

Intro to Neural Nets

Session 5: Working with Image Data (CNNs)

Session Agenda

Convolutional Neural Networks (CNNs)

- What CNNs try to accomplish
- What is a convolution?
 - Padding, strides, filters
- What is pooling?
 - Max, min, avg pooling.

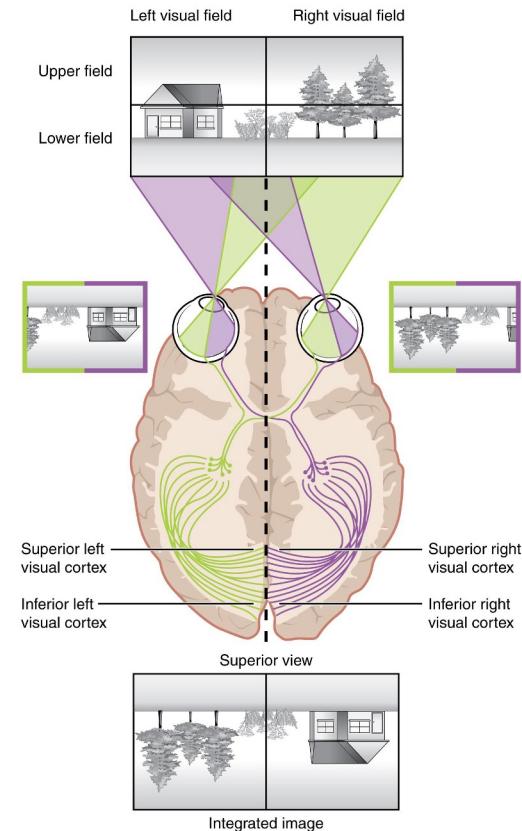
Other Stuff

- CNN specific techniques to avoid overfitting (data augmentation).
- Extracting feature representations from your trained model.
- Adapting pre-trained models (transfer learning).

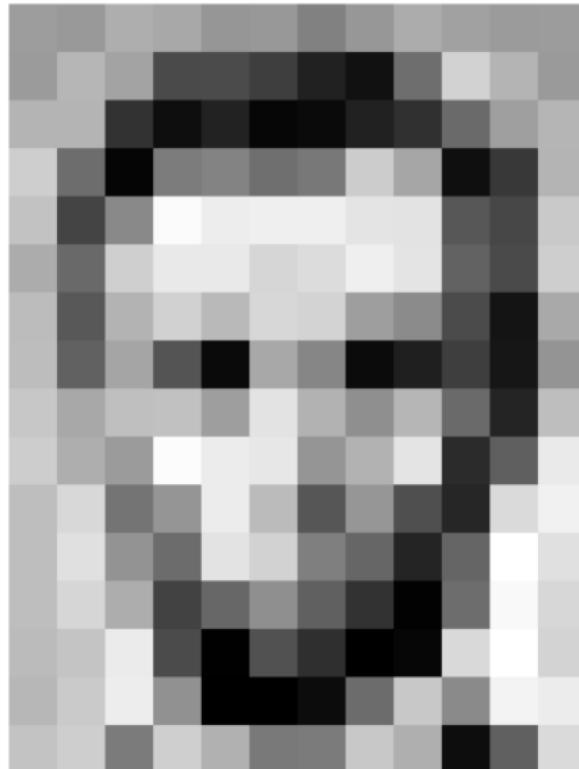
Inspiration for Convnets

Our Visual System

- Human eye is basically a 576-megapixel video camera.
- For comparison, the Pixel 6 camera is 50-megapixels.
- The human field of vision is not a square; something like a video camera that records individual image frames comprised of $24,000 \times 24,000$ pixels.



Images are Numbers



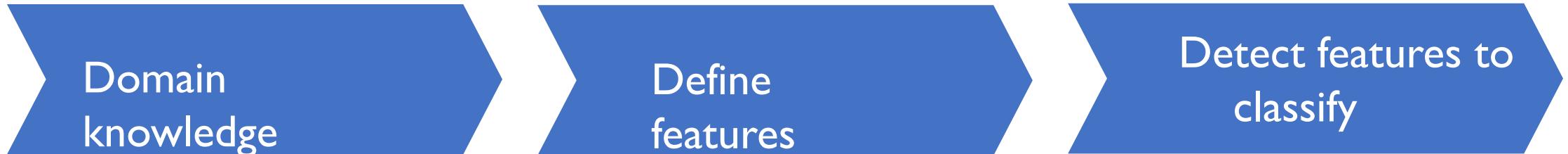
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	84	6	10	33	48	105	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	199	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	35	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

What the computer sees

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	35	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

An image is just a matrix of numbers
[0,255]! i.e., 1080x1080x3 for an RGB
image

Manual Feature Extraction



Problems?

Manual Feature Extraction

Domain knowledge

Define features

Detect features to classify



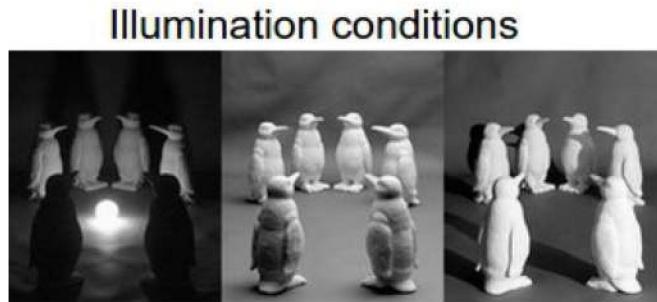
Scale variation



Deformation



Occlusion



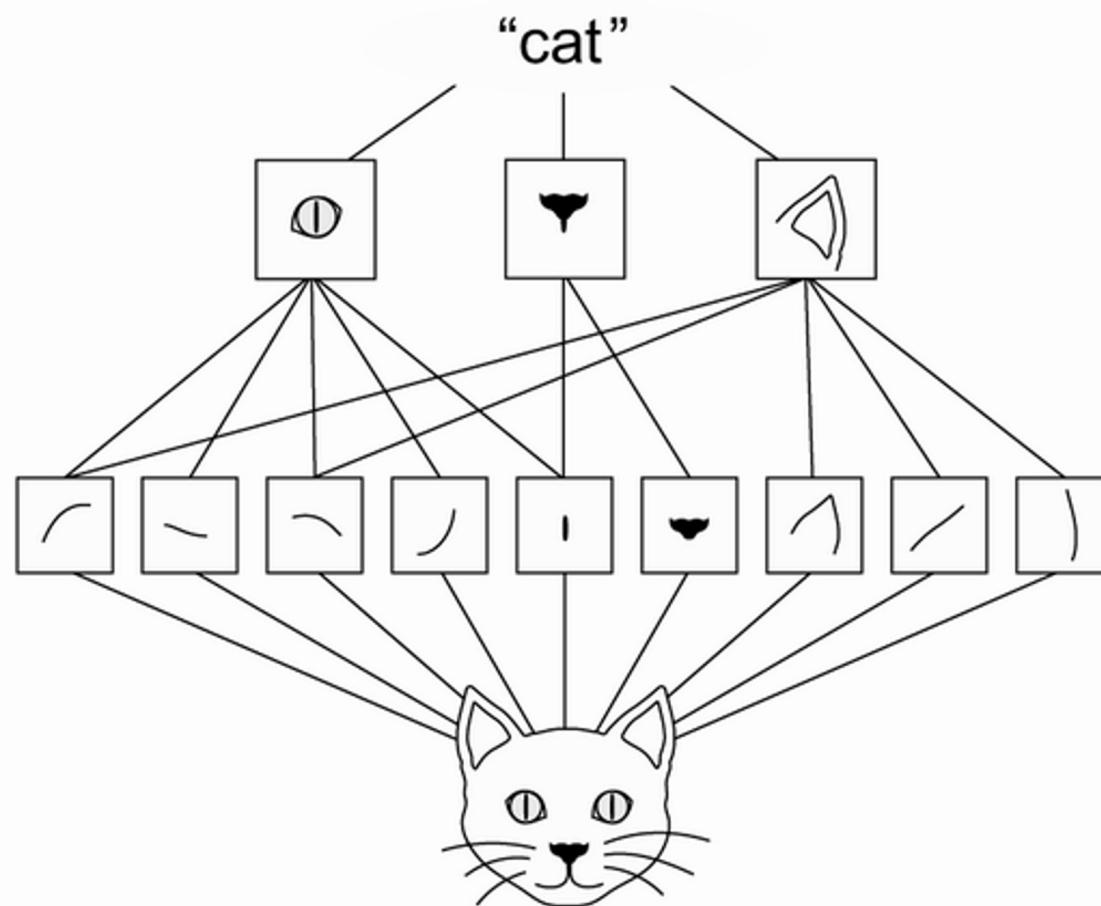
We Don't Detect Features 'Manually' Like That

How Does Your Visual System Work?

- We think that your brain processes individual visual receptors in groups, identifies combinations of inputs in proximity to one another that imply something like an edge (edge detection), combines that with color and so on. These low-level features are then processed together to arrive at higher level objects (e.g., a nose, a mouth, an eye).
- Those higher-level features are then processed together to yield a face (perhaps someone we know or do not know). Hence why you might have a hard time recognizing someone who has a new haircut, or who is wearing a facemask!



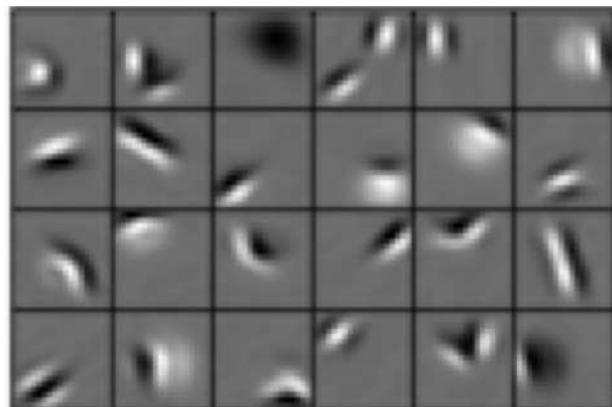
Feature Detection / Aggregation



Learning Feature Representations

Can we learn a **hierarchy of features** directly from the data instead of hand engineering?

Low level features



Edges, dark spots

Mid level features



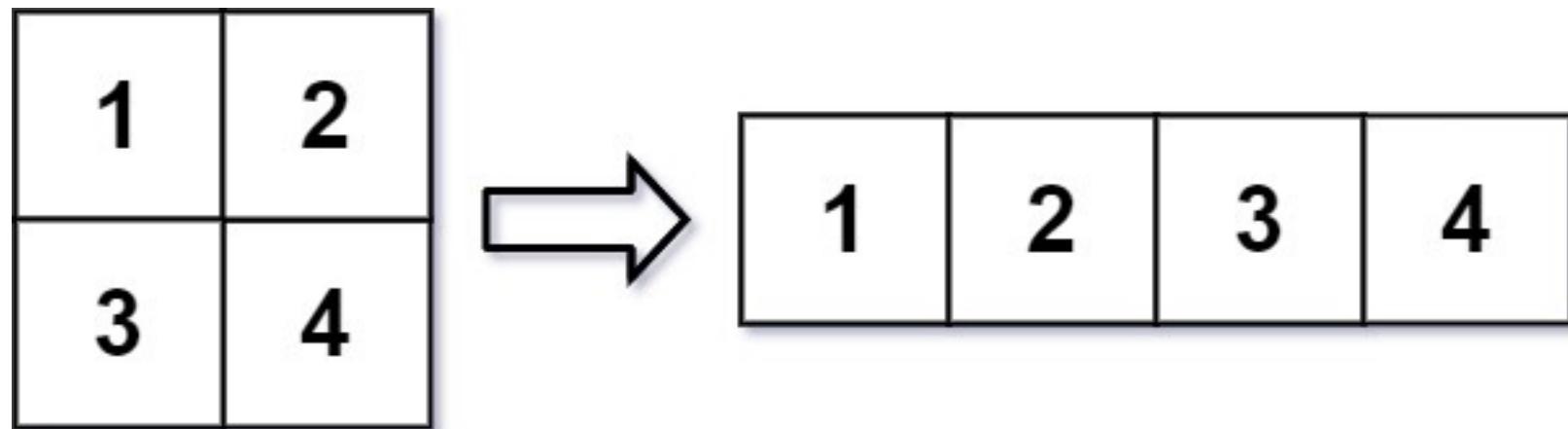
Eyes, ears, nose

High level features



Facial structure

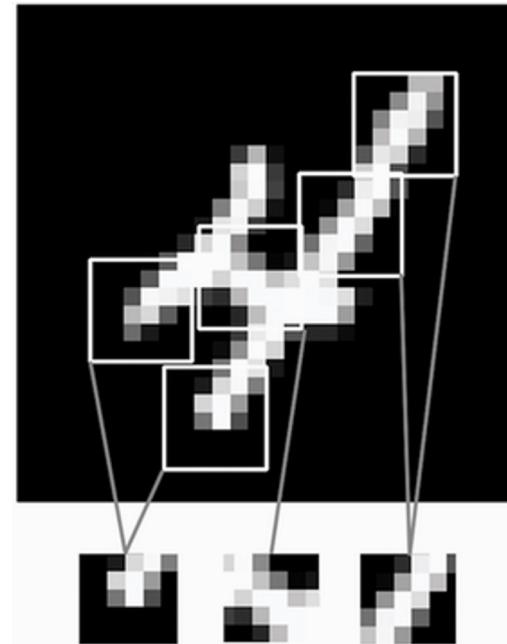
Why are Fully Connected Dense Layers (MLPs) Not Good for This?



We Need to Preserve Spatial Arrangements

Hone-in On Sub-sections of the Image

- So, if we have a 28x28 image, we might separately consider 3x3 pixel subsection of that image. Each subsection (they can be overlapping) is represented by its own node in the first hidden layer.
- That local input matrix (subfield) is considered in tandem with a ‘filter’ a matrix of weights. A filter might be something like



The Convolution Operation

Consider in Matrix Representation

- We have the raw image data (a.k.a. input feature map), the filter, and the result of passing our filter over our image (a.k.a. output feature map). We will have one output feature map for a given image, per filter (each filter is intended to detect a different type of feature).
- The filter elements are just weights for the Conv layer; we learn the filter values as part of the backpropagation process. So, the CNN will figure out what features to look for to predict the label (probably what a baby does when its first board and first learning how to process visual information).

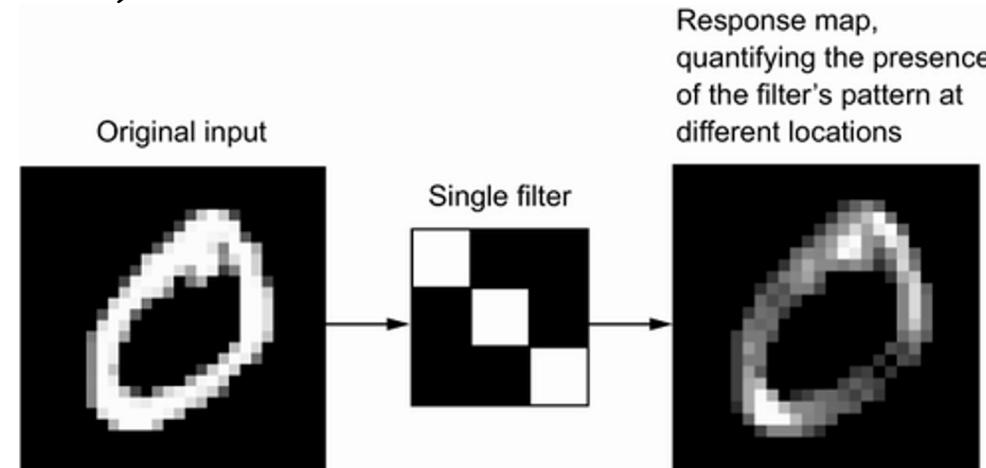
$$\begin{array}{|c|c|c|c|c|c|} \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline 10 & 10 & 10 & 0 & 0 & 0 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline 1 & 0 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline 0 & 30 & 30 & 0 \\ \hline 0 & 30 & 30 & 0 \\ \hline 0 & 30 & 30 & 0 \\ \hline 0 & 30 & 30 & 0 \\ \hline \end{array}$$


© Gordon Burtch, 2025

The Convolution Operation

Consider in Matrix Representation

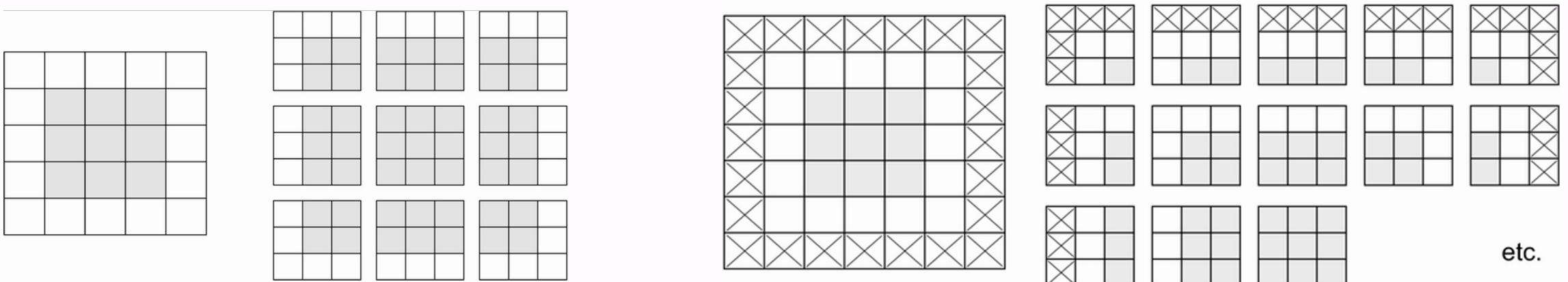
- We have the raw image data (a.k.a. input feature map), the filter, and the result of passing our filter over our image (a.k.a. output feature map). We will have one output feature map for a given image, per filter (each filter is intended to detect a different type of feature).
- The filter elements are just weights for the Conv layer; we learn the filter values as part of the backpropagation process. So, the CNN will figure out what features to look for to predict the label (probably what a baby does when its first board and first learning how to process visual information).



Padding

Padding

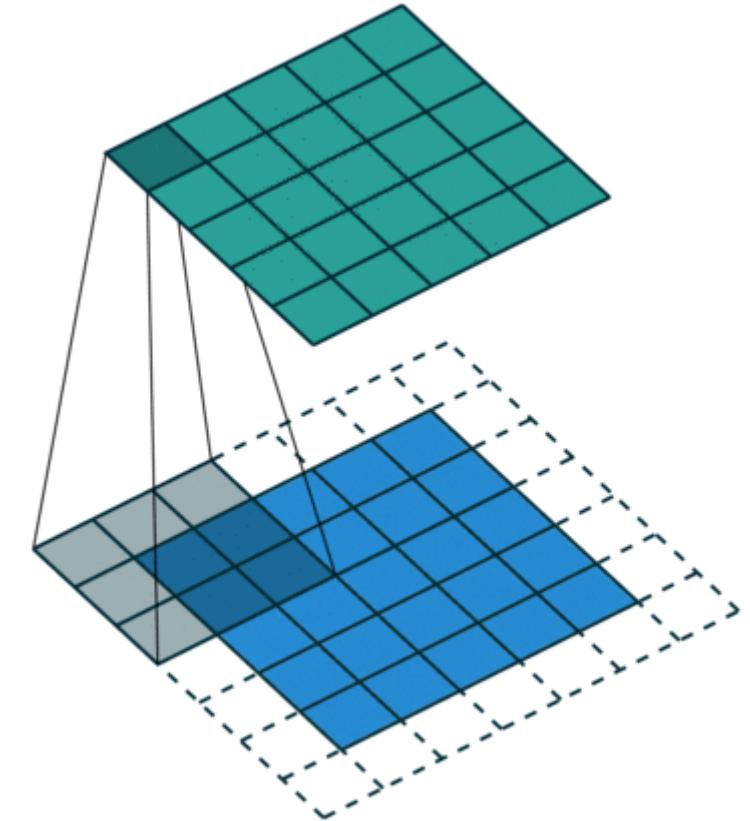
- To prevent the transformation from down-sampling (reducing the size of the matrix during convolution to output), we can pad the edges of the image with 0's.



Padding Technique

0	0	0	0	0	0	0	0
0							0
0							0
0							0
0							0
0							0
0							0
0	0	0	0	0	0	0	0

Zero-padding added to image



Add appropriate padding to keep the dimension after filtering

$n \times n$ input image with $f \times f$ filters

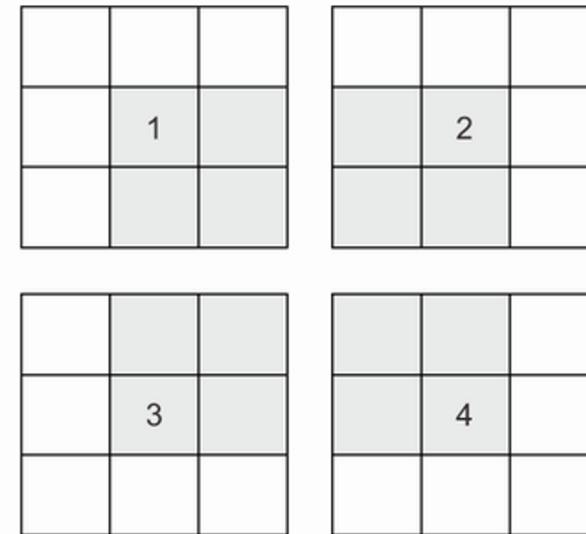
add $p = (f - 1)/2$ padding keep the same dimension of input image

Strides

Strides

- Often, we will pass the filter over every pixel cell, but we don't have to; we might pass over every other cell. This is what strides refers to (skipping).

	1		2	
	3		4	



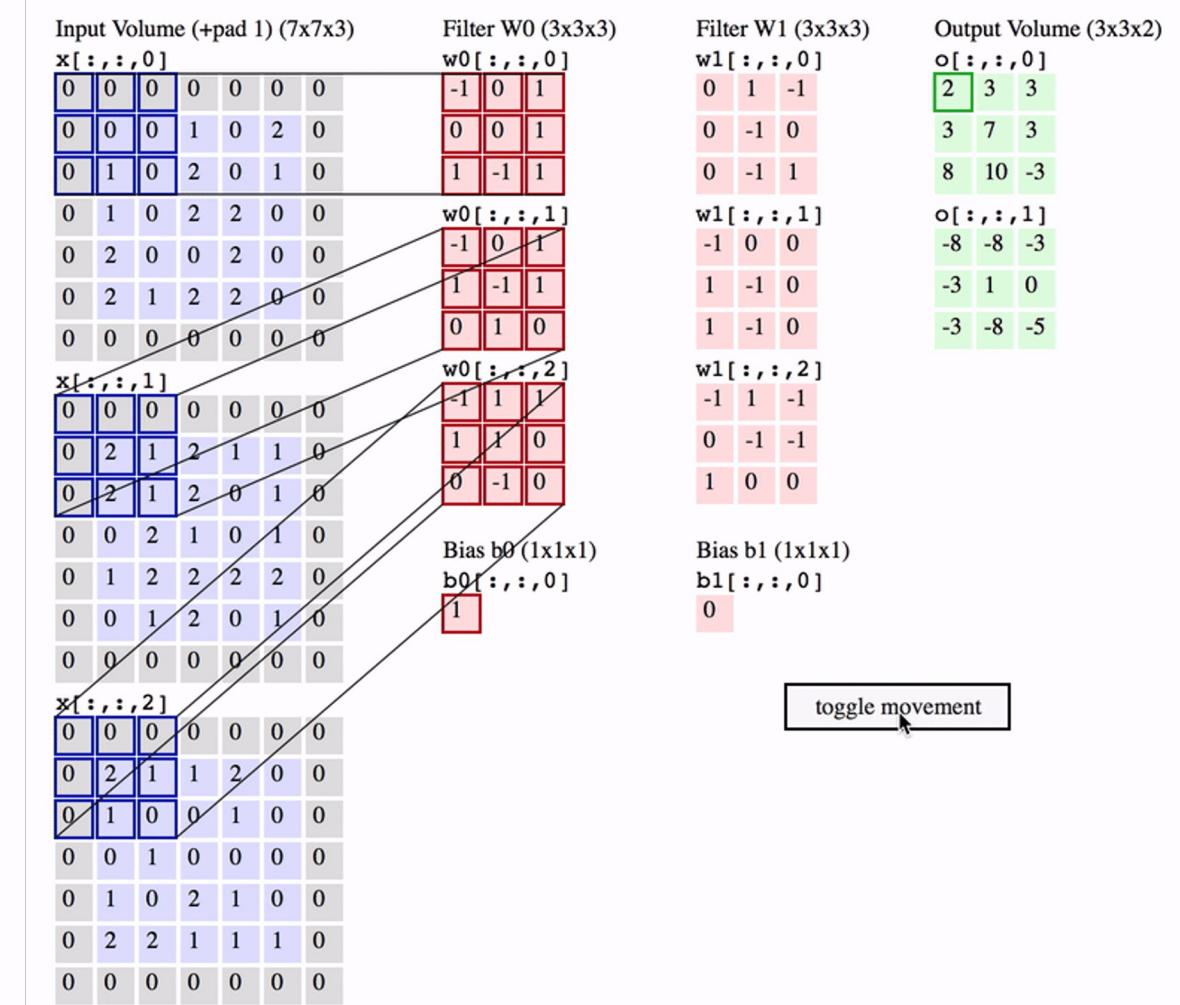
Padding and Strides

Padding

- To prevent the transformation from down-sampling (reducing the size of the matrix during convolution to output), we can pad the edges of the image with 0's.

Strides

- Often, we will pass the filter over every pixel cell, but we don't have to; we might pass over every other cell.
- This is what strides refers to (skipping).



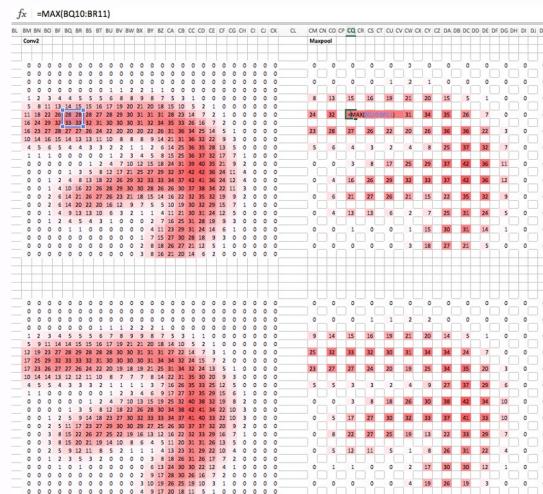
What is Pooling?

Down-sampling Detected Features

- The idea is to compress the resulting data down into a coarser representation, to reduce model complexity, and to also force attention toward a broader section of the original image (helps reduce overfitting).

Forcing Attention to Larger Blocks of the Original Image

- Because we typically use stride = pool width, the pooling output is aggregating over segments of the input.



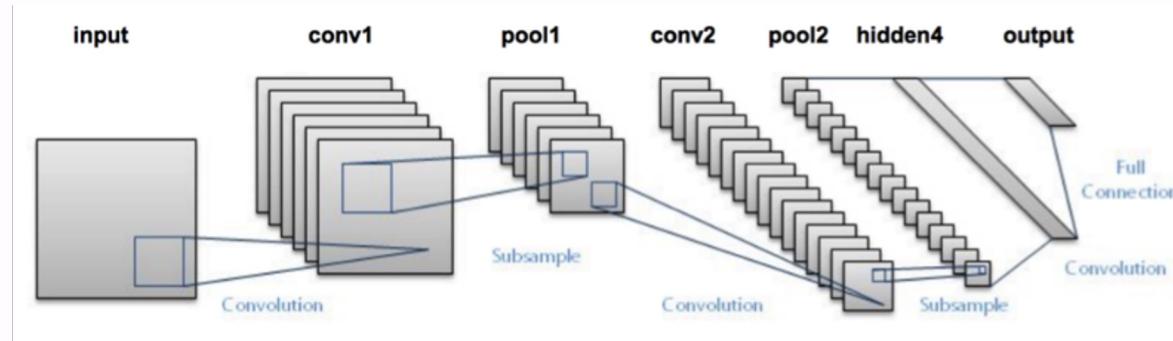
Basic Image Labeling Topology

Convolution Layers Apply (Multiple) Filters

- Each filter is a matrix of trainable weights.
- As you move through the network, the number of features typically rises exponentially.
- More filters as you move along means it allows more permutations / combinations

Progressively Smaller Filter Maps

- Smaller filter map arises from the pooling steps, which means that each element of the final map distills features (high level features, derived from low level features, derived from raw pixels) derived from a larger segment of the original picture.



ImageNet Challenge

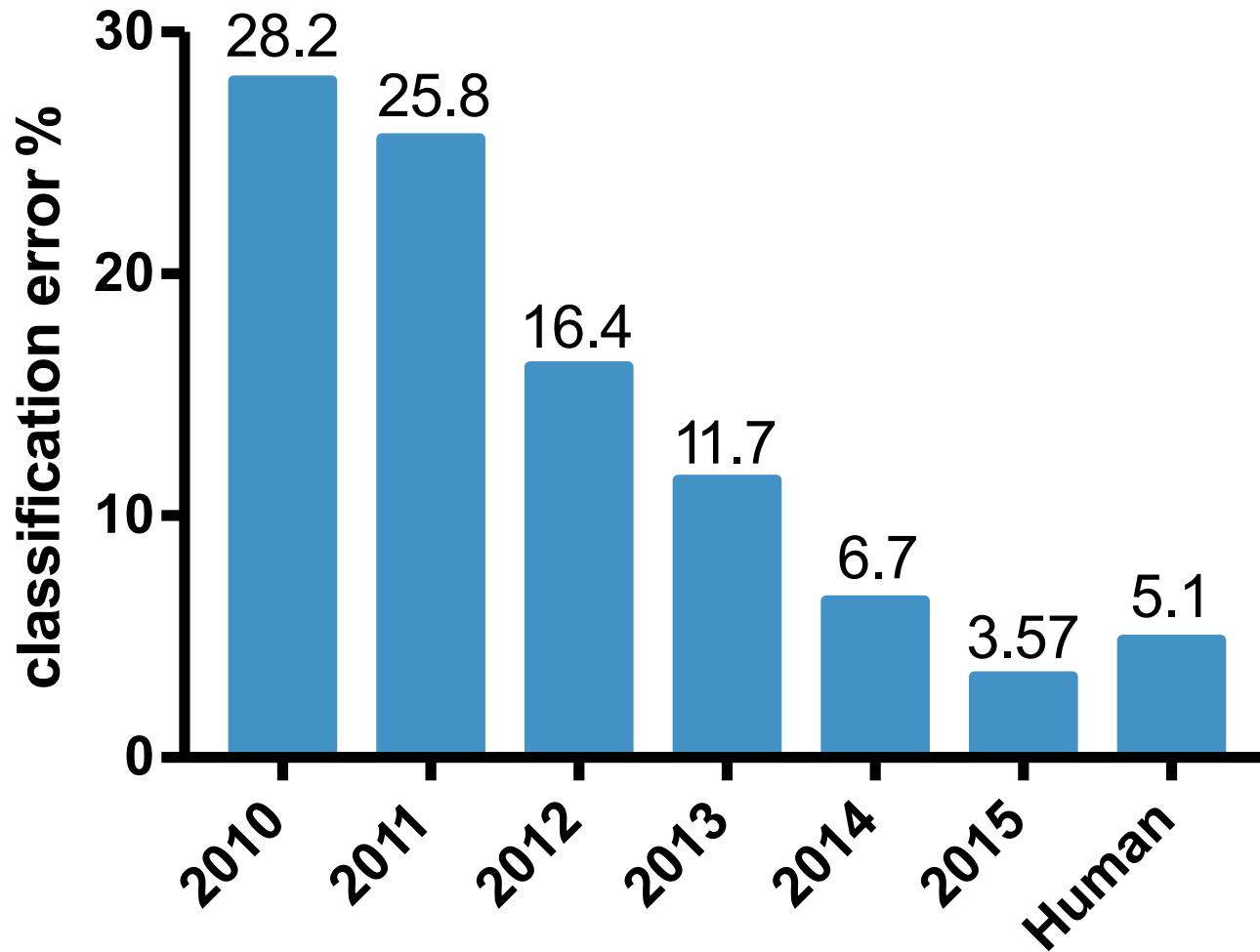


ImageNet Large Scale Visual Recognition Challenges



Classification task: “Top 5 error among 1000 categories”: rate at which the model does not output correct label in top 5 predictions

ImageNet Challenge: Classification Task



2012: AlexNet. First CNN to win.

- 8 layers, 61 million parameters

2013: ZFNet

- 8 layers, more filters

2014: VGG

- 19 layers

2014: GoogLeNet

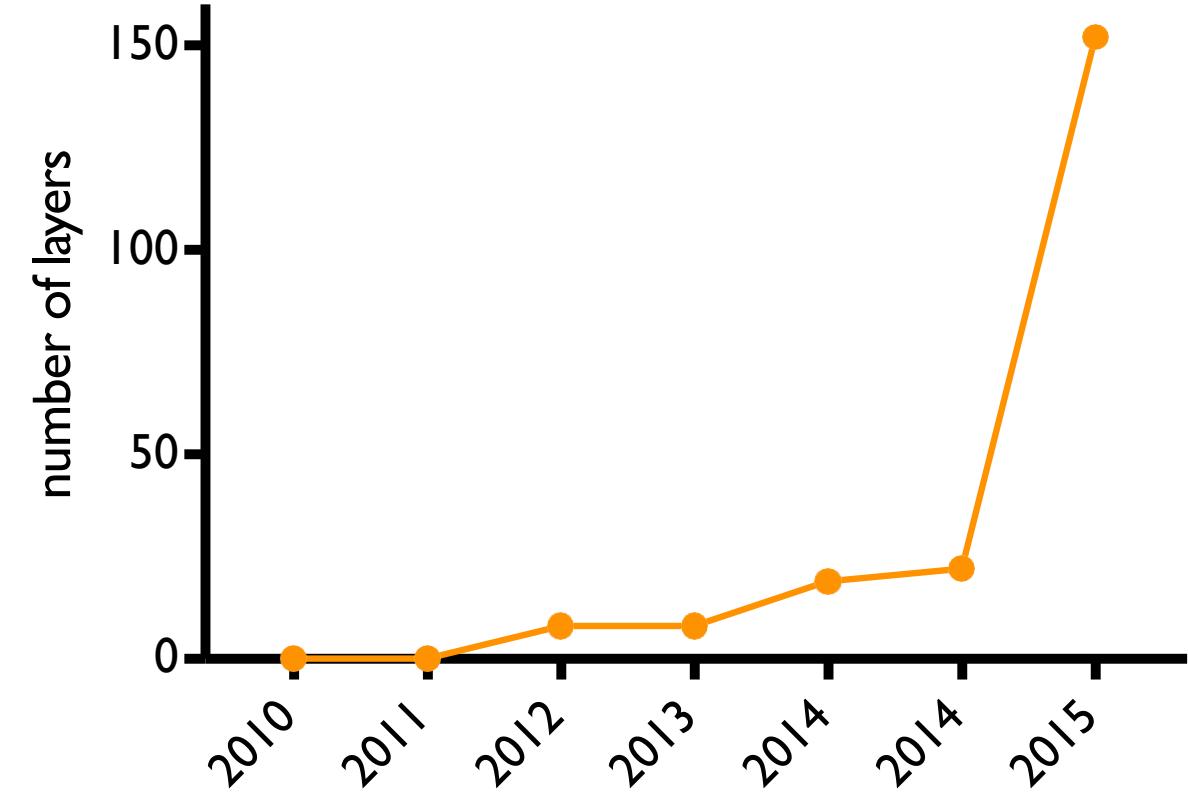
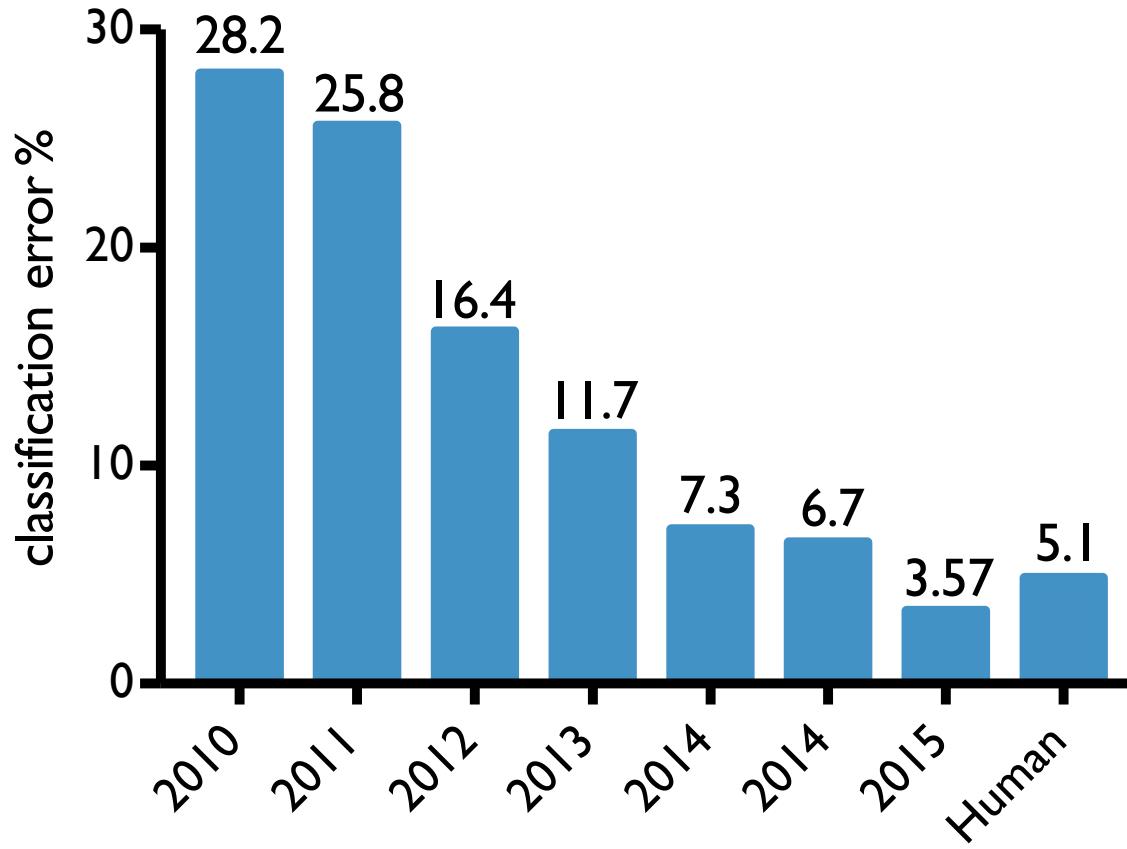
- “Inception” modules

- 22 layers, 5 million parameters

2015: ResNet

- 152 layers

ImageNet Challenge: Classification Task



Computer Vision

6 big tasks

1. *Image and video classification*
2. Object detection
3. Semantic segmentation
4. Motion detection
5. Scene text recognition
6. Visual Q&A



Smartphones

SenselD mobile face unlock solution
Emoticon recognition

Smart cars

License plate recognition
Gesture recognition
Road condition detection

Entertainment

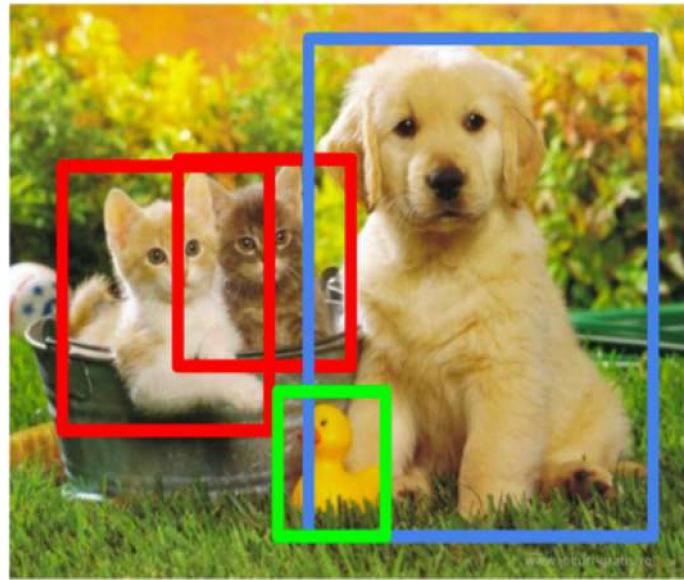
Smart Health

Food testing
CT analysis

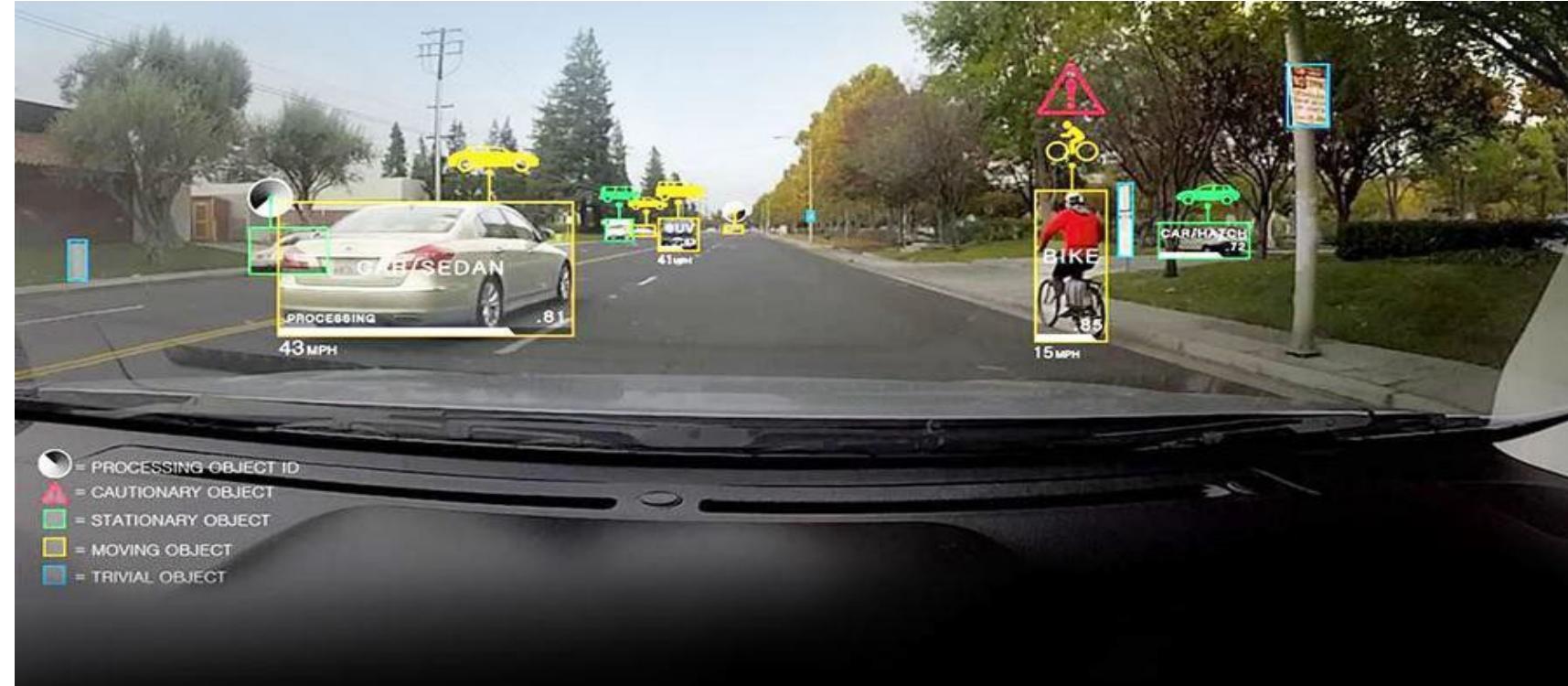
Enterprise business

Security system

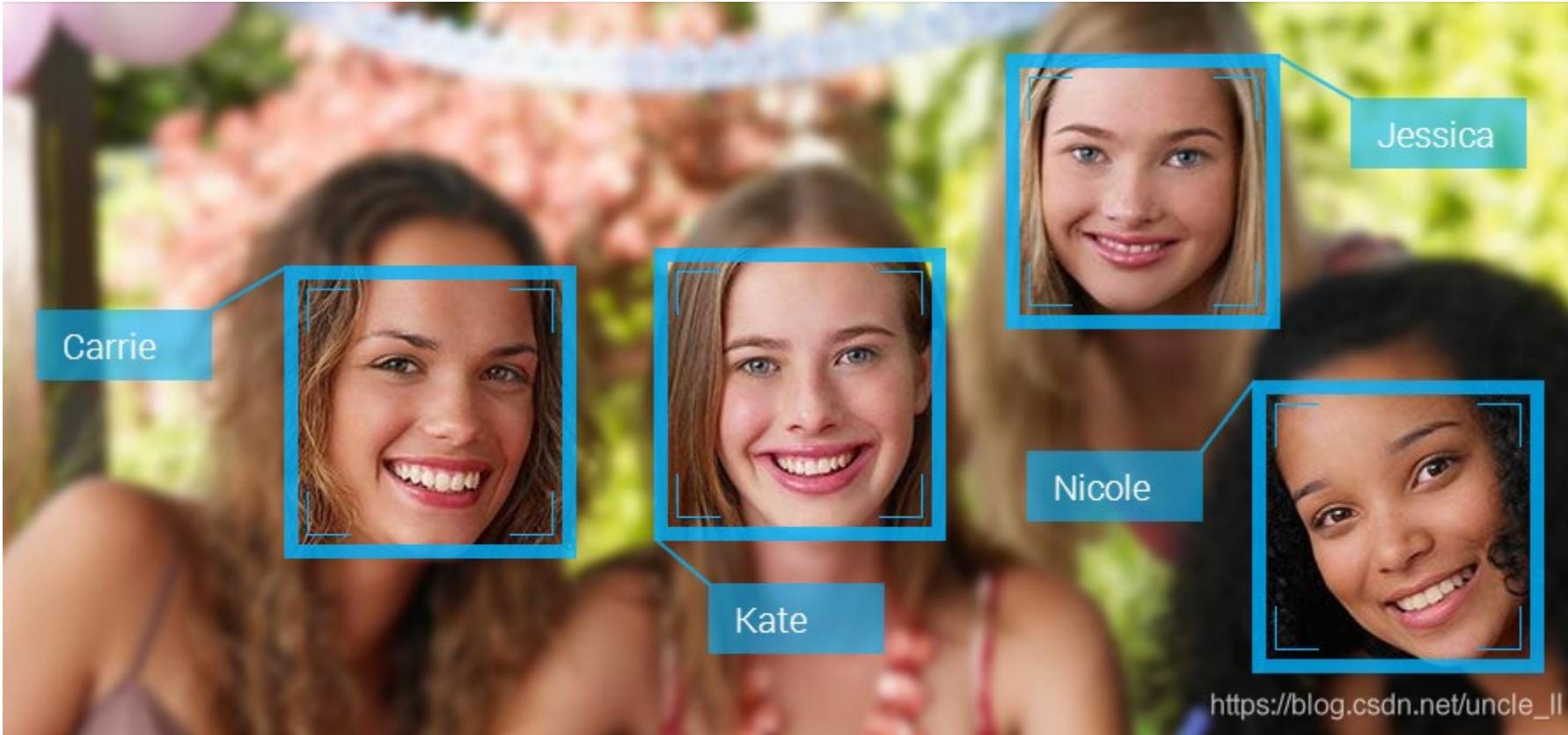
Task 2. Object Detection: The Foundation of Unmanned Vehicles



CAT, DOG,
DUCK



Face detection: Phone face unlock



Face detection + emotion classification



A photograph of a man holding a young girl. Both faces are framed by colored boxes indicating detected emotions: a yellow box surrounds the girl's face with the word "Surprise", and a teal box surrounds the man's face with the word "Surprise".

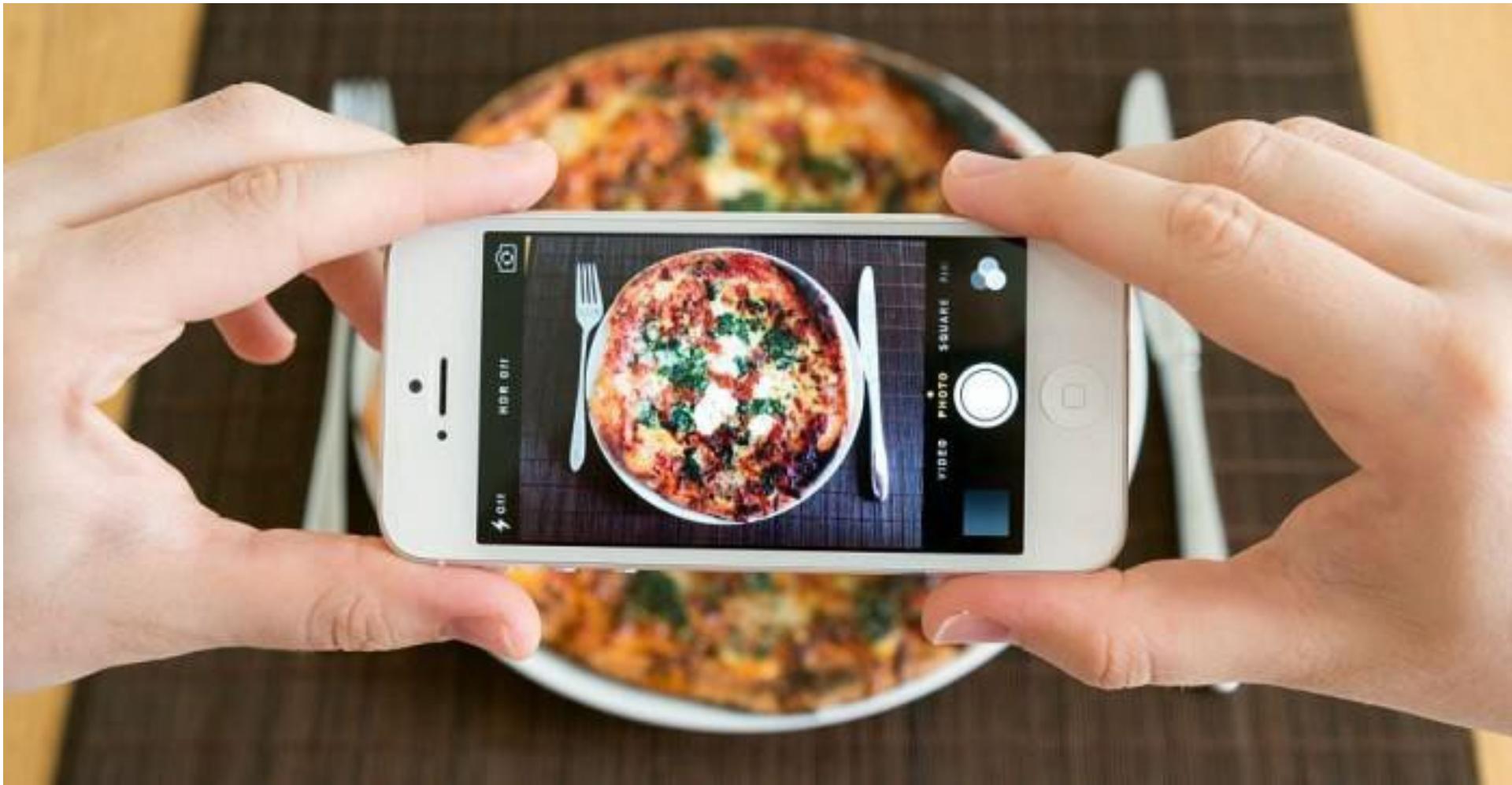
Neutral:	██████████	Anger:	██████████
Happiness:	██████████	Disgust:	██████████
Surprise:	██████████	Fear:	██████████
Sadness:	██████████	Contempt:	██████████



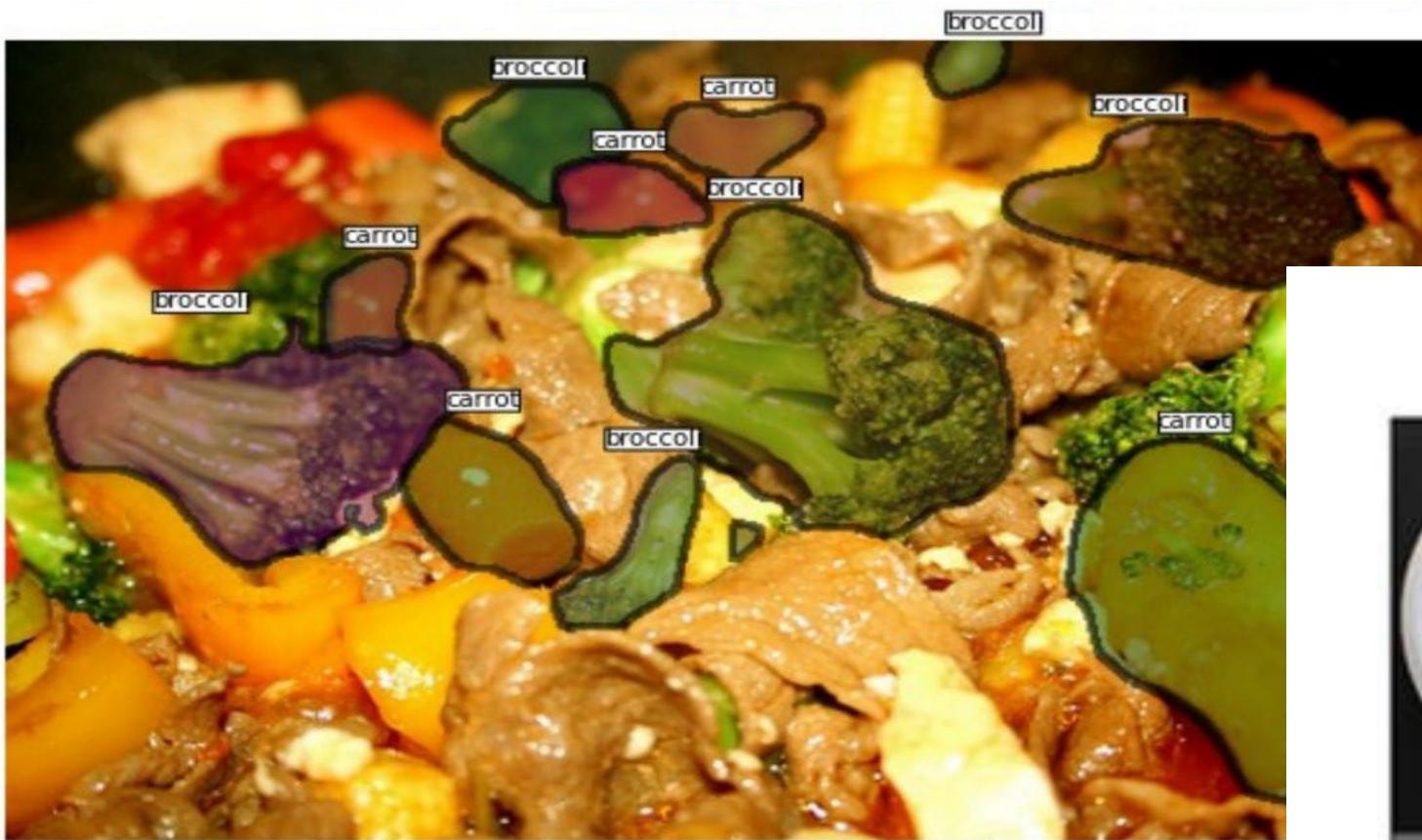
 Microsoft

Get started for free at projectoxford.ai

Food semantic segmentation + nutrient content analysis



Food semantic segmentation + nutrient content analysis



Segmentation:



Portion & Nutrition Estimation:

Sandwich: 146g
French Fries: 85g
Strawberries: 114g

Nutrition Facts	
Serving Size 1 (1g)	
Serving Per Container 1	
Amount Per Serving	
Calories	623
% Daily Value*	
Total Fat 43g	66%
Saturated Fat 0g	0%
Trans Fat 0g	0%
Sodium 0mg	0%
Total Carbohydrate 102g	34%
Dietary Fiber 0g	0%
Sugars 6g	
Protein 10g	20%

Motion Detection



The ActivityNet logo features a red square with a white silhouette of a person performing a dynamic action, followed by the text "ACTIVITYNET" in a bold, sans-serif font.

Peeling Potatoes

Playing Badminton

Polishing Shoes

Shoveling Snow

Horse Riding

Vacuuming Floor

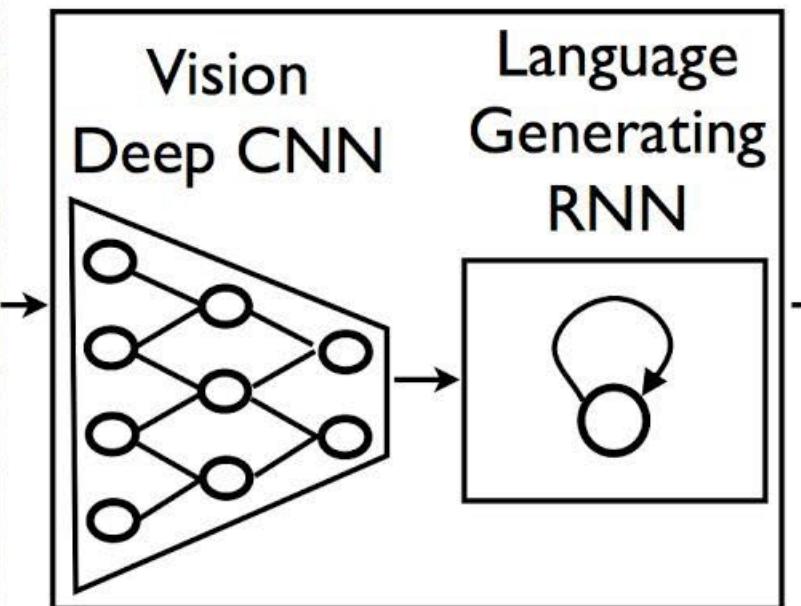
This section displays a grid of six images, each representing a different activity: peeling potatoes, playing badminton, polishing shoes, shoveling snow, horse riding, and vacuuming a floor. Each image is labeled with its corresponding activity name.

Scene text recognition (Document Parsing)



Task 6. Machine Vision + Natural Language Processing

Generate representations based on images



A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

Visual Question and Answer (VQA) challenge



What vegetable is on the plate?
Neural Net: **broccoli**
Ground Truth: broccoli



What color are the shoes on the person's feet ?
Neural Net: **brown**
Ground Truth: brown



How many school busses are there?
Neural Net: **2**
Ground Truth: 2



What sport is this?
Neural Net: **baseball**
Ground Truth: baseball



What is on top of the refrigerator?
Neural Net: **magnets**
Ground Truth: cereal



What uniform is she wearing?
Neural Net: **shorts**
Ground Truth: girl scout



What is the table number?
Neural Net: **4**
Ground Truth: 40

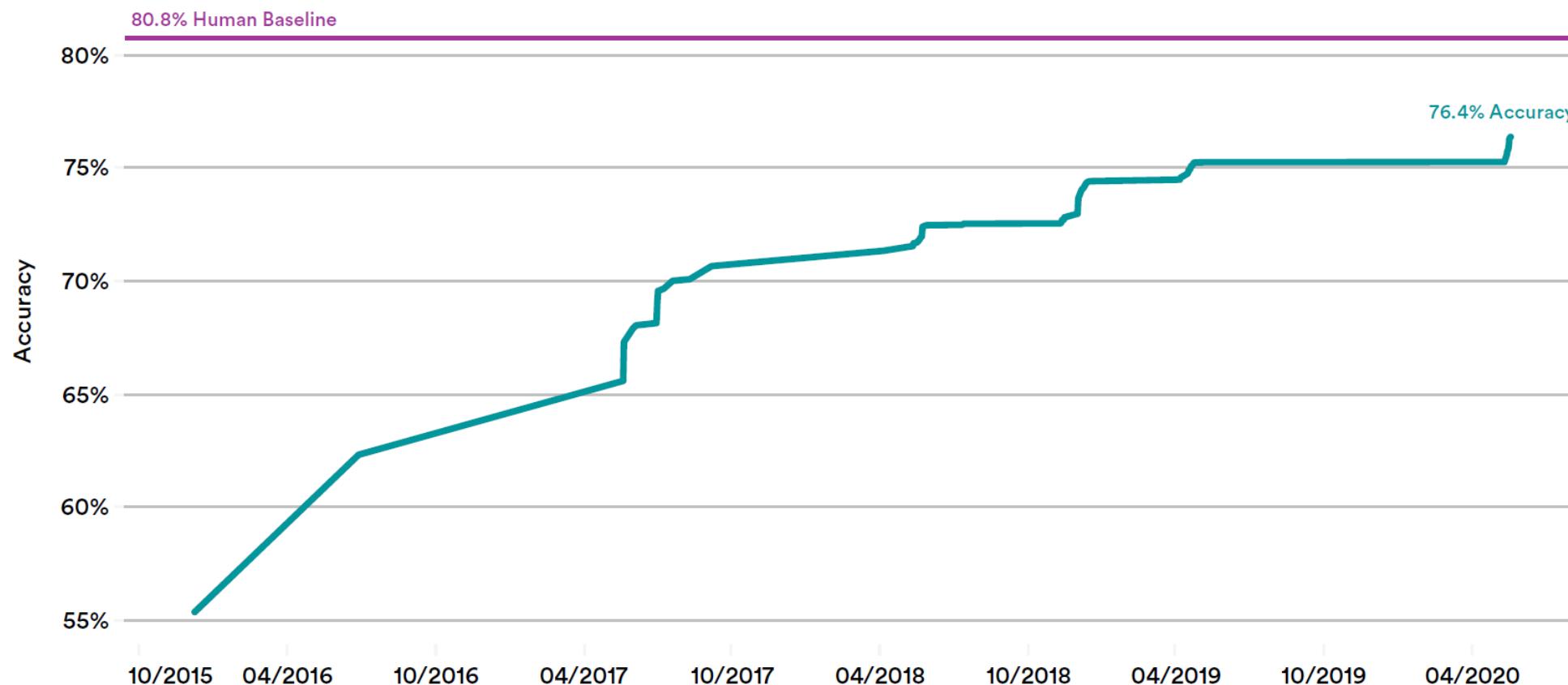


What are people sitting under in the back?
Neural Net: **bench**
Ground Truth: tent

Visual Question and Answer (VQA) challenge

VISUAL QUESTION ANSWERING (VQA) CHALLENGE: ACCURACY

Source: VQA Challenge, 2020 | Chart: 2021 AI Index Report



Deep Learning for Computer Vision: Summary

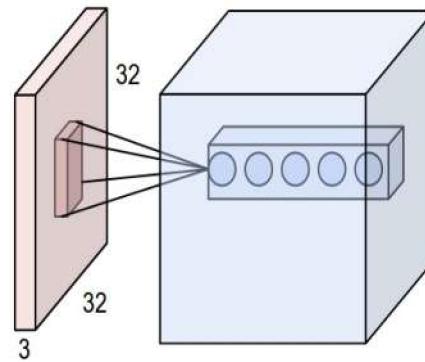
Foundations

- What is computer vision?
- Representing images
- Convolutions for feature extraction



CNNs

- CNN architecture
- Application to classification



Applications

- Object detection
- Object segmentation
- Motion detection
- Scene text recognition
- Visual Q&A

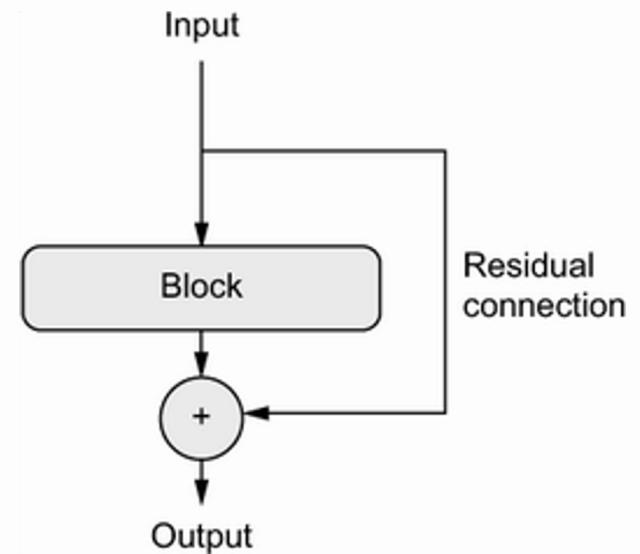


Residual Connections

Construct Repeating ‘Blocks’ of Layers

- Convolution, Pooling, Residual Connection, Convolution, Pooling, Residual Connection...
- In deep networks, vanishing gradient problem arises because there is noise in every step, and if we go far enough back the noise overwhelms the signal. The residual connections help ensure we have more signal than noise.

```
1 inputs = keras.Input(shape=(32, 32, 3))
2 x = layers.Rescaling(1./255)(inputs)
3
4 def residual_block(x, filters, pooling=False):
5     residual = x
6     x = layers.Conv2D(filters, 3, activation="relu", padding="same")(x)
7     x = layers.Conv2D(filters, 3, activation="relu", padding="same")(x)
8     if pooling:
9         x = layers.MaxPooling2D(2, padding="same")(x)
10        residual = layers.Conv2D(filters, 1, strides=2)(residual)
11    elif filters != residual.shape[-1]:
12        residual = layers.Conv2D(filters, 1)(residual)
13    x = layers.add([x, residual])
14    return x
15
16 x = residual_block(x, filters=32, pooling=True)
17 x = residual_block(x, filters=64, pooling=True)
18 x = residual_block(x, filters=128, pooling=False)
19
20 x = layers.GlobalAveragePooling2D()(x)
21 outputs = layers.Dense(1, activation="sigmoid")(x)
22 model = keras.Model(inputs=inputs, outputs=outputs)
23 model.summary()
```

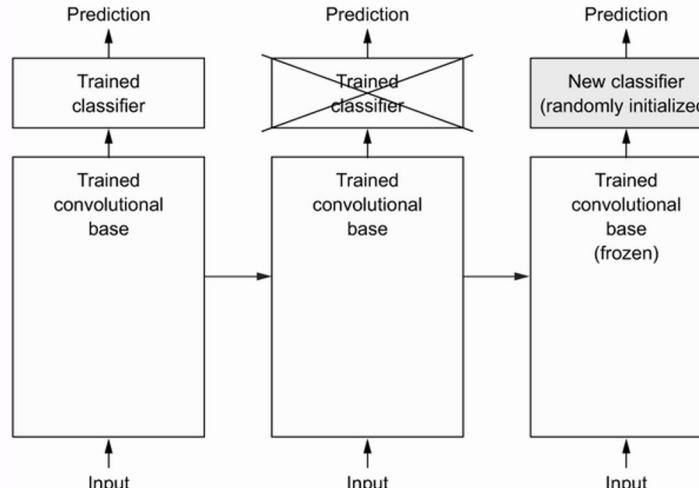


Pre-Trained Models: Feature Extraction

Take the convolutional base layers from someone else's model, then...

Two Options

- Feed Data Through Model Base: feed your images through convolutional base, take the outputs, and then use those as your predictors, feeding them into a network of dense layers.
- Freeze Model Base and Include in Network: Take the convolutional base layers from someone else's model and freeze them (make parameters non-trainable), then stack your (trainable) Dense layers onto the end. This lets you add data-augmentation to the front of the model.



Pre-Trained Models: Fine Tuning

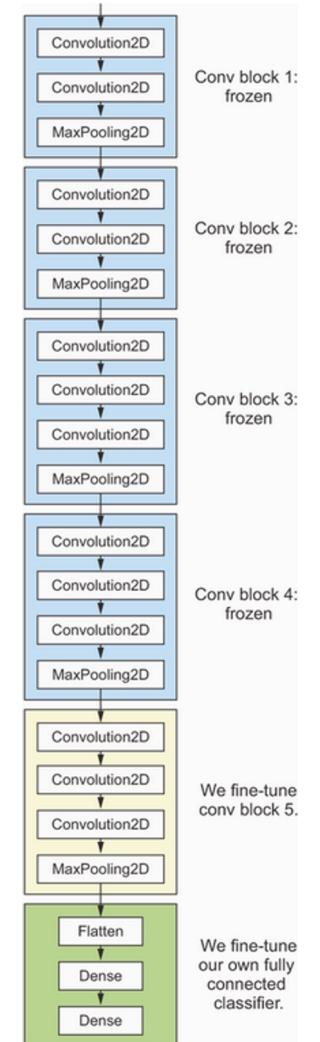
Take the convolutional base layers from someone else's model, then...

Freeze Only the First Several Layers

- Allow your network to modify / update the last few convolutional base layers as part of training, along with your own Dense layers...
- Iterate over the layers in the network and set the last few to be trainable.

Listing 8.27 Freezing all layers until the fourth from the last

```
1 conv_base.trainable = True
2 for layer in conv_base.layers[:-4]:
3     layer.trainable = False
```



Questions?