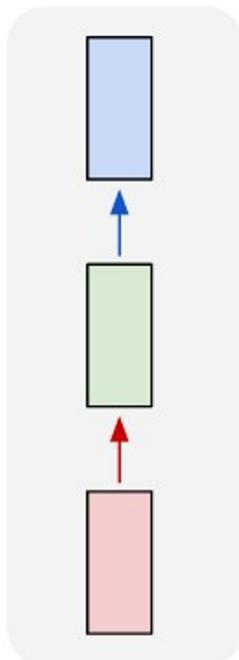
Instance Segmentation: Detection + Segmentation



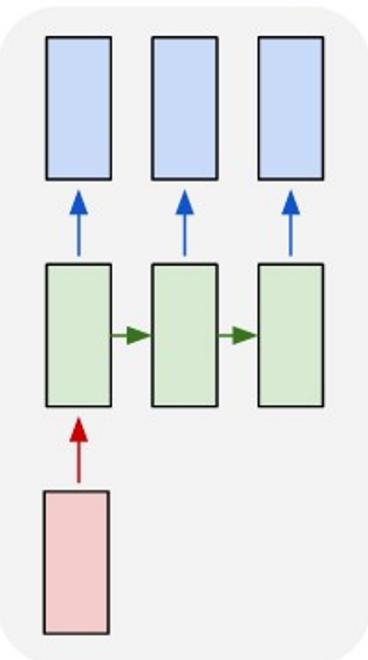
He et al, "Mask R-CNN", ICCV 2017

Recurrent Neural Networks: Process Sequences

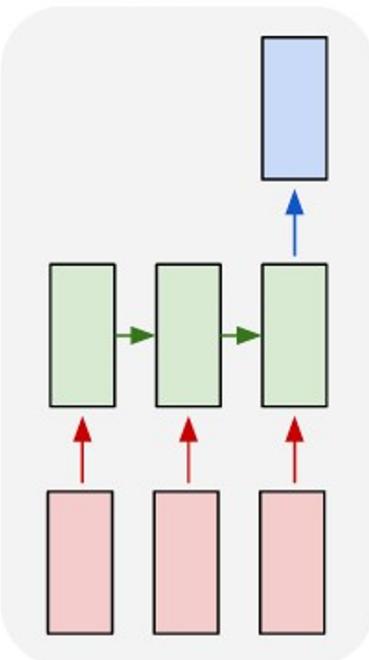
one to one



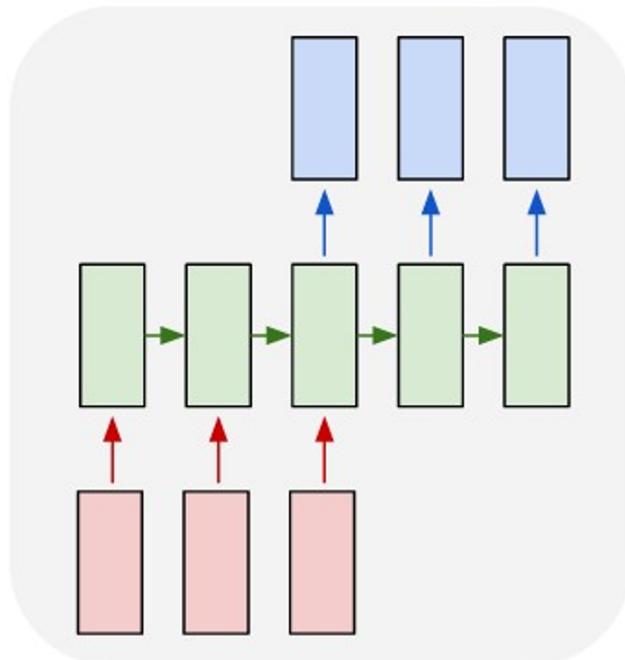
one to many



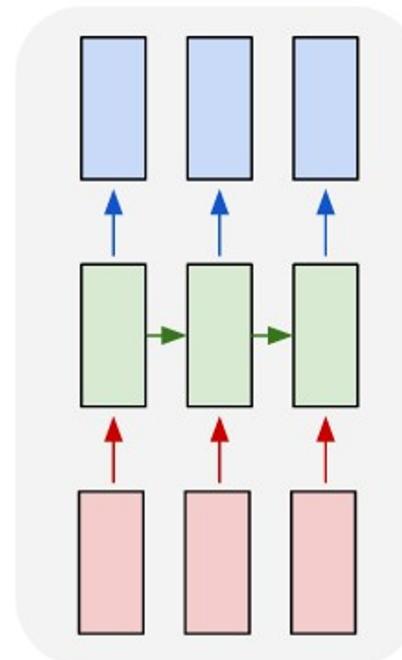
many to one



many to many

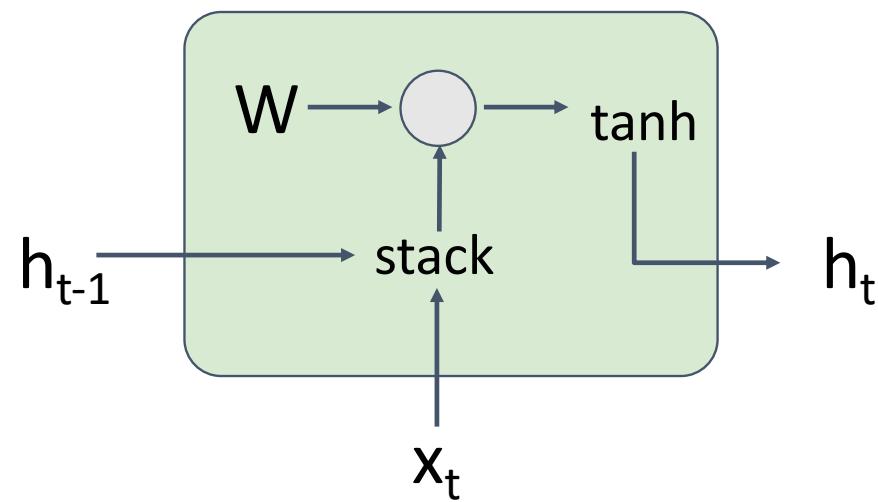


many to many

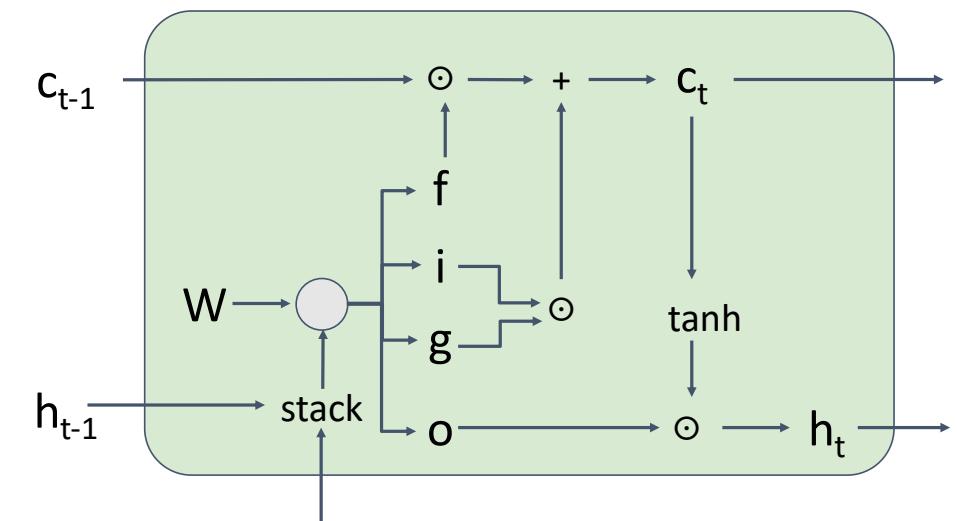


Recurrent Neural Networks: Architectures

Vanilla Recurrent Network

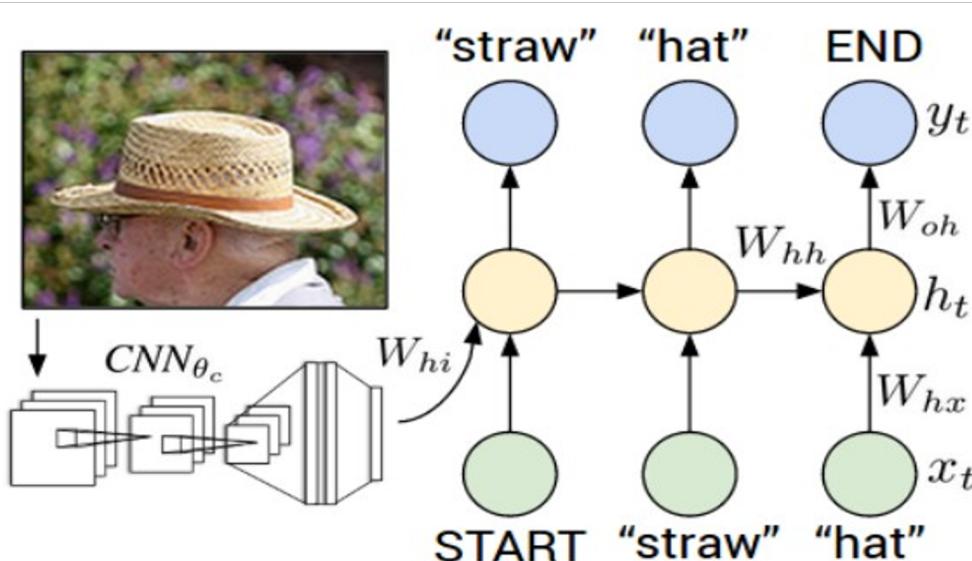


Long Short Term Memory (LSTM)



Recurrent Neural Networks: Captioning

Captions generated using [neuraltalk2](#)
All images are [CC0 Public domain](#): [cat](#),
[suitcase](#), [cat tree](#), [dog](#), [bear](#), [surfers](#),
[tennis](#), [giraffe](#), [motorcycle](#)



A white teddy bear sitting in the grass



A dog is running in the grass with a frisbee



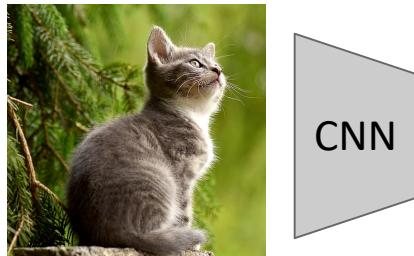
Two giraffes standing in a grassy field



A man riding a dirt bike on a dirt track

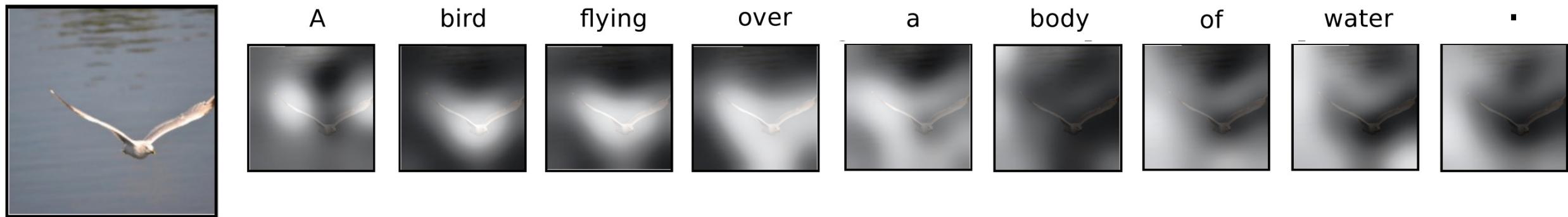
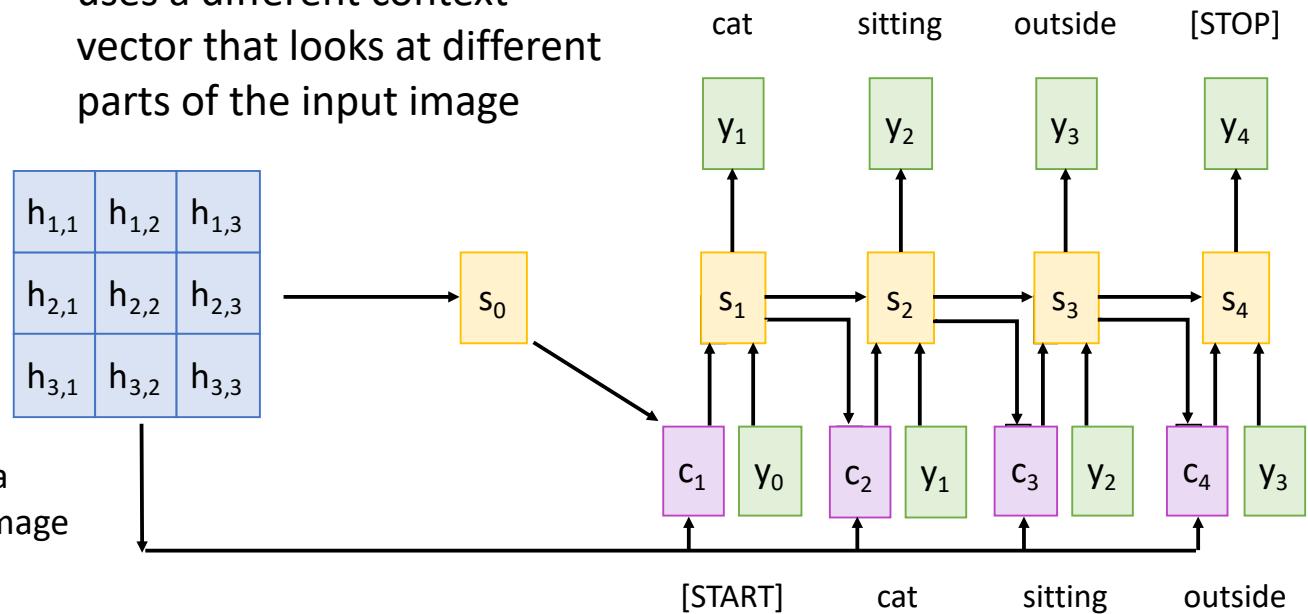
Attention

$$\begin{aligned} e_{t,i,j} &= f_{att}(s_{t-1}, h_{i,j}) \\ a_{t,:,:} &= \text{softmax}(e_{t,:,:}) \\ c_t &= \sum_{i,j} a_{t,i,j} h_{i,j} \end{aligned}$$



Use a CNN to compute a grid of features for an image

Each timestep of decoder uses a different context vector that looks at different parts of the input image



Self-Attention Layer

One **query** per **input vector**

Inputs:

Input vectors: X (Shape: $N_x \times D_x$)

Key matrix: W_K (Shape: $D_x \times D_Q$)

Value matrix: W_V (Shape: $D_x \times D_V$)

Query matrix: W_Q (Shape: $D_x \times D_Q$)

Computation:

Query vectors: $Q = XW_Q$

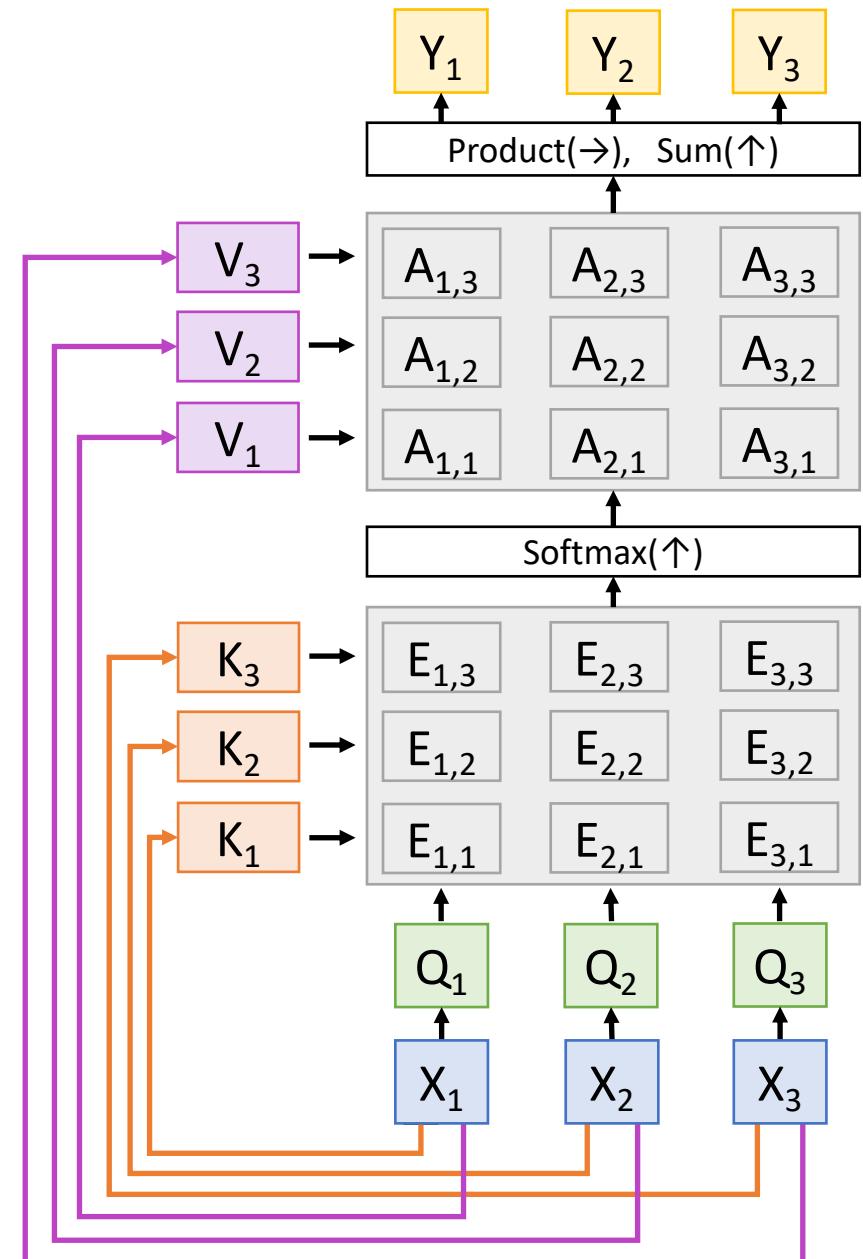
Key vectors: $K = XW_K$ (Shape: $N_x \times D_Q$)

Value Vectors: $V = XW_V$ (Shape: $N_x \times D_V$)

Similarities: $E = QK^T$ (Shape: $N_x \times N_x$) $E_{i,j} = Q_i \cdot K_j / \text{sqrt}(D_Q)$

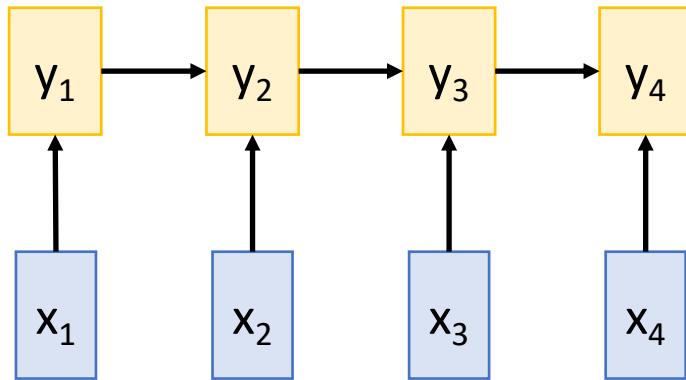
Attention weights: $A = \text{softmax}(E, \text{dim}=1)$ (Shape: $N_x \times N_x$)

Output vectors: $Y = AV$ (Shape: $N_x \times D_V$) $Y_i = \sum_j A_{i,j} V_j$



Processing Sequences

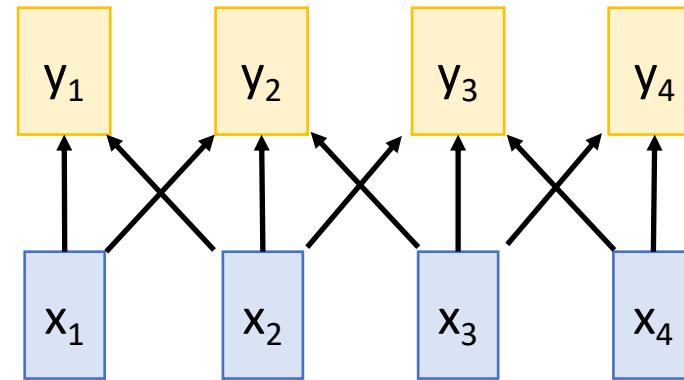
Recurrent Neural Network



Works on **Ordered Sequences**

- (+) Good at long sequences:
After one RNN layer, h_T "sees"
the whole sequence
- (-) Not parallelizable: need to
compute hidden states
sequentially

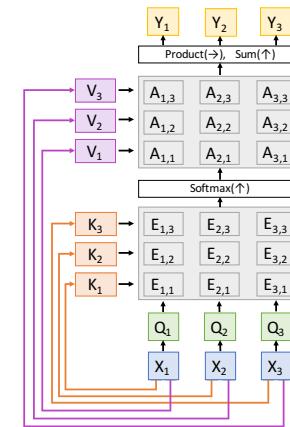
1D Convolution



Works on **Multidimensional Grids**

- (-) Bad at long sequences: Need
to stack many conv layers for
outputs to "see" the whole
sequence
- (+) Highly parallel: Each output
can be computed in parallel

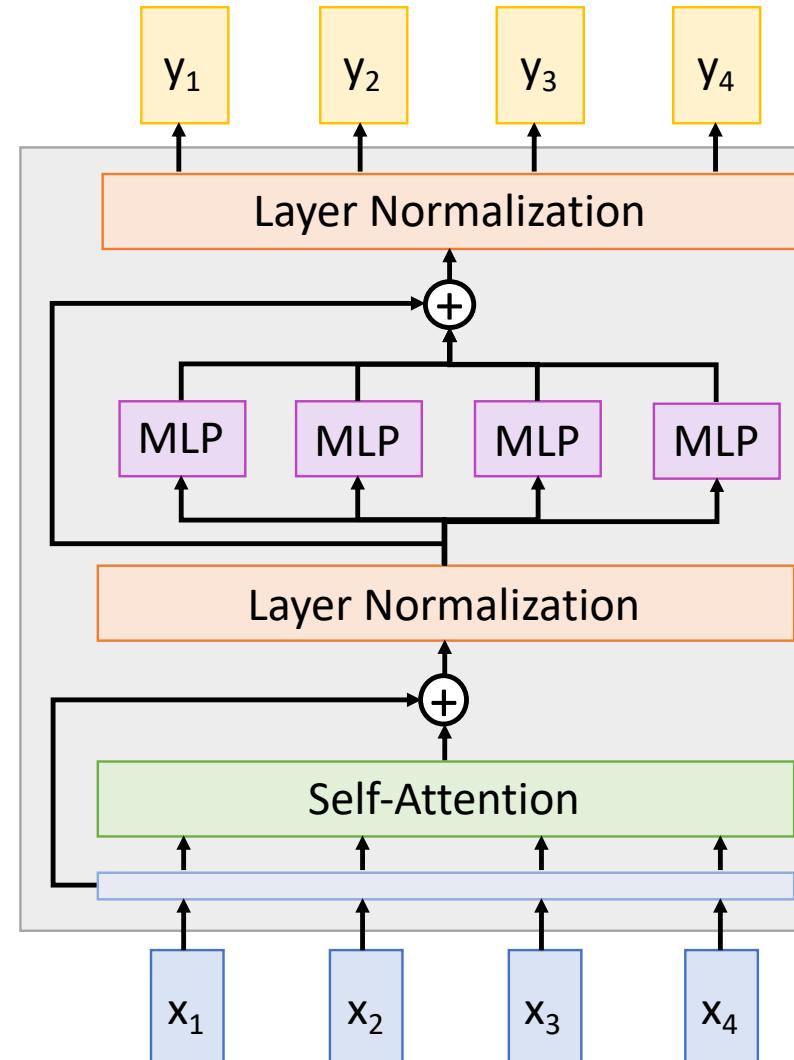
Self-Attention



Works on **Sets of Vectors**

- (-) Good at long sequences: after
one self-attention layer, each
output "sees" all inputs!
- (+) Highly parallel: Each output
can be computed in parallel
- (-) Very memory intensive

Attention is all you need: The Transformer



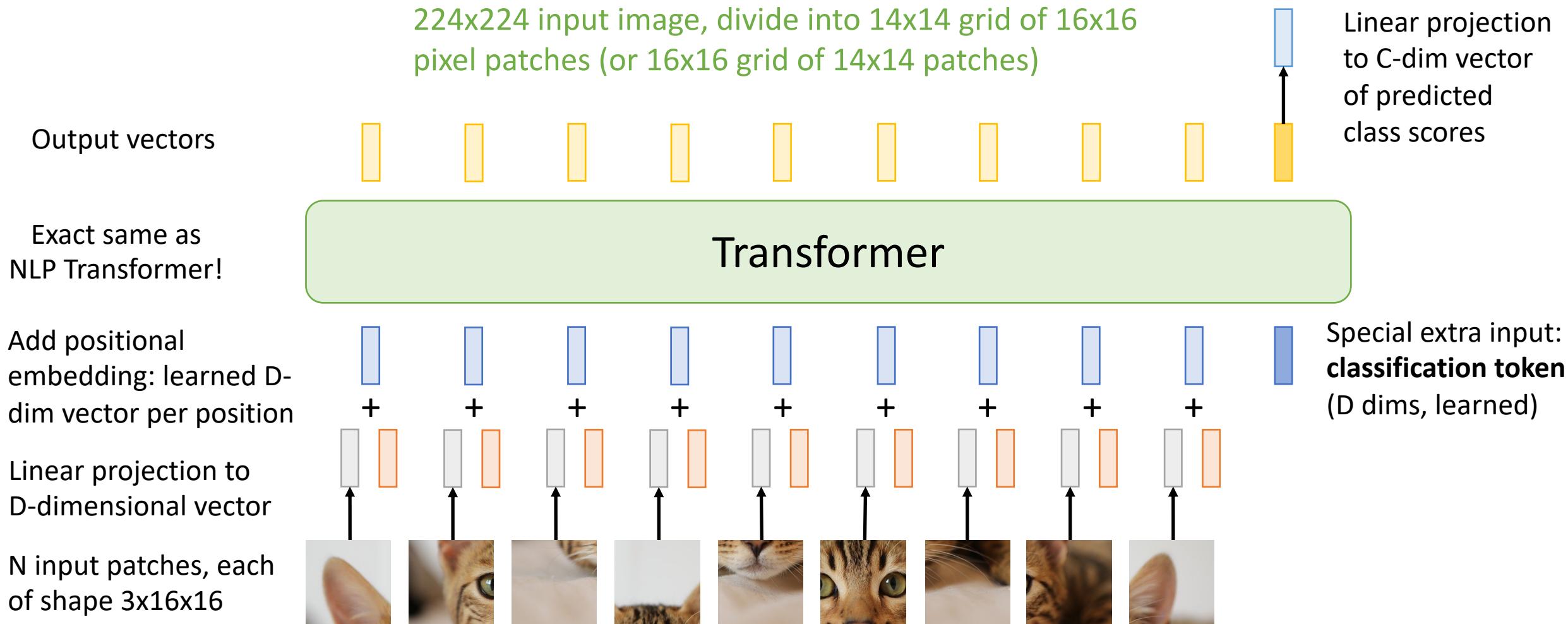
Vaswani et al, "Attention is all you need", NeurIPS 2017

Scaling up Transformers

Model	Layers	Width	Heads	Params	Data	Training
Transformer-Base	12	512	8	65M		8x P100 (12 hours)
Transformer-Large	12	1024	16	213M		8x P100 (3.5 days)
BERT-Base	12	768	12	110M	13 GB	
BERT-Large	24	1024	16	340M	13 GB	
XLNet-Large	24	1024	16	~340M	126 GB	512x TPU-v3 (2.5 days)
RoBERTa	24	1024	16	355M	160 GB	1024x V100 GPU (1 day)
GPT-2	48	1600	?	1.5B	40 GB	
Megatron-LM	72	3072	32	8.3B	174 GB	512x V100 GPU (9 days)
Turing-NLG	78	4256	28	17B	?	256x V100 GPU
GPT-3	96	12,288	96	175B	694GB	?
Gopher	80	16,384	128	280B	10.55 TB	4096x TPUv3 (38 days)
PaLM	118	18,432	48	540B		6144x TPUv4

Chowdhery et al, "PaLM: Scaling Language Modeling with Pathways", arXiv 2022

Vision Transformer (ViT)



Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021

[Cat image](#) is free for commercial use under a [Pixabay license](#)

Generative Models

Autoregressive Models directly maximize likelihood of training data:

$$p_{\theta}(x) = \prod_{i=1}^N p_{\theta}(x_i|x_1, \dots, x_{i-1})$$

Good image quality, can evaluate with perplexity. Slow to generate data, needs tricks to scale up.

Variational Autoencoders introduce a latent z , and maximize a lower bound:

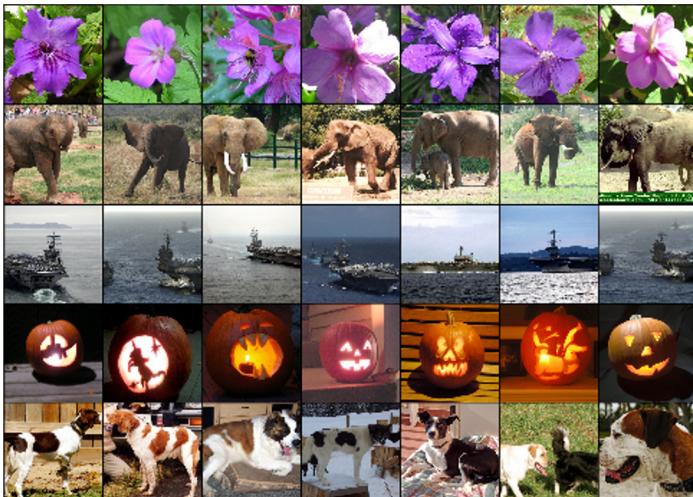
$$p_{\theta}(x) = \int_Z p_{\theta}(x|z)p(z)dz \geq E_{z \sim q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}\left(q_{\phi}(z|x), p(z)\right)$$

Latent z allows for powerful interpolation and editing applications.

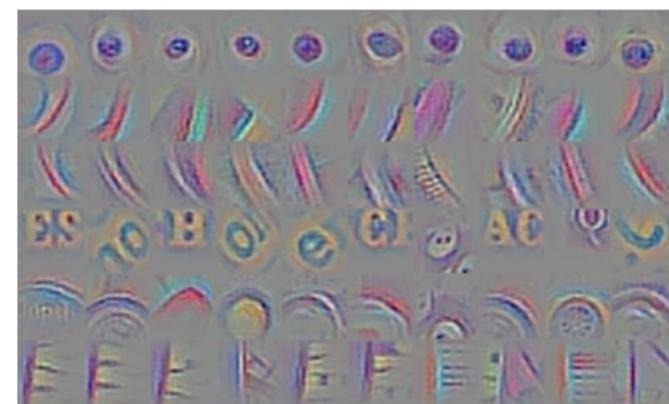
Generative Adversarial Networks give up on modeling $p(x)$, but allow us to draw samples from $p(x)$. Difficult to evaluate, but best qualitative results today

Visualizing and Understanding CNNs

Nearest Neighbor

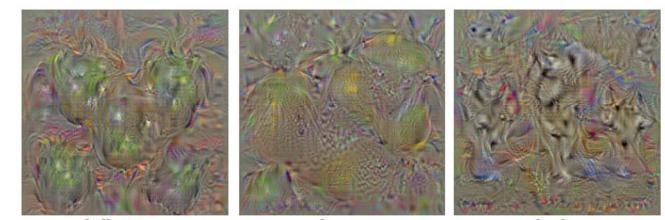


Maximally Activating Patches



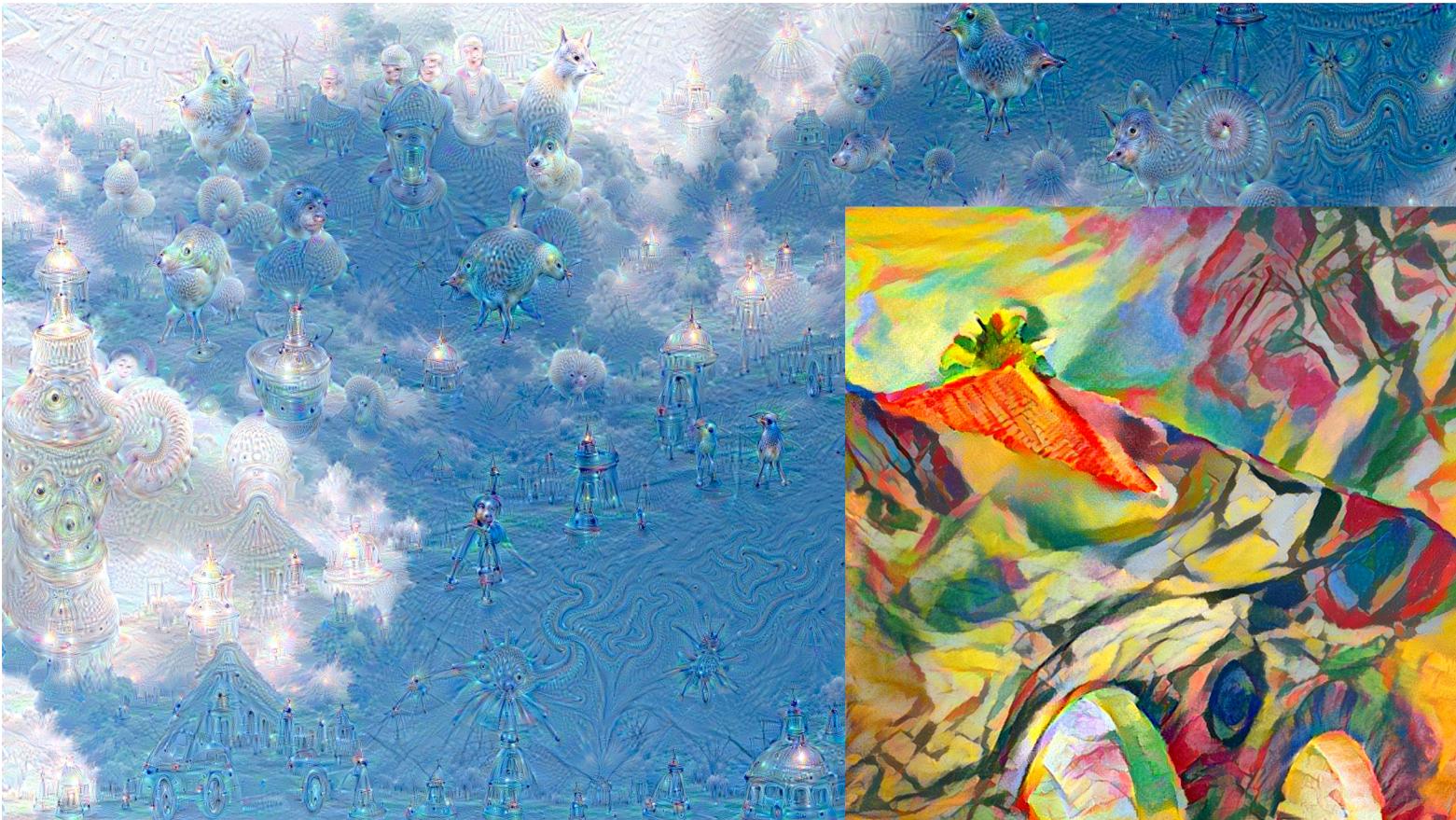
(Guided) Backprop

Synthetic Images via Gradient Ascent

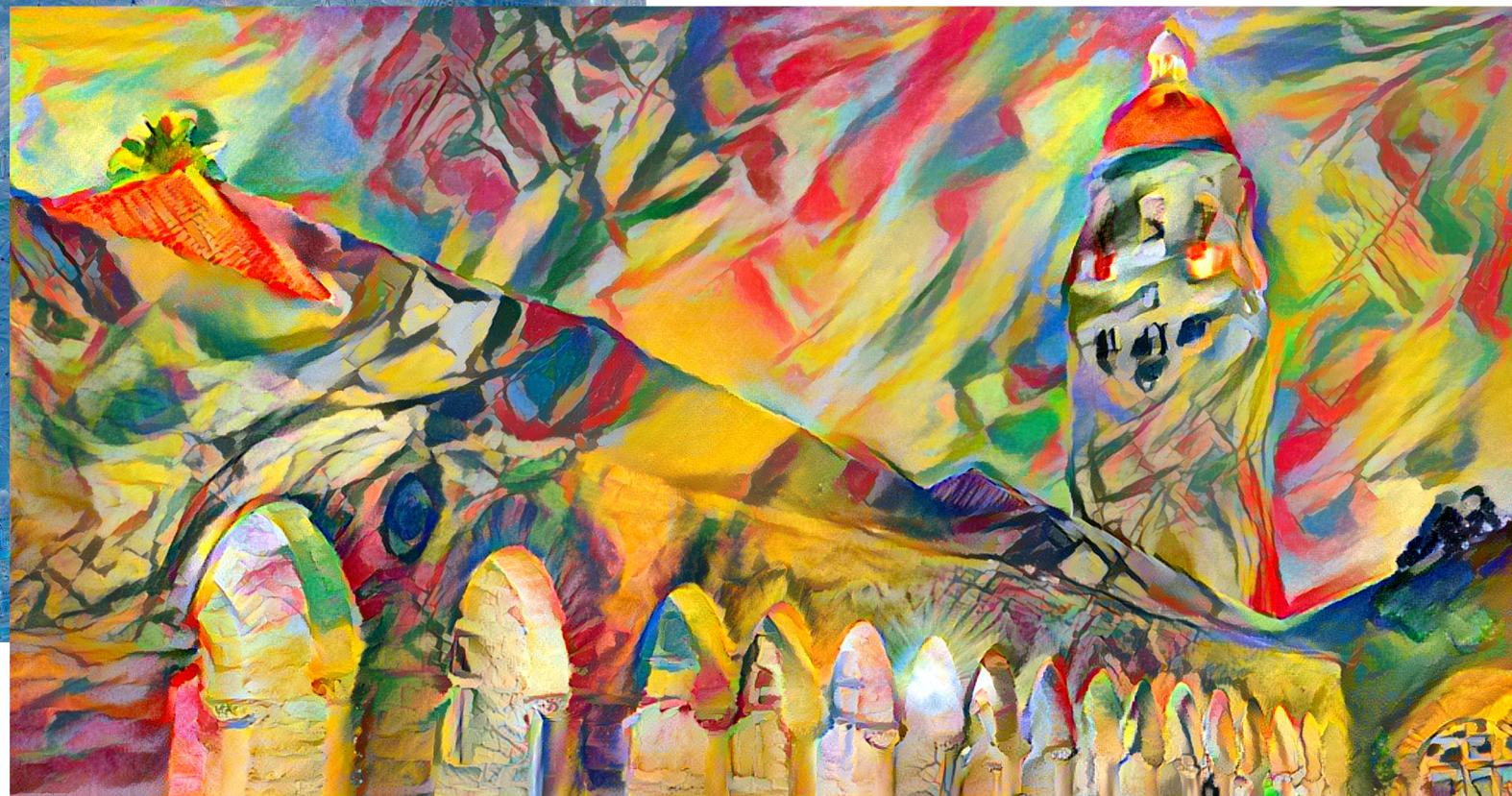


Feature Inversion

Making Art with CNNs



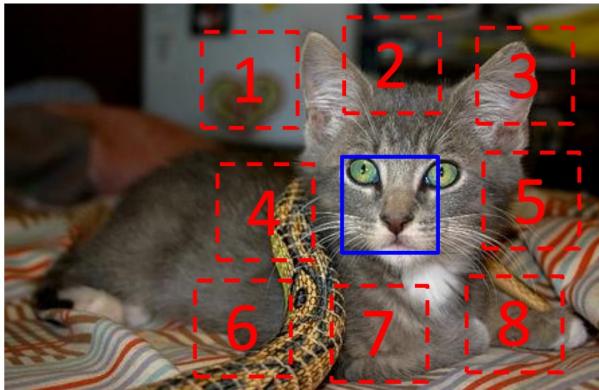
DeepDream



Style Transfer

Self-Supervised Learning

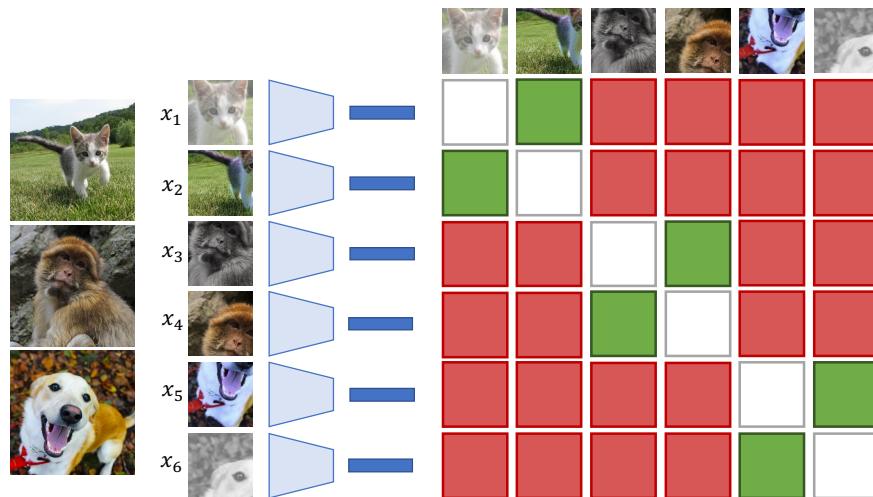
Context Prediction



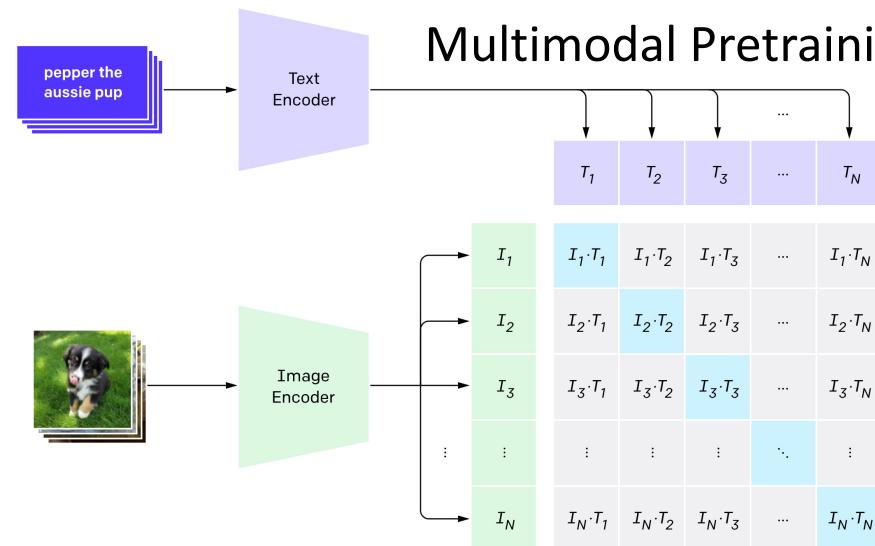
Colorization



Contrastive Learning

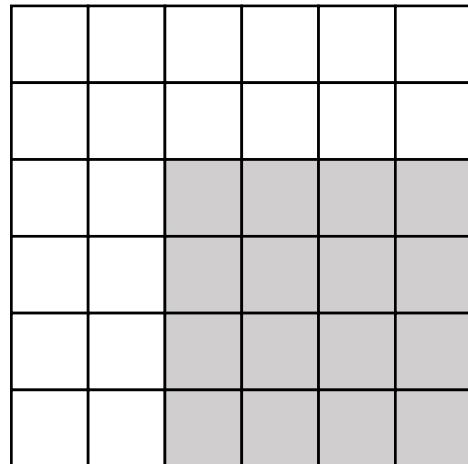


Multimodal Pretraining



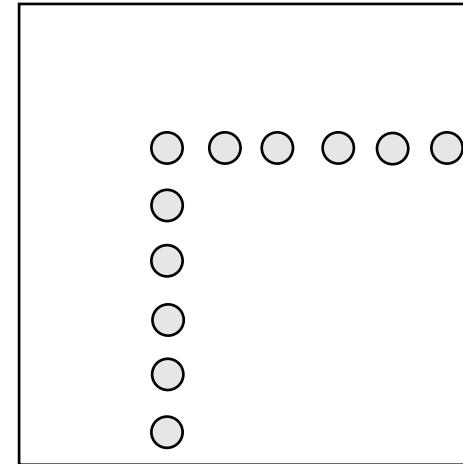
Adding a Dimension: 3D Shapes

8
8
2
2
2
2

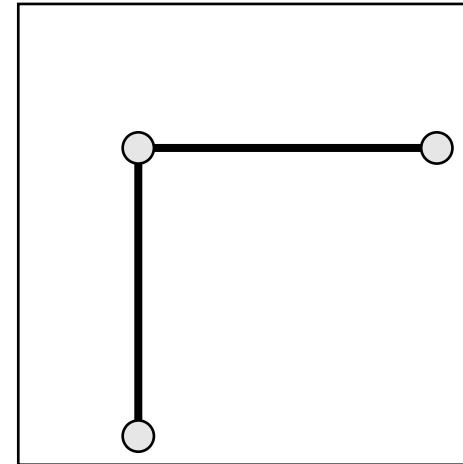


Depth
Map

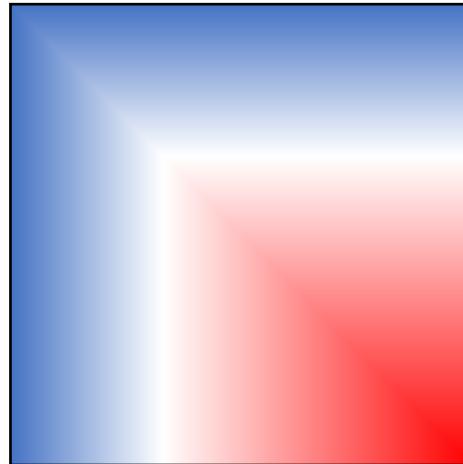
Voxel
Grid



Pointcloud



Mesh



Implicit
Surface

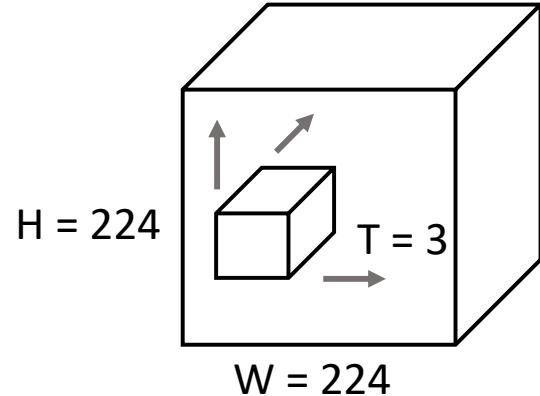
Adding a Dimension: NeRF



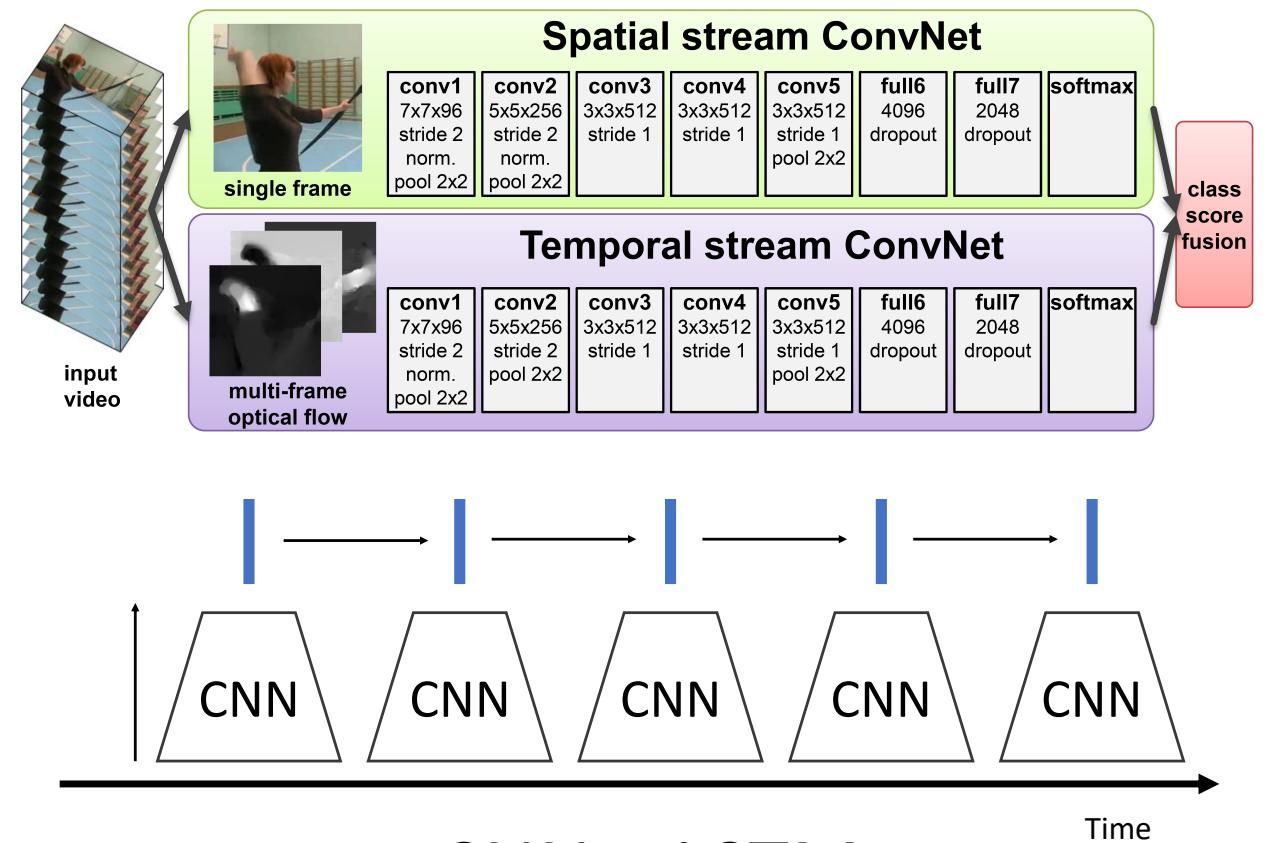
Mildenhall et al, "Representing Scenes as Neural Radiance Fields for View Synthesis", ECCV 2020

Adding a Dimension: Deep Learning on Video

3D CNNs



Two Stream Networks



Self-Attention

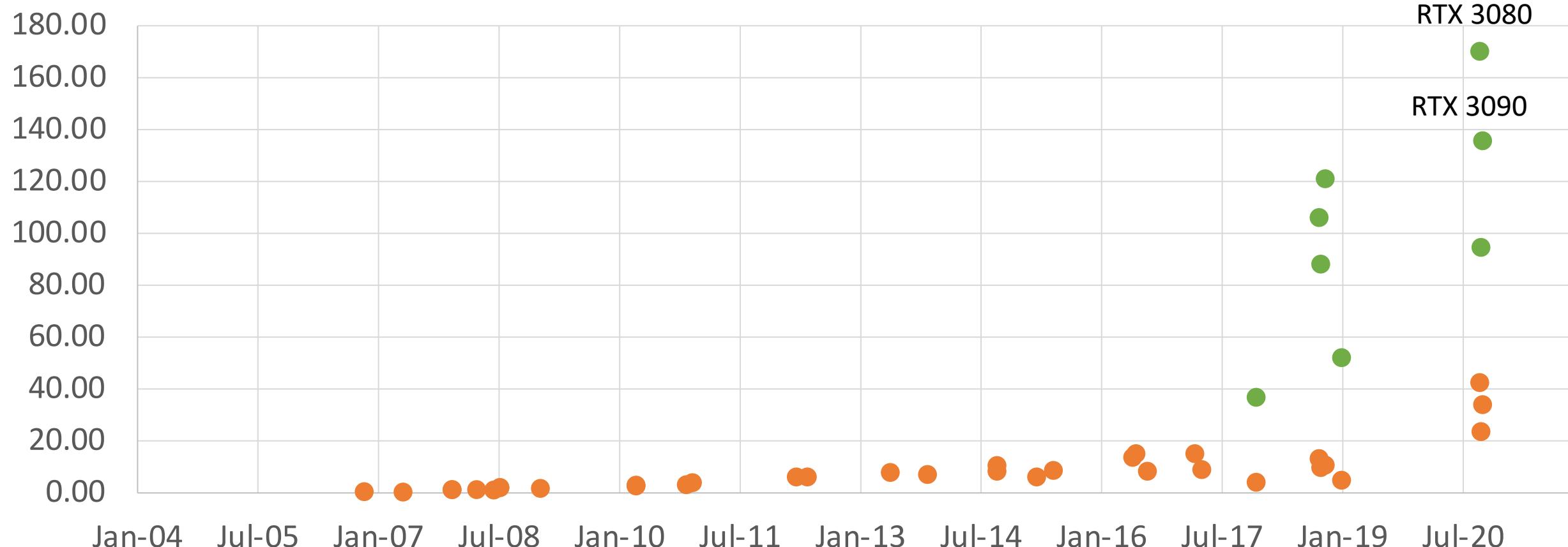
What's Next?

Bigger Models, More Data, More Compute

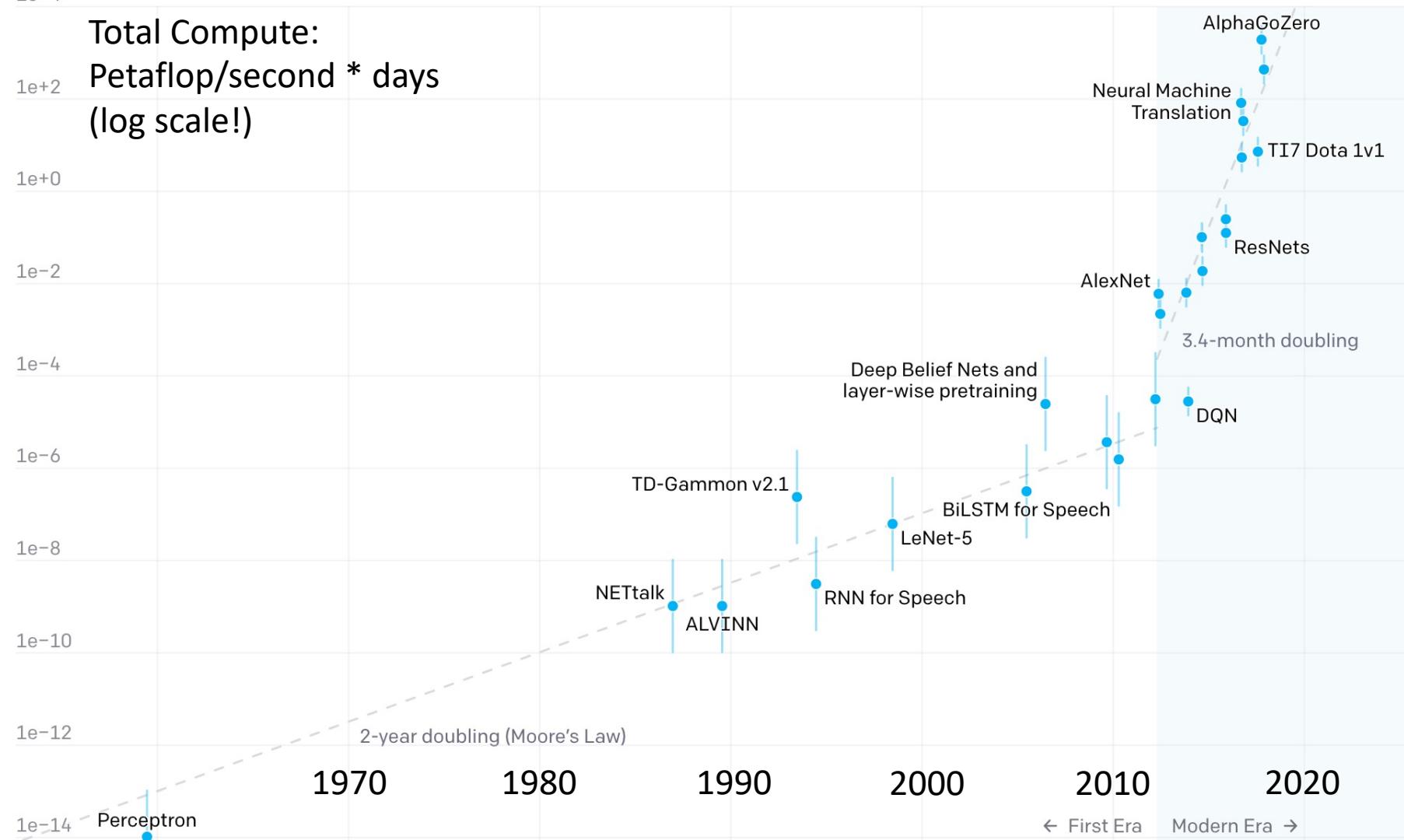
Bigger Models, More Data, More Compute

GFLOP per Dollar

● GPU (Tensor Core)



Bigger Models, More Data, More Compute



Bigger Models, More Data, More Compute

Two examples that came out since the start of this class:

PaLM: Transformer Language Model with 540B parameters
Trained on 780 billion tokens using 6144 TPU-v4 chips

DALL-E 2: Text-to-Image model trained on 650M image/text pairs

Both show qualitative leaps in performance over prior systems

Bigger Models, More Data, More Compute: PaLM

Model can explain jokes with zero-shot prompting!

Prompt:

I will explain these jokes:

(1) The problem with kleptomaniacs is that they always take things literally.

Explanation: This joke is wordplay. Someone who "takes things literally" is someone who doesn't fully understand social cues and context, which is a negative trait. But the definition of kleomania is someone who literally takes things.

(2) Always borrow money from a pessimist. They'll never expect it back.

Explanation: Most people expect you to pay them back when you borrow money, however a pessimist is someone who always assumes the worst, so if you borrow money from them, they will expect that you won't pay them back anyways.

Bigger Models, More Data, More Compute: PaLM

Model can explain jokes with zero-shot prompting!

Input: I tried 10,000 random restarts of my neural network, but I was accused of overfitting. I guess no good seed goes unpunished.

Bigger Models, More Data, More Compute: PaLM

Model can explain jokes with zero-shot prompting!

Input: I tried 10,000 random restarts of my neural network, but I was accused of overfitting. I guess no good seed goes unpunished.

Model Output: This joke is a pun. A neural network is a computer program that can learn from data. A "seed" is a number that is used to initialize a random number generator. A "good seed" is a number that produces a good random number generator. The phrase "no good deed goes unpunished" means that if you do something good, you will be punished for it.

Bigger Models, More Data, More Compute: PaLM

Model can explain jokes with zero-shot prompting!

Input: I was going to fly to visit my family on April 6th. My mom said "Oh great, your step dad's poetry reading is that night!" So now I'm flying in on April 7th.

Bigger Models, More Data, More Compute: PaLM

Model can explain jokes with zero-shot prompting!

Input: I was going to fly to visit my family on April 6th. My mom said "Oh great, your step dad's poetry reading is that night!" So now I'm flying in on April 7th.

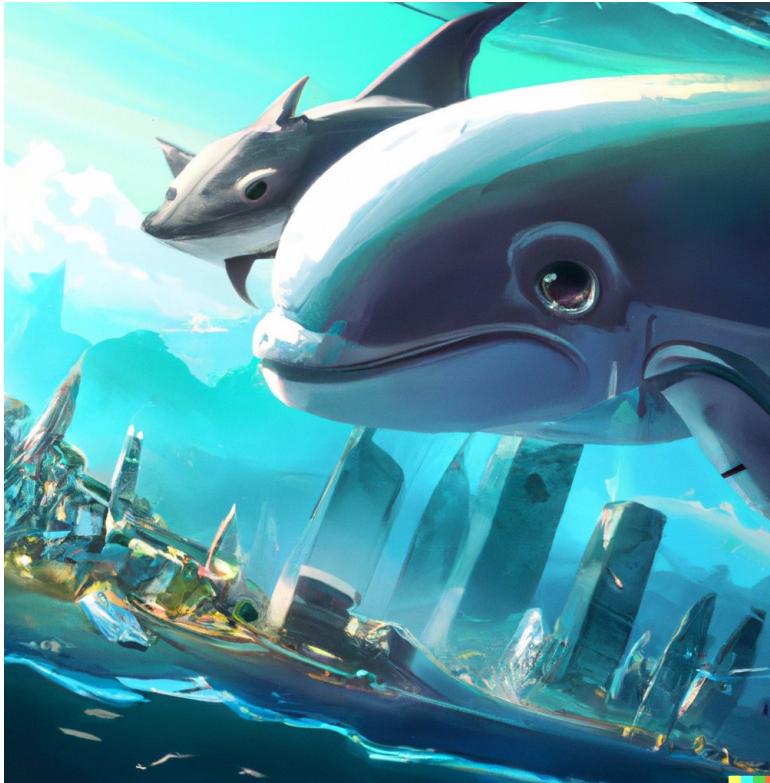
Model Output: The joke is that the speaker's mother is trying to get them to go to their step dad's poetry reading, but the speaker doesn't want to go, so they are changing their flight to the day after the poetry reading.

Bigger Models, More Data, More Compute: DALL-E 2

A rabbit detective sitting on a park bench and reading a newspaper in a victorian setting



A shark and a dolphin cruise hand-in-hand with an undersea city in the background



Robot dinosaurs versus monster trucks in the colosseum



Source: <https://twitter.com/sama/status/1511724264629678084>

Bigger Models, More Data, More Compute,
More problems

ML Systems can encode bias
Large models can lack common sense
Who should control models and data?

Bigger Models, More Data, More Compute,
More problems

ML Systems can encode bias

Large models can lack common sense

Who should control models and data?

Stepping Back: Why Build ML Systems?

Automate decision making, so machines can make decision instead of people.

Ideal: Automated decisions can be cheaper, more accurate, more impartial, improve our lives

Reality: If we aren't careful, automated decisions can encode bias, harm people, make lives worse

Allocative Harms

- Some systems decide how to *allocate resources*
- If the system is biased, it may allocate resources unfairly or perpetuate inequality
- Examples:
 - Sentencing criminals
 - Loan applications
 - Mortgage applications
 - Insurance rates
 - College admissions
 - Job applications

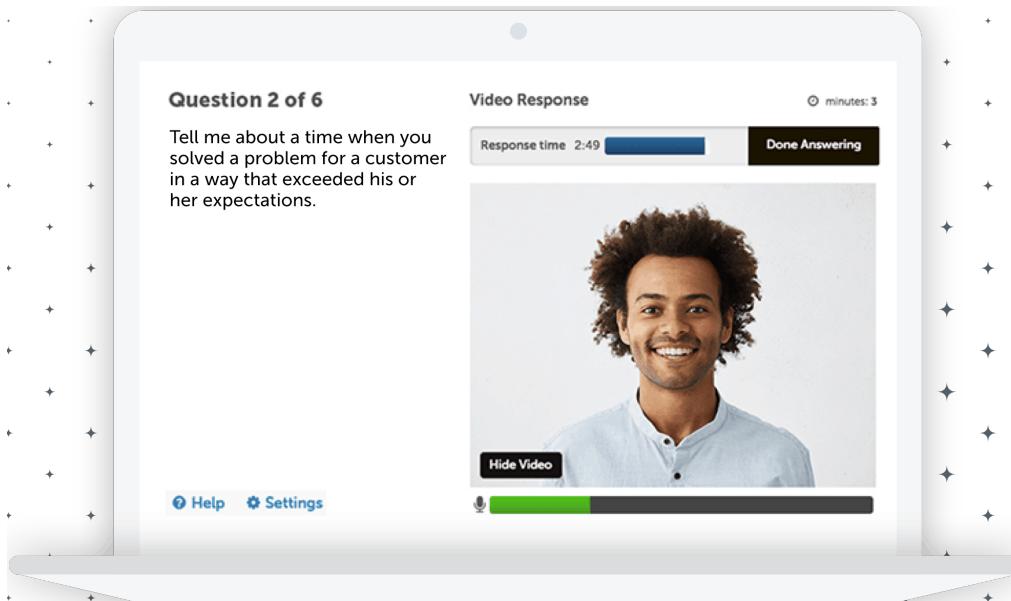
Barocas et al, "The Problem With Bias: Allocative Versus Representational Harms in Machine Learning", SIGCIS 2017
Kate Crawford, "The Trouble with Bias", NeurIPS 2017 Keynote

Example: Video Interviewing

Technology

A face-scanning algorithm increasingly decides whether you deserve the job

HireVue claims it uses artificial intelligence to decide who's best for a job. Outside experts call it 'profoundly disturbing.'



Source: <https://www.washingtonpost.com/technology/2019/10/22/ai-hiring-face-scanning-algorithm-increasingly-decides-whether-you-deserve-job/>

<https://www.hirevue.com/platform/online-video-interviewing-software>

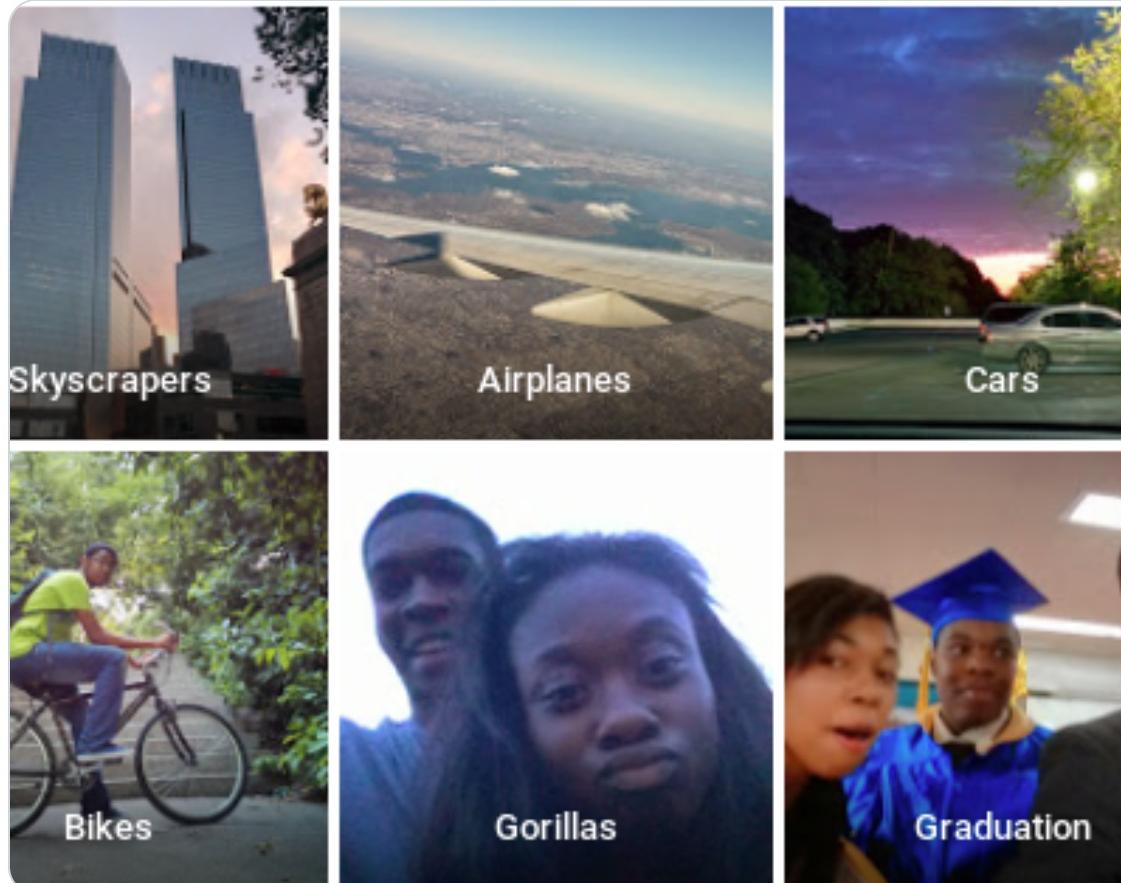
Example Credit: Timnit Gebru

Representational Harms

A system reinforces harmful stereotypes

Representational Harms: Image classifiers

A system reinforces harmful stereotypes



Barocas et al, "The Problem With Bias: Allocative Versus Representational Harms in Machine Learning", SIGCIS 2017

Kate Crawford, "The Trouble with Bias", NeurIPS 2017 Keynote

Source: <https://twitter.com/jackyalcine/status/615329515909156865> (2015)

Representational Harms: Machine Translation

The screenshot shows the Google Translate interface. At the top, it says "English translation makes assumptions". Below that, the source language is "HUNGARIAN - DETECTED" and the target language is "ENGLISH". The Hungarian input is: "Ő szép. Ő okos. Ő olvas. Ő mosogat. Ő épít. Ő varr. Ő tanít. Ő főz. Ő kutat. Ő gyereket nevel. Ő zenél. Ő takarító. Ő politikus. Ő sok pénzt keres. Ő süteményt süt. Ő professzor. Ő asszisztens." The English output is: "She is beautiful. He is clever. He reads. She washes the dishes. He builds. She sews. He teaches. She cooks. He's researching. She is raising a child. He plays music. She's a cleaner. He is a politician. He makes a lot of money. She is baking a cake. He's a professor. She's an assistant." A red "X" is next to the first sentence in Hungarian, indicating a mistake. Below the interface, the text "Hungarian does not use gendered pronouns" is displayed.

English translation makes assumptions

HUNGARIAN - DETECTED POLISH PO ENGLISH POLISH PORTUGUESE

Ő szép. Ő okos. Ő olvas. Ő mosogat. Ő épít. Ő varr. Ő tanít. Ő főz. Ő kutat. Ő gyereket nevel. Ő zenél. Ő takarító. Ő politikus. Ő sok pénzt keres. Ő süteményt süt. Ő professzor. Ő asszisztens. |

She is beautiful. He is clever. He reads. She washes the dishes. He builds. She sews. He teaches. She cooks. He's researching. She is raising a child. He plays music. She's a cleaner. He is a politician. He makes a lot of money. She is baking a cake. He's a professor. She's an assistant.

Hungarian does not use gendered pronouns

Source: https://www.reddit.com/r/europe/comments/m9uphb/hungarian_has_no_gendered_pronouns_so_google

Representational Harms: Machine Translation

The screenshot shows the Google Translate web interface. The source text "ő szép" is detected as HUNGARIAN. The target language is English. The translation provided is "she is beautiful" (feminine), which is marked as gender-specific. Below this, another suggestion is shown: "he is beautiful" (masculine). The interface includes standard translation controls like microphone, speaker, and share icons.

≡ Google Translate

HUNGARIAN - DETECTED ENGLISH ↔ HUNGARIAN ENGLISH SPANISH

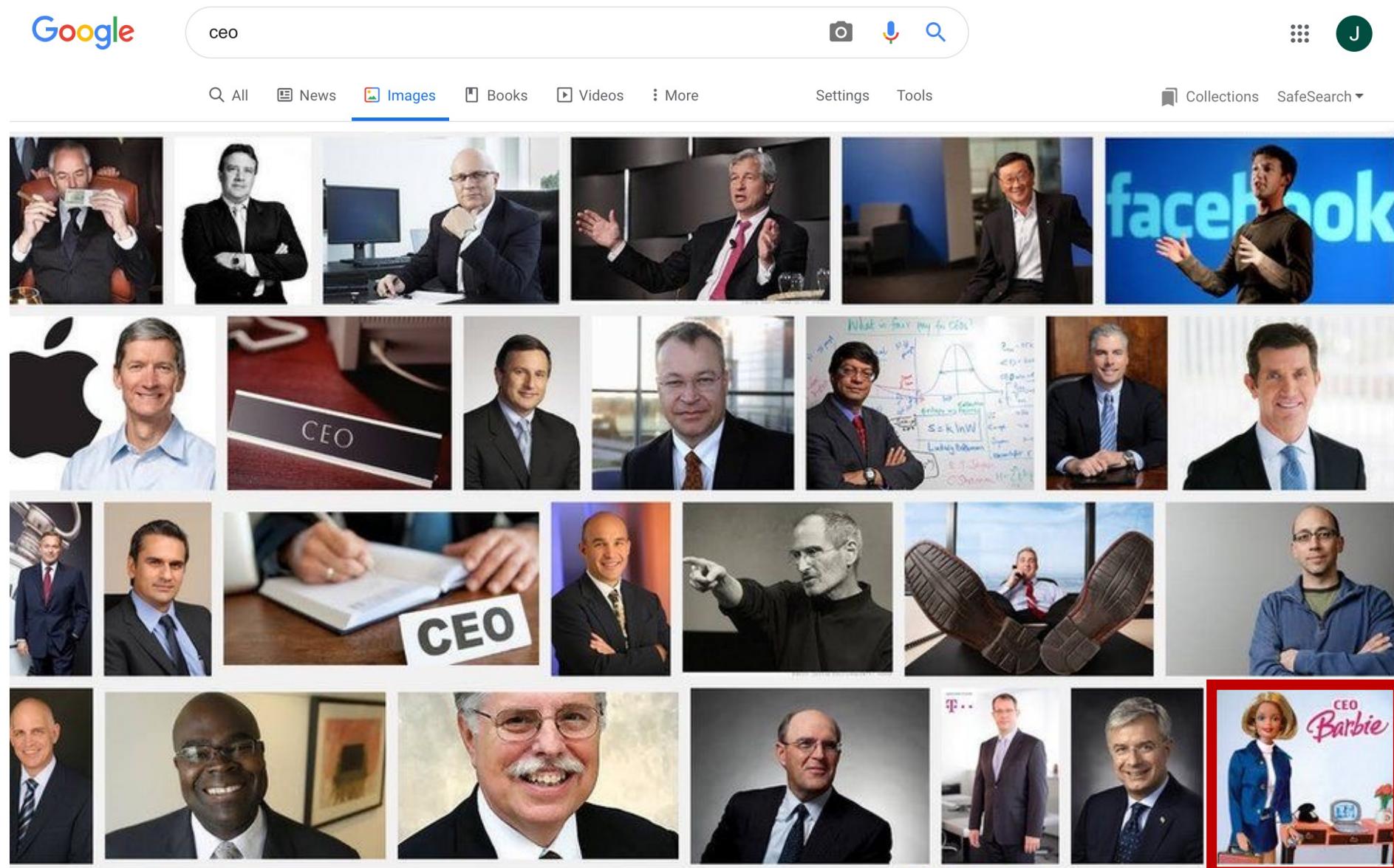
ő szép

Possible solution:
Change the task; offer
multiple suggestions

Translations are gender-specific. [LEARN MORE](#)

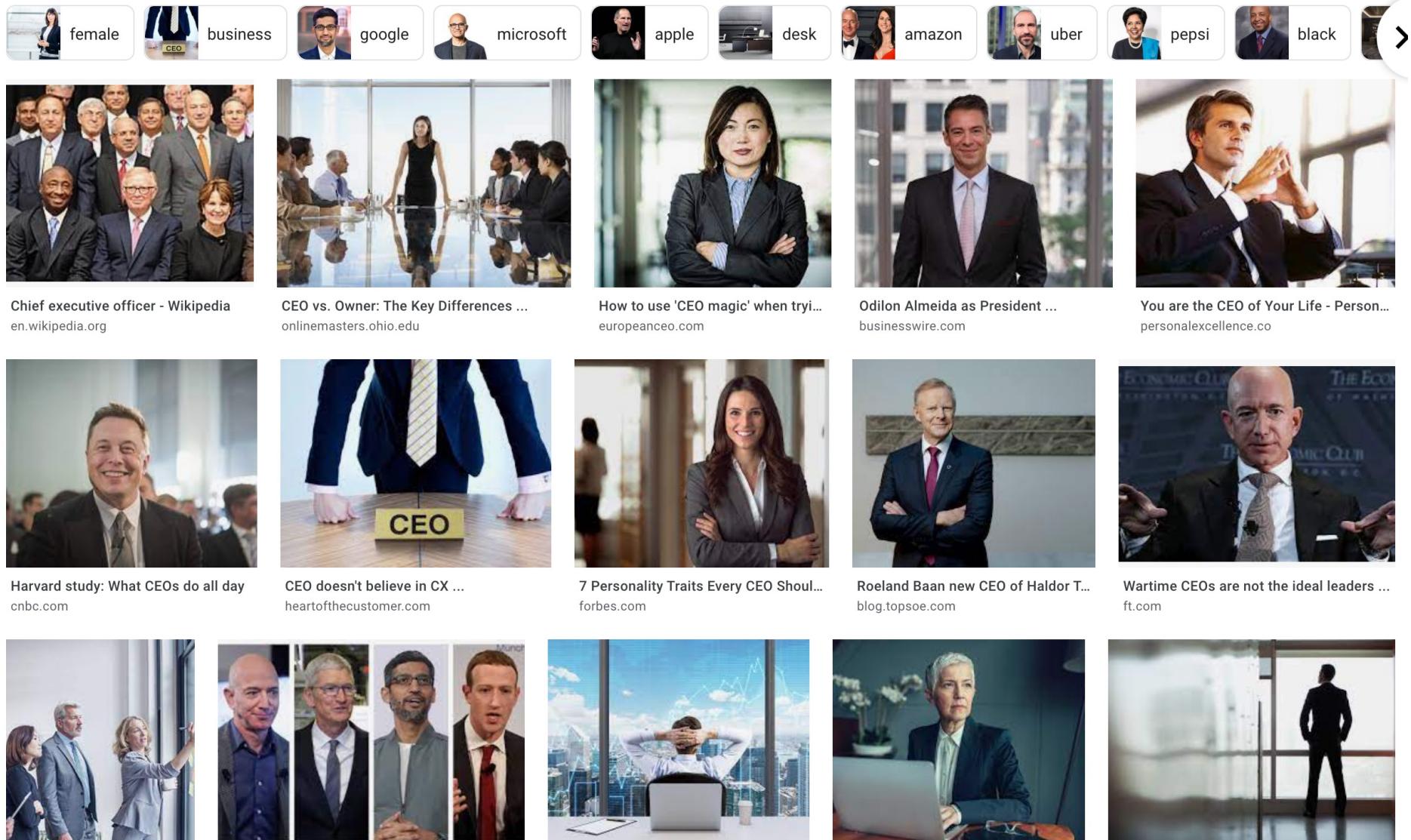
she is beautiful (feminine)

he is beautiful (masculine)



First
woman:
CEO
Barbie =(

Source: <https://www.bbc.com/news/newsbeat-32332603>



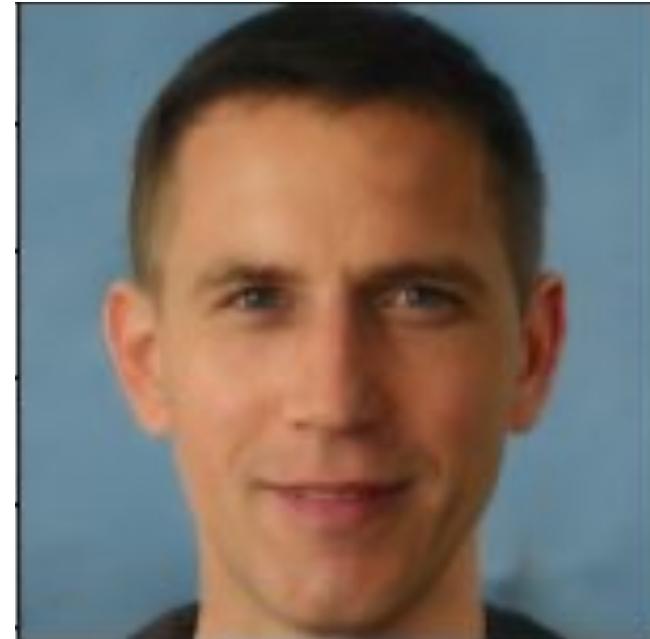
Recent results more diverse

Representational Harm in Super-Resolution

Input: Low-Resolution Face



Output: High-Resolution Face



Menon et al, "PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models", CVPR 2020
Example source: <https://twitter.com/Chicken3gg/status/1274314622447820801>

Representational Harm in DALL-E 2

Text Prompt: “lawyer”



Ramesh et al, “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022

<https://github.com/openai/dalle-2-preview/blob/main/system-card.md>

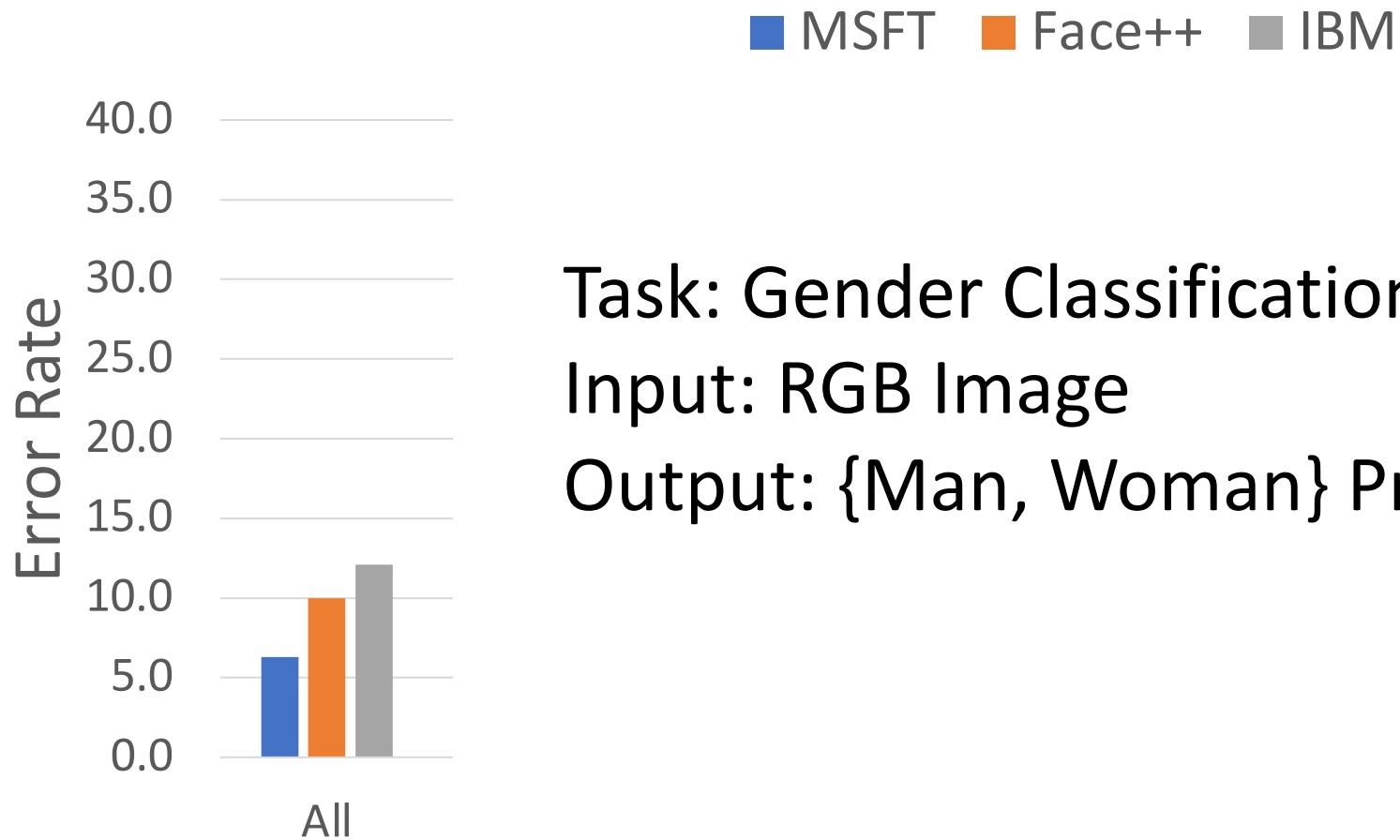
Representational Harm in DALL-E 2

Text Prompt: “nurse”



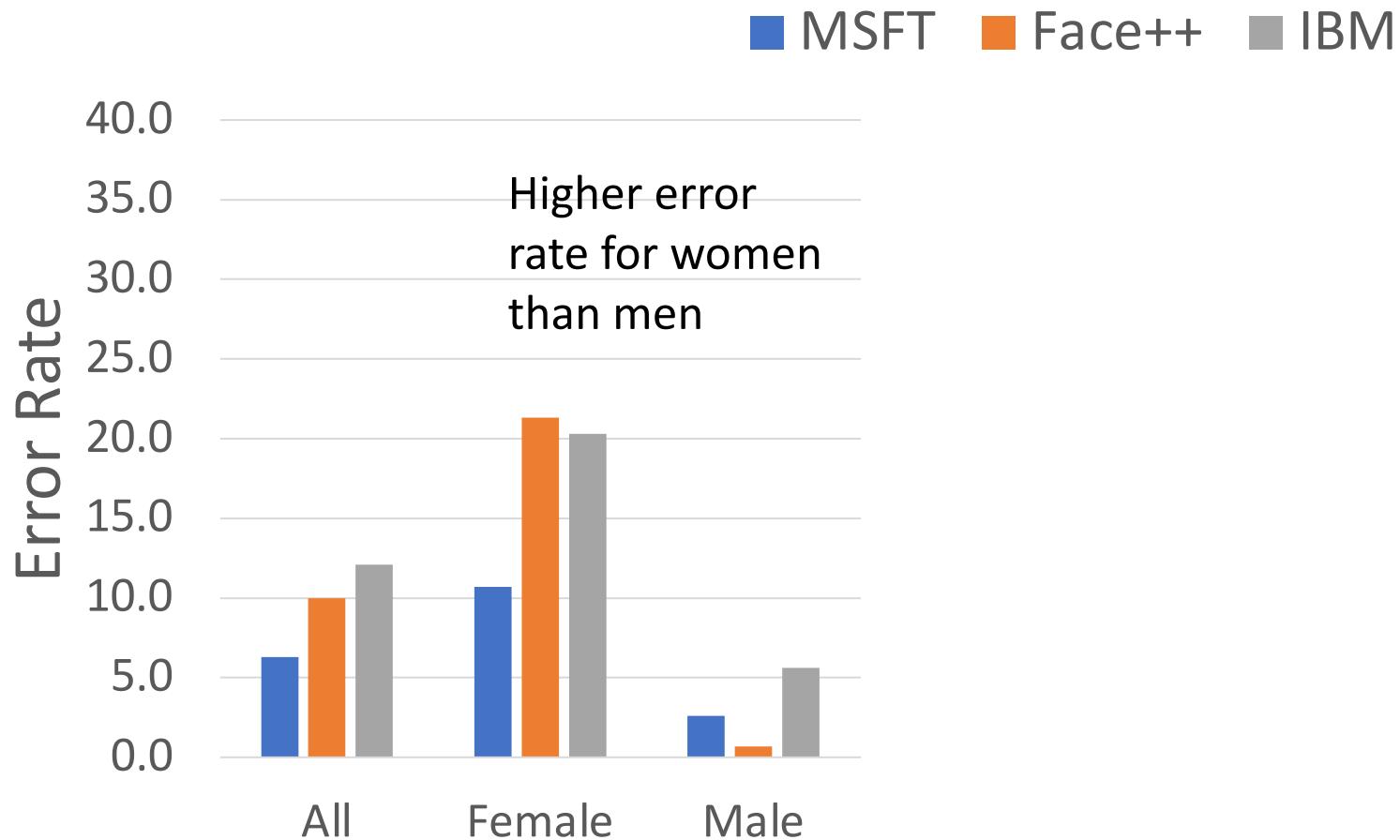
Ramesh et al, “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022
<https://github.com/openai/dalle-2-preview/blob/main/system-card.md>

Gender Shades: Intersectionality



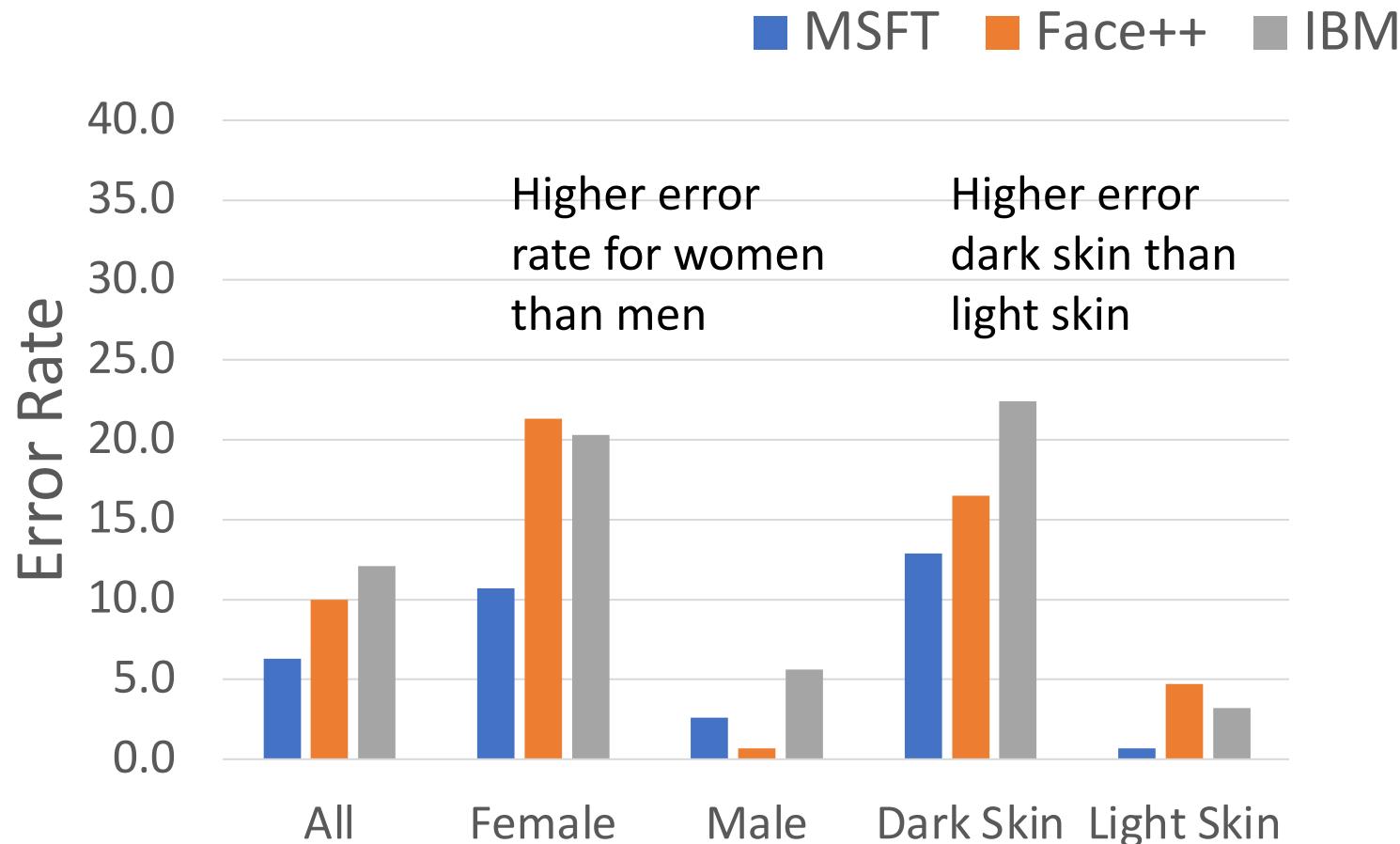
Task: Gender Classification
Input: RGB Image
Output: {Man, Woman} Prediction

Gender Shades: Intersectionality



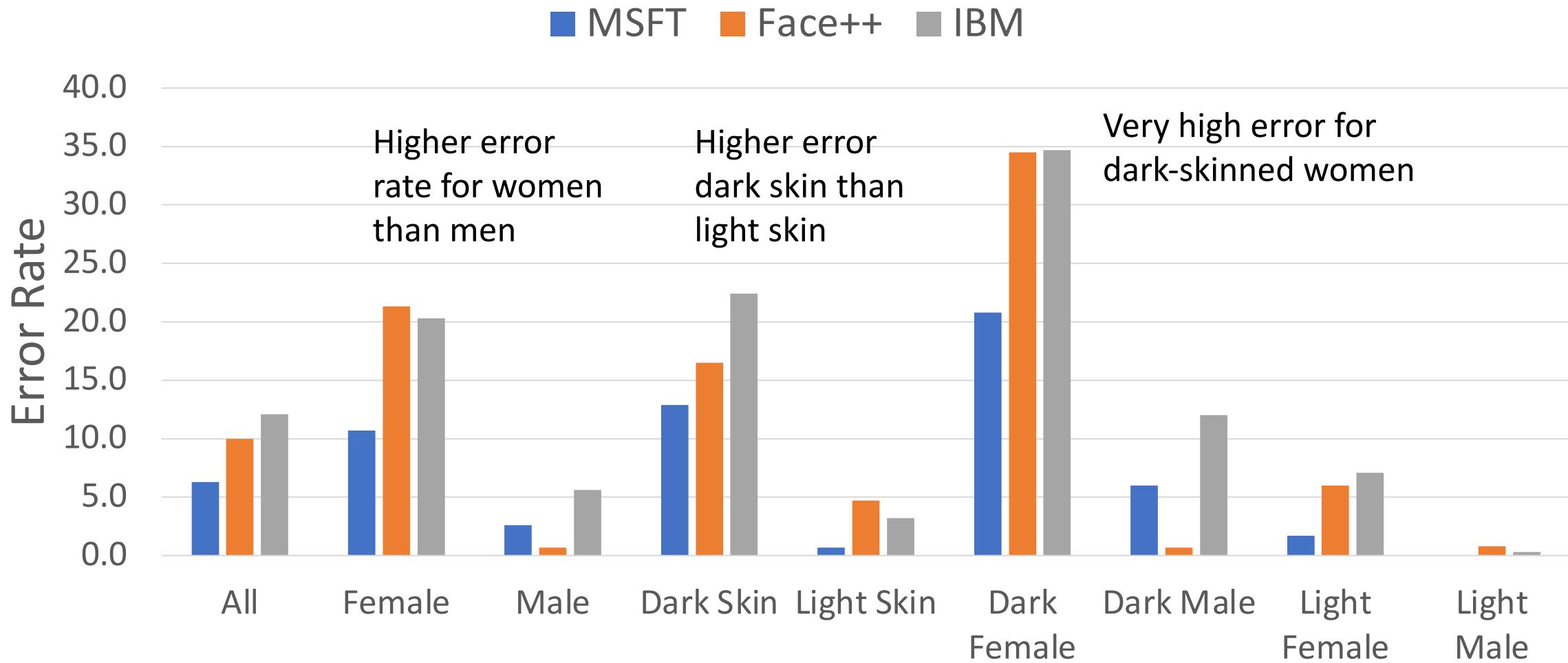
Buolamwini and Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification", FAT* 2018

Gender Shades: Intersectionality



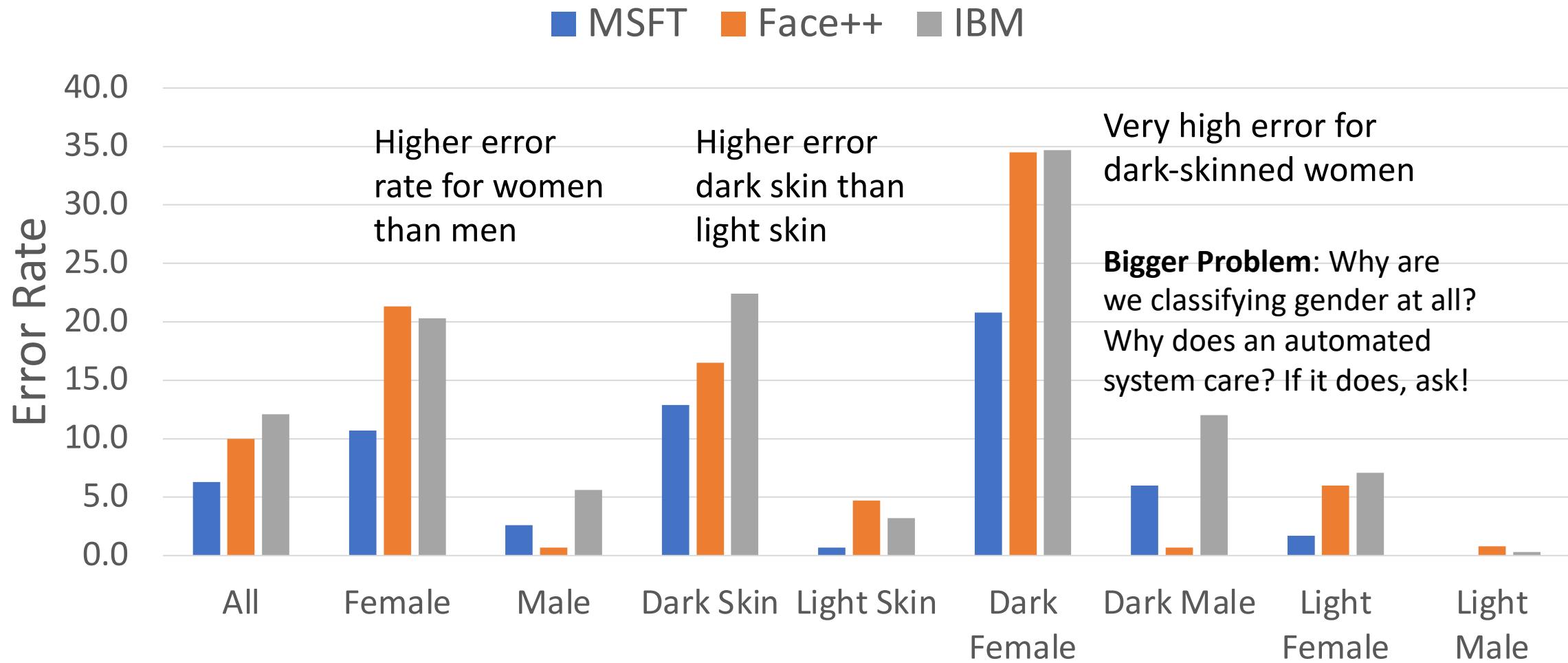
Buolamwini and Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification", FAT* 2018

Gender Shades: Intersectionality



Buolamwini and Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification", FAT* 2018

Gender Shades: Intersectionality



Buolamwini and Gebru, "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification", FAT* 2018

Datasheets for Datasets

Idea: A standard list of questions to answer when releasing a dataset. Who created it? Why? What is in it? How was it labeled?

A Database for Studying Face Recognition in Unconstrained Environments	Labeled Faces in the Wild
<p>Motivation</p> <p>For what purpose was the dataset created? Was there a specific task in mind? Was there a specific gap that needed to be filled? Please provide a description.</p> <p>Labeled Faces in the Wild was created to provide images that can be used to study face recognition in the unconstrained setting where image characteristics (such as pose, illumination, resolution, focus), subject demographic makeup (such as age, gender, race) or appearance (such as hairstyle, makeup, clothing) cannot be controlled. The dataset was created for the specific task of pair matching: given a pair of images each containing a face, determine whether or not the images are of the same person.¹</p> <p>Who created this dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organization)?</p> <p>The initial version of the dataset was created by Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, most of whom were researchers at the University of Massachusetts Amherst at the time of the dataset's release in 2007.</p> <p>Who funded the creation of the dataset? If there is an associated grant, please provide the name of the grantor and the grant name and number.</p> <p>The construction of the LFW database was supported by a United States National Science Foundation CAREER Award.</p>	<p>What data does each instance consist of? "Raw" data (e.g., unprocessed text or images) or features? In either case, please provide a description.</p> <p>Each instance contains a pair of images that are 250 by 250 pixels in JPEG 2.0 format.</p> <p>Is there a label or target associated with each instance? If so, please provide a description.</p> <p>Each image is accompanied by a label indicating the name of the person in the image.</p> <p>Is any information missing from individual instances? If so, please provide a description, explaining why this information is missing (e.g., because it was unavailable). This does not include intentionally removed information, but might include, e.g., redacted text.</p> <p>Everything is included in the dataset.</p> <p>Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? If so, please describe how these relationships are made explicit.</p> <p>There are no known relationships between instances except for the fact that they are all individuals who appeared in news sources.</p>

Model Cards

Idea: A standard list of questions to answer when releasing a trained model. Who created it? What data was it trained on? What should it be used for? What should it **not** be used for?

Model Card	
<ul style="list-style-type: none">• Model Details. Basic information about the model.<ul style="list-style-type: none">– Person or organization developing model– Model date– Model version– Model type– Information about training algorithms, parameters, fairness constraints or other applied approaches, and features– Paper or other resource for more information– Citation details– License– Where to send questions or comments about the model	<ul style="list-style-type: none">– Evaluation factors
<ul style="list-style-type: none">• Intended Use. Use cases that were envisioned during development.<ul style="list-style-type: none">– Primary intended uses– Primary intended users– Out-of-scope use cases	<ul style="list-style-type: none">• Metrics. Metrics should be chosen to reflect potential real-world impacts of the model.<ul style="list-style-type: none">– Model performance measures– Decision thresholds– Variation approaches
<ul style="list-style-type: none">• Factors. Factors could include demographic or phenotypic groups, environmental conditions, technical attributes, or others listed in Section 4.3.<ul style="list-style-type: none">– Relevant factors	<ul style="list-style-type: none">• Evaluation Data. Details on the dataset(s) used for the quantitative analyses in the card.<ul style="list-style-type: none">– Datasets– Motivation– Preprocessing
	<ul style="list-style-type: none">• Training Data. May not be possible to provide in practice. When possible, this section should mirror Evaluation Data. If such detail is not possible, minimal allowable information should be provided here, such as details of the distribution over various factors in the training datasets.
	<ul style="list-style-type: none">• Quantitative Analyses<ul style="list-style-type: none">– Unitary results– Intersectional results
	<ul style="list-style-type: none">• Ethical Considerations
	<ul style="list-style-type: none">• Caveats and Recommendations

Mitchell et al, "Model Cards for Model Reporting", FAccT 2019

Model Cards

Out-of-Scope Use Cases

Some models are just for research and not to be deployed. Make it clear!

Any deployed use case of the model - whether commercial or not - is currently out of scope. Non-deployed use cases such as image search in a constrained environment, are also not recommended unless there is thorough in-domain testing of the model with a specific, fixed class taxonomy. This is because our safety assessment demonstrated a high need for task specific testing especially given the variability of CLIP's performance with different class taxonomies. This makes untested and unconstrained deployment of the model in any use case currently potentially harmful.

Certain use cases which would fall under the domain of surveillance and facial recognition are always out-of-scope regardless of performance of the model. This is because the use of artificial intelligence for tasks such as these can be premature currently given the lack of testing norms and checks to ensure its fair use.

<https://github.com/openai/CLIP/blob/main/model-card.md>

Bigger Models, More Data, More Compute,
More problems

ML Systems can encode bias

Large models can lack common sense

Who should control models and data?

Large Models Lack Common Sense

Some plants surrounding a lightbulb



A lightbulb surrounding some plants



Large vision + language models cannot correctly pair images with captions

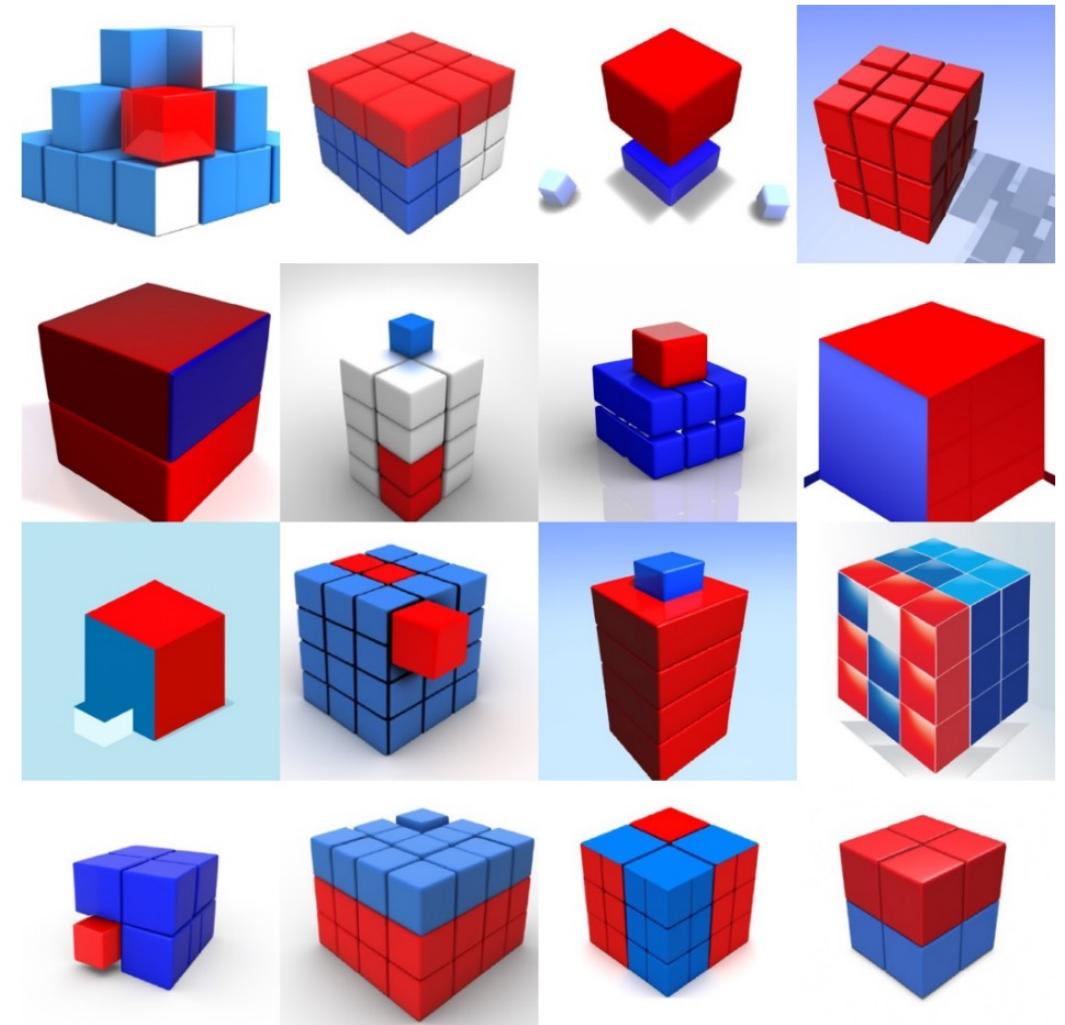
Thrush et al, "Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality", CVPR 2022

Large Models Lack Common Sense

Samples from DALL-E 2 for the prompt:

“a red cube on top of a blue cube”

Simple compositions of objects,
attributes, relationships often not
respected



Ramesh et al, “Hierarchical Text-Conditional Image Generation with CLIP Latents”, arXiv 2022

Large Models Lack Common Sense: GPT-2

Bold = prompt written by human

Italic = completion written by GPT-2

I was born in 1950. In the year 2025 my age will be 35. That was only a few years ago. Most things in life just continue to improve.

I see a black dog and a brown horse. The bigger animal's color is *black*, and the smaller is *brown*.

Examples generated using <https://taktotransformer.com/>

Large Models Lack Common Sense: GPT-3

Bold = prompt written by human

Italic = completion written by GPT-3

At the party, I poured myself a glass of lemonade, but it turned out to be too sour, so I added a little sugar. I didn't see a spoon handy, so I stirred it with a paper napkin. But that turned out to be a bad idea because *the napkin disintegrated in the glass. After I finished the drink, I threw the napkin away in a wastebasket—but when I picked up the glass to wash it out, there was a big black spot on my hand.*

Source: <https://cs.nyu.edu/~davise/papers/GPT3CompleteTests.html>

Large Models Lack Common Sense

Open question: Can large models learn common sense about the world from lots of (internet) data?
Or are there fundamental limitations?

Bigger Models, More Data, More Compute,
More problems

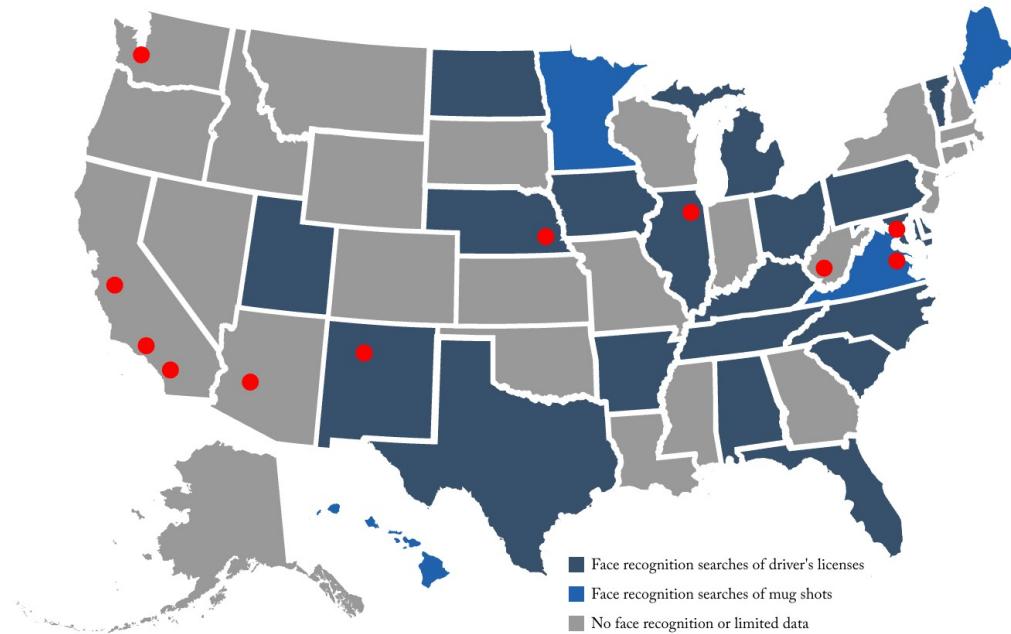
ML Systems can encode bias
Large models can lack common sense
Who should control models and data?

Who should control data?

Image copyright != Consent to use in a dataset

Who should control data?

Image copyright != Consent to use in a dataset



“One in two American adults is in a law enforcement face recognition network.”

Garvie, Bedoya, and Frankle: “The Perpetual Line-Up”, 2016, <https://www.perpetuallineup.org/>

Birhane and Prabhu, “Large Image Datasets: A Pyrrhic Win for Computer Vision?”, WACV 2021

Who should control models?

The largest models (e.g. PaLM, DALL-E 2) can only be trained by large non-academic institutions. Is this a problem?

Who should control models?

The largest models (e.g. PaLM, DALL-E 2) can only be trained by large non-academic institutions. Is this a problem?

Should governments regulate the use of ML-based solutions?

Bigger Models, More Data, More Compute,
More problems

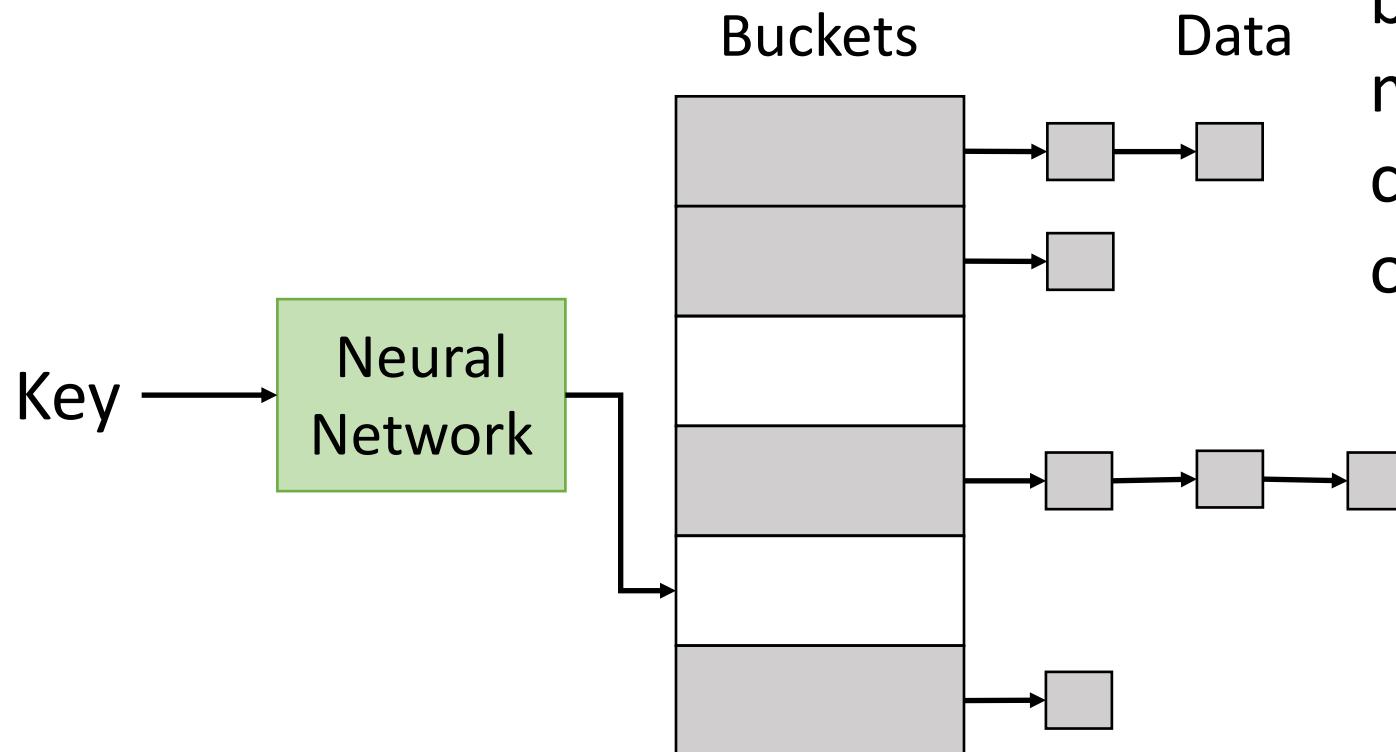
ML Systems can encode bias
Large models can lack common sense
Who should control models and data?

Deep Learning is Here to Stay

Deep Learning is Here to Stay
and will impact more than vision, speech, NLP

Deep Learning for Computer Science

Traditional Hash Table

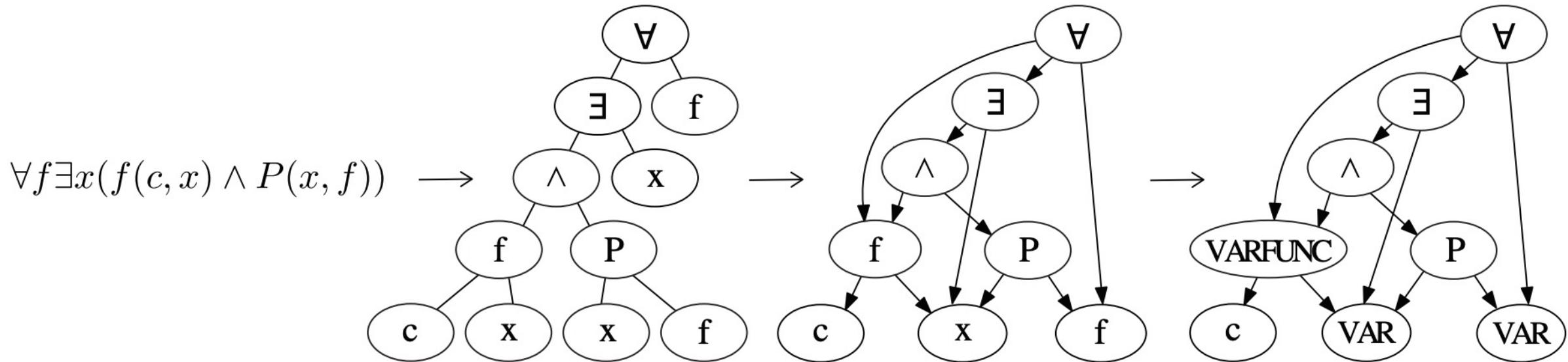


Learn to assign keys to buckets in a way that minimizes hash collisions for the types of data you encounter

Kraska et al, "The Case for Learned Index Structures", SIGMOD 2018

Deep Learning for Mathematics

Convert mathematical expressions into graphs,
process then with graph neural networks!



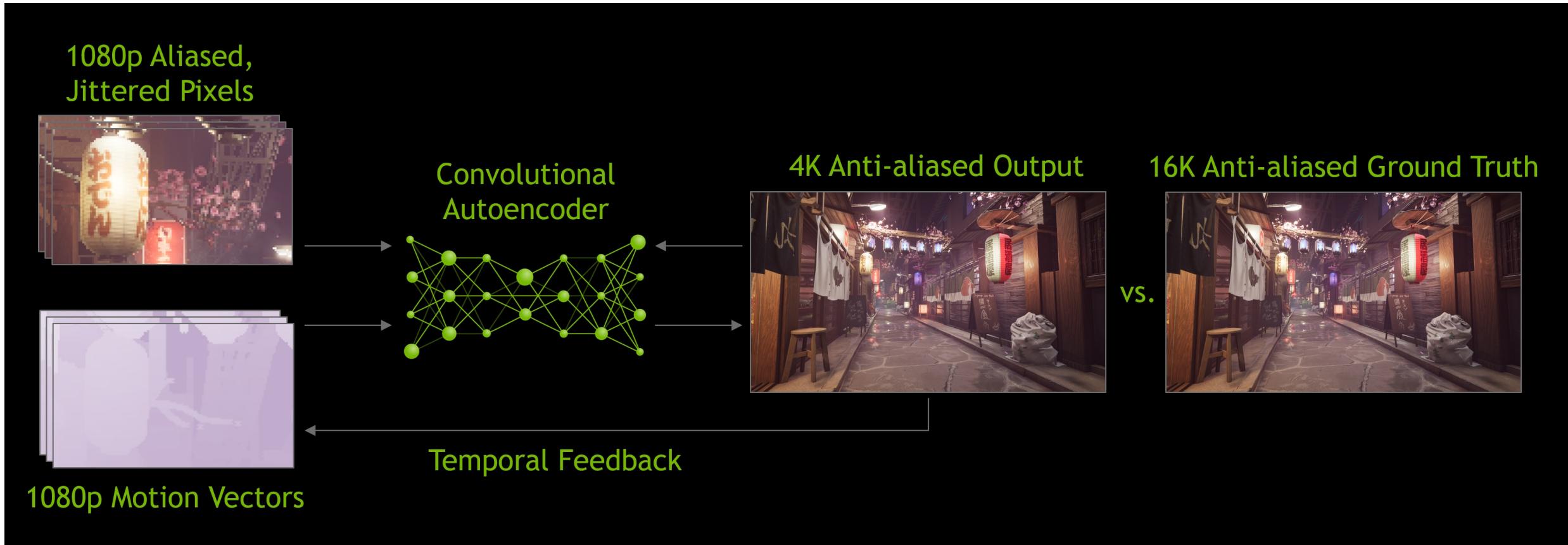
Applications: Theorem proving, symbolic integration

Wang et al, "Premise Selection for Theorem Proving by Deep Graph Embedding", NeurIPS 2017

Kaliszyk et al, "Reinforcement Learning of Theorem Proving", NeurIPS 2018

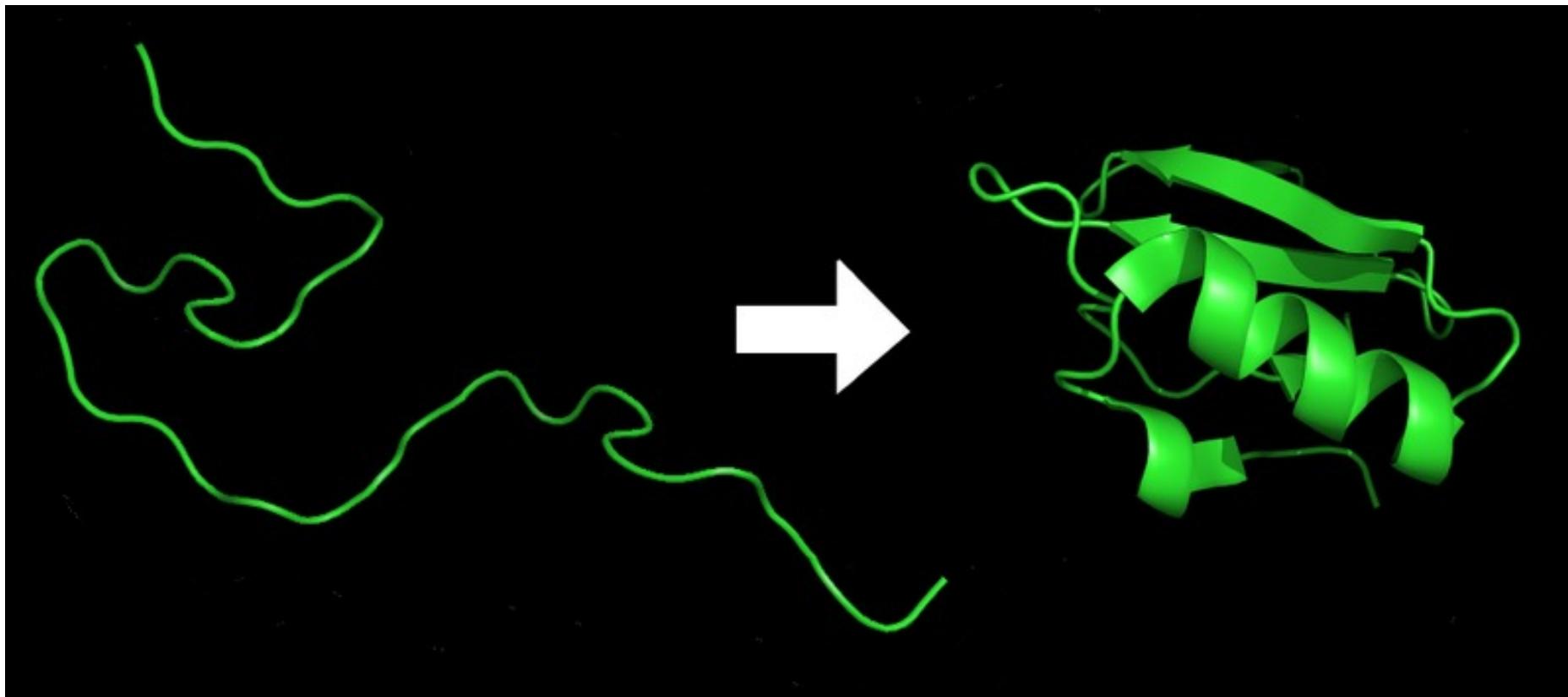
Lample and Charton, "Deep Learning for Symbolic Mathematics", arXiv 2019

Deep Learning for Graphics: NVIDIA DLSS



<https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/>

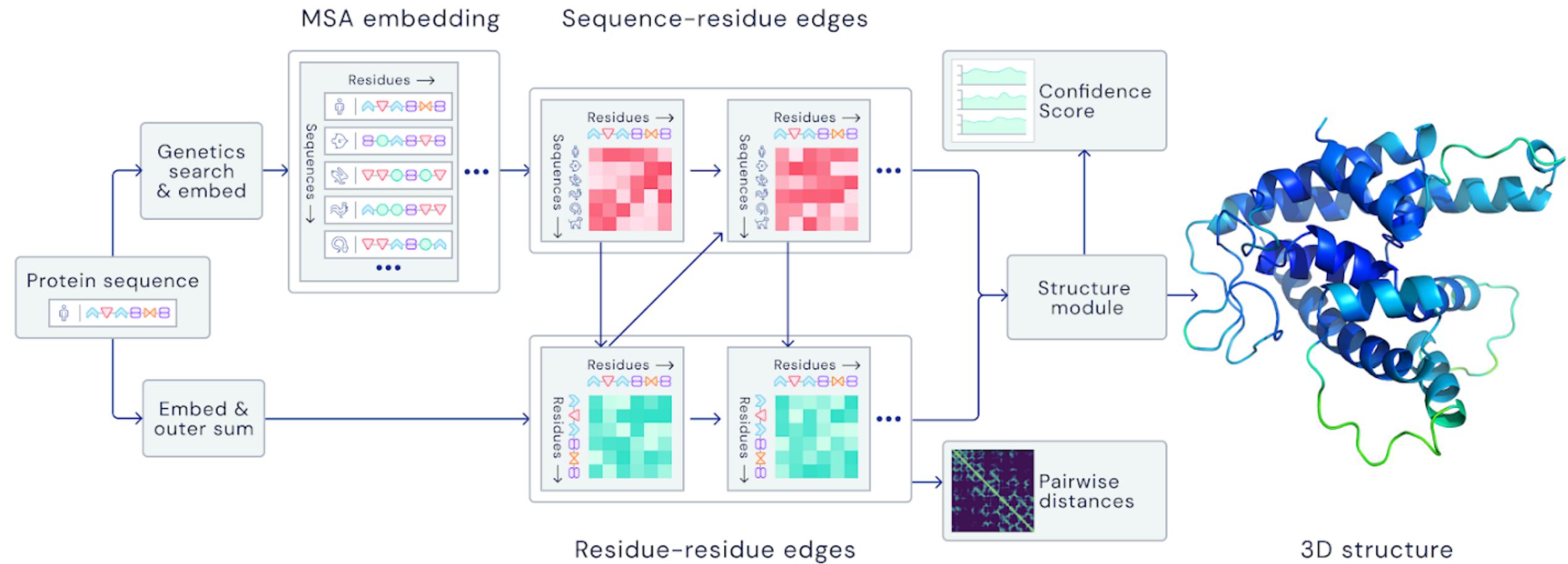
Deep Learning for Science: Protein Folding



Input: 1D sequence of amino acids

Output: 3D protein structure

Deep Learning for Science: AlphaFold 2



<https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology>

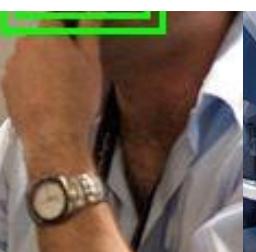
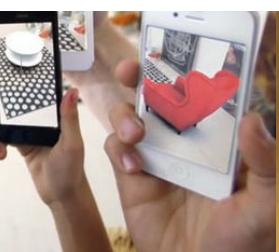




Computer Vision Technology

Can Better Our Lives

Now is a great time to be working in
computer vision and machine learning!



Thanks GSIs and IAs!

Graduate Student Instructors



Karan Desai (KD)



Janpreet Singh (JS)



Jim Yang



Wallace Sui (WS)

Instructional Aides



Gaurav Kaul



Zubin T Aysola

The End!