Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 2.1: Unimodal Representations

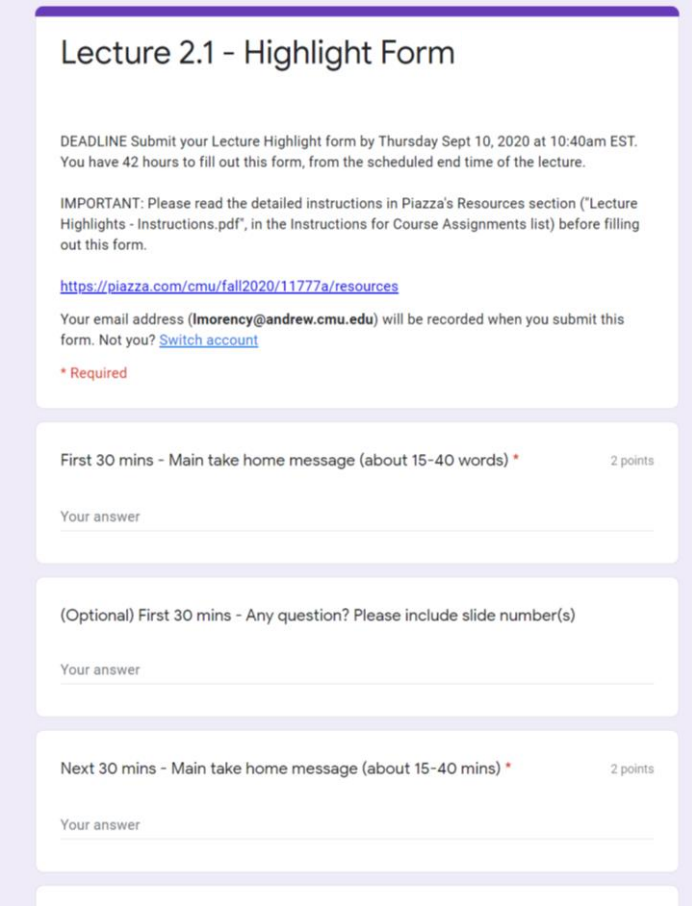
Louis-Philippe Morency

** Co-lecturer: Paul Liang. Original course co-developed with Tadas Baltrusaitis. Spring 2021 and 2022 editions taught by Yonatan Bisk*

Administrative Stuff

Lecture Highlight Form

https://forms.gle/g6zovyaK2QwUvXW17



Deadline: Tuesday 8pm ET

(for Thursday's lecture, the deadline is Thursday 8pm ET)

Use your Andrew CMU email

You will need to login using this address

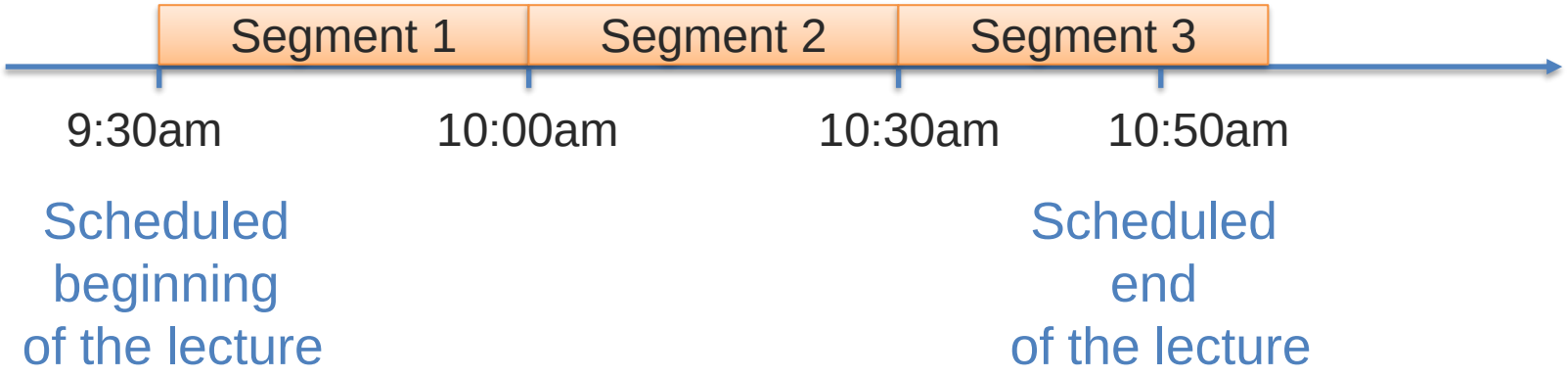
New form for each lecture

Posted on Piazza's Resources section

Ask questions about the lecture

- Will be answered either online or at the next lecture

Lecture Highlight Form - Segments



- Segment 1 starts at 9:30am, even if the lecture starts slightly later.
- Segment 3 ends whenever the lecture ends
- Slides happening around the segment borders (+/- 5min of 10:00am and 10:30am) can be included in either neighboring segment.

Lecture Highlight Form - Grading

For each segment

- Two sentences (10+ words each; complete English sentences) describing two main points described in this segment

For the whole lecture

- Your main two take-aways from the lecture
  - 10+ words each; complete English sentences
- Be as concrete as possible in your take-home messages
  - Avoid generic summaries like: “This is about multimodal”

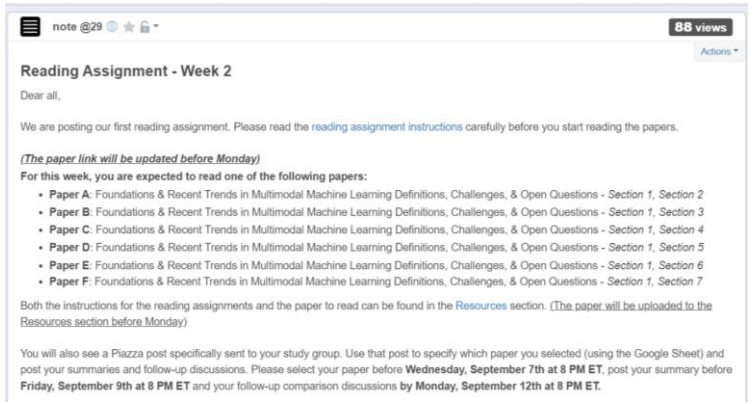
Each submission is worth 1 point

- Final grade is the sum of your top 16 submissions

Reading Assignments – Piazza Posts

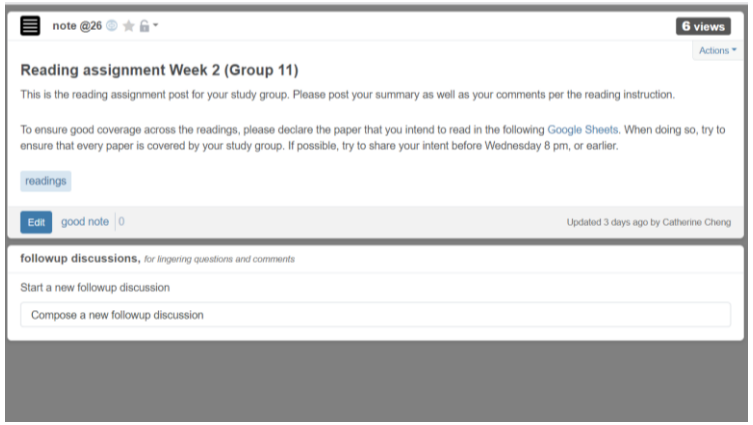
For each reading assignment, 2 instruction posts will be created:

1



- Sent to everyone
- Contains list of reading options

2

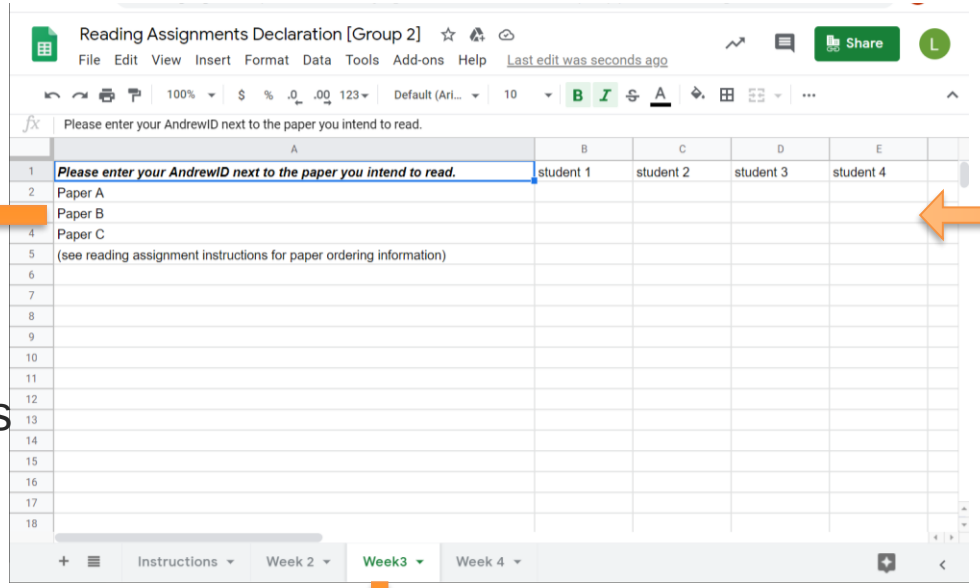


- Sent separately to each study group
- Link to personalized signup sheet
- Post your summary as top-level
- Post your follow-up posts

Reading Assignments – Signup Sheet

Each study group has its own signup sheet:

Sign-up here for the paper option you would like to read and summarize



The details for the paper options are in the first Piazza post

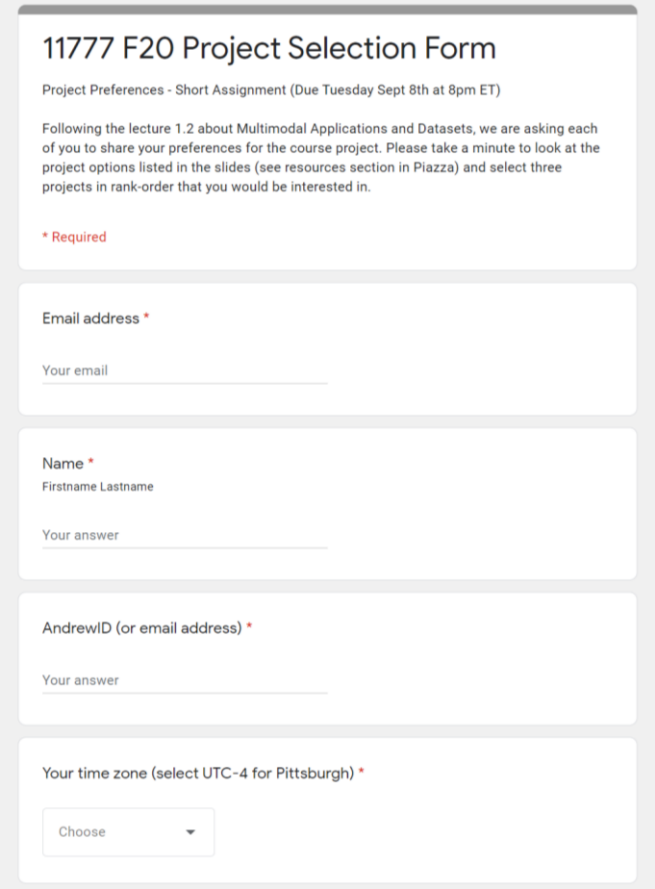
A different tab for each reading assignment

Reading Assignments – Weekly Schedule

Four main steps for the reading assignments

1. **Monday 8pm:** Official start of the assignment
2. **Wednesday 8pm:** Select your paper
3. **Friday 8pm:** Post your summary
4. **Monday 8pm:** End of the reading assignment

Team Matching – Project Preference Form



11777 F20 Project Selection Form

Project Preferences - Short Assignment (Due Tuesday Sept 8th at 8pm ET)

Following the lecture 1.2 about Multimodal Applications and Datasets, we are asking each of you to share your preferences for the course project. Please take a minute to look at the project options listed in the slides (see resources section in Piazza) and select three projects in rank-order that you would be interested in.

* Required

Email address *

Your email

Name *

Firstname Lastname

Your answer

AndrewID (or email address) *

Your answer

Your time zone (select UTC-4 for Pittsburgh) *

Choose

Deadline: Today at 8pm!!

- Every students should submit a form
- Students on the waitlist are also encouraged to submit a form
- A summary will be shared to help you find potential teammates

Team Matching – Thursday Event

Thursday around 10:30am ET

(later part of the lecture)

- Detailed instructions will be shared during lecture
- Event optional for students who already have a full team

Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 2.1: Unimodal Representations

Louis-Philippe Morency

** Co-lecturer: Paul Liang. Original course co-developed with Tadas Baltrusaitis. Spring 2021 and 2022 editions taught by Yonatan Bisk*

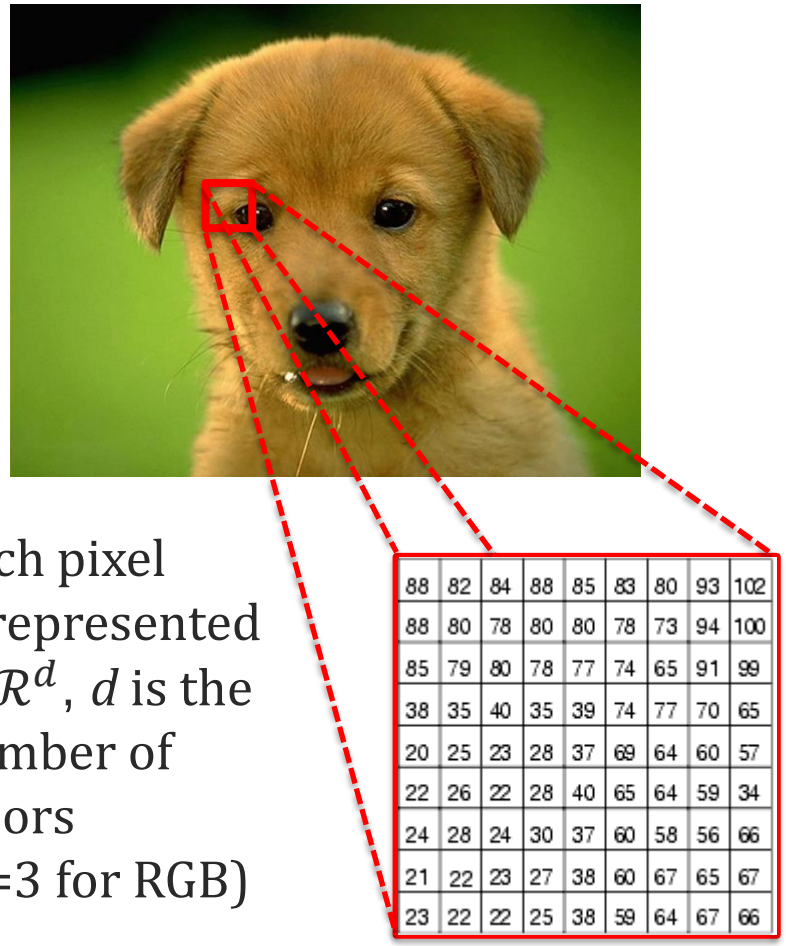
Lecture Objectives

- Unimodal basic representations
- Dimension of heterogeneity
- Image representations
  - Image gradients, edges, kernels
- Convolution neural network (CNN)
  - Convolution and pooling layers
- Visualizing CNNs
- Region-based CNNs

Unimodal Basic Representations

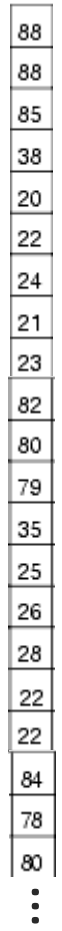
Unimodal Representation – Visual Modality

Color image



Each pixel is represented in $\mathcal{R}^d$, d is the number of colors (d=3 for RGB)

Input observation x_i

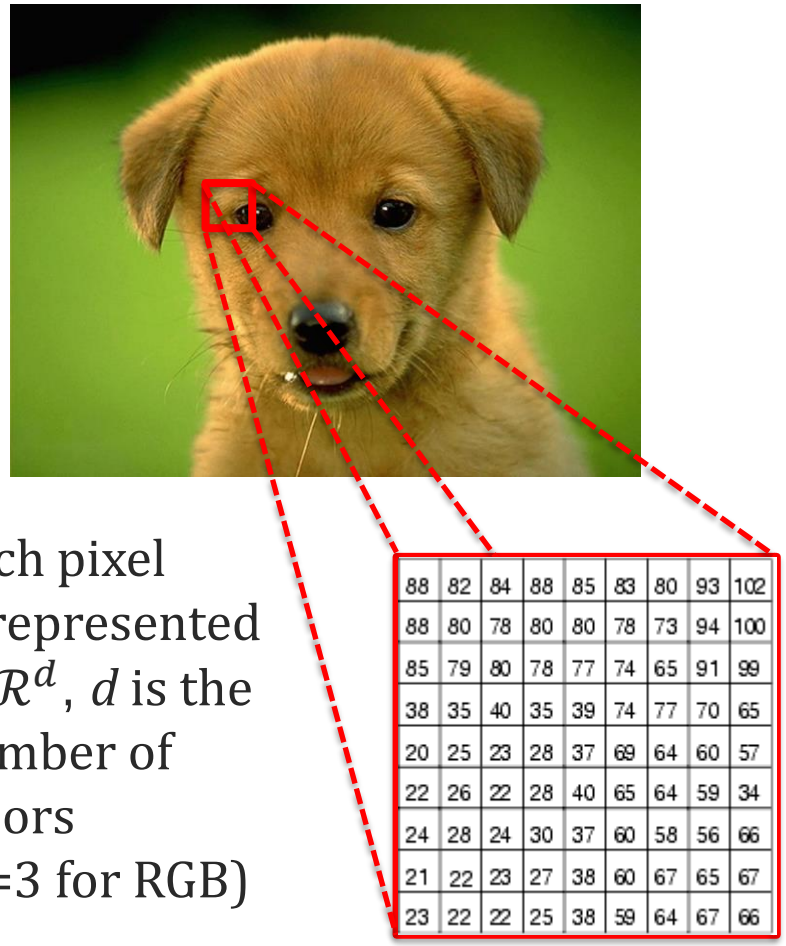


Binary classification problem

Dog ?

label $y_i \in \mathcal{Y} = \{0,1\}$

Unimodal Representation – Visual Modality



Each pixel is represented in $\mathcal{R}^d$, d is the number of colors (d=3 for RGB)

Input observation x_i

88
88
85
38
20
22
24
21
23
82
80
79
35
25
26
28
22
22
84
78
80
⋮

Multi-class classification problem

Duck
-or-
Cat
-or-
Dog
-or-
Pig
-or-
Bird ?

label $y_i \in \mathcal{Y} = \{0,1,2,3, \dots\}$

Unimodal Representation – Language Modality

Written language

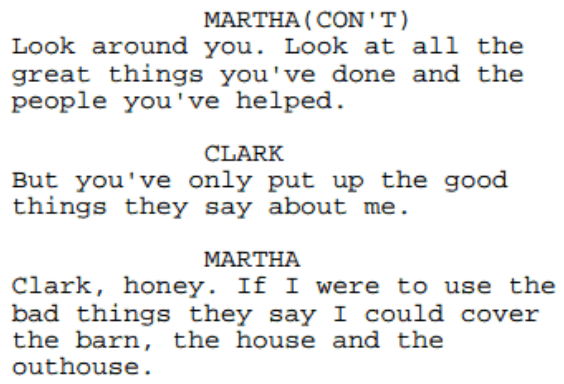
★★★★★ **Masterful!**

By Antony Witheyman - January 12, 2006

Ideal for anyone with an interest in disguises who likes to see the subject tackled in a humourous manner.

0 of 4 people found this review helpful

Spoken language



MARTHA(CON'T)
Look around you. Look at all the great things you've done and the people you've helped.

CLARK
But you've only put up the good things they say about me.

MARTHA
Clark, honey. If I were to use the bad things they say I could cover the barn, the house and the outhouse.

Input observation x_i

"one-hot" vector

$|x_i|$ = number of words in dictionary

Word-level classification

Part-of-speech ?
(noun, verb,…)

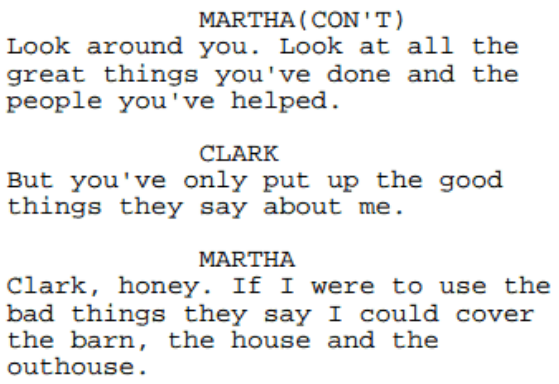
Sentiment ?
(positive or negative)

Named entity ?
(names of person,…)

Unimodal Representation – Language Modality

Written language

Spoken language

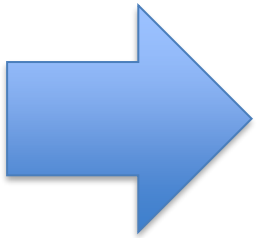


Input observation x_i

"bag-of-word" vector

$|x_i|$ = number of words in dictionary

Document-level classification

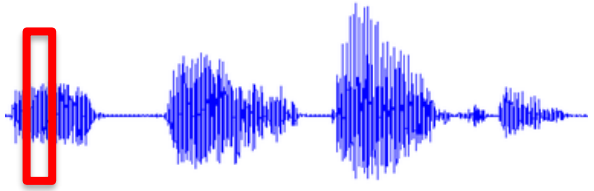


Sentiment ?
(positive or negative)

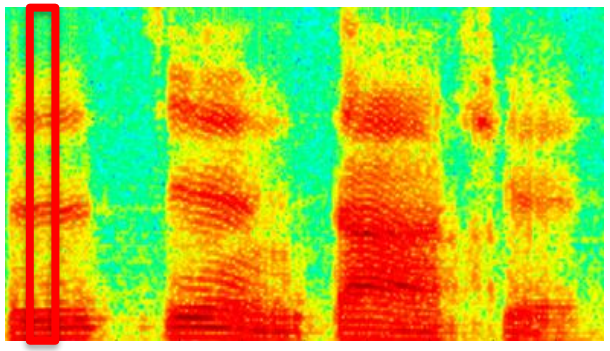
What happens with word ordering?

Unimodal Representation – Acoustic Modality

Digitalized acoustic signal



- Sampling rates: 8~96kHz
- Bit depth: 8, 16 or 24 bits
- Time window size: 20ms
  - Offset: 10ms



Spectogram

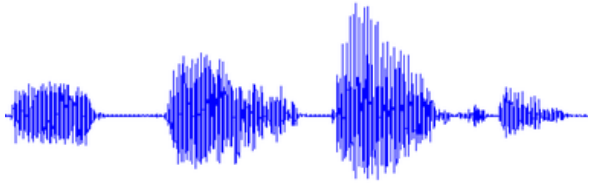
Input observation x_i

0.21
0.14
0.56
0.45
0.9
0.98
0.75
0.34
0.24
0.11
0.02

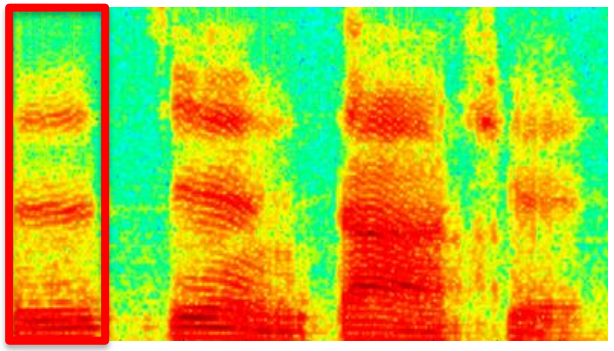
Spoken word ?

Unimodal Representation – Acoustic Modality

Digitalized acoustic signal



- Sampling rates: 8~96kHz
- Bit depth: 8, 16 or 24 bits
- Time window size: 20ms
  - Offset: 10ms



Spectogram

Input observation x_i

0.21
0.14
0.56
0.45
0.9
0.98
0.75
0.34
0.24
0.11
0.02
0.24
0.26
0.58
0.9
0.99
0.79
0.45
0.34
0.24
⋮

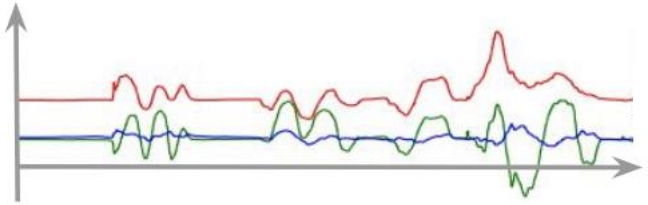
Emotion ?

Spoken word ?

Voice quality ?

What invariance naturally exists in acoustic signals?

Unimodal Representation – Sensors



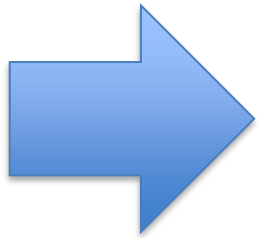
Force-Torque Sensor

Time series data across six-axis Force-Torque sensor:
T × 6 signal.

Proprioception

Measure values internal to the system (robot); e.g. motor speed, wheel load, **robot arm joint angles**, battery voltage.

Time series data across current position and velocity of the end-effector:
T × 2d signal.



Next action

Lee et al., Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks. ICRA 2019

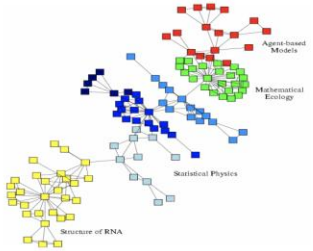
Unimodal Representation – Tables

Table-to-text generation

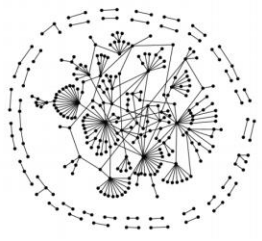
Bao et al., Table-to-Text: Describing Table Region with Natural Language. AAAI 2018

Unimodal Representation – Graphs

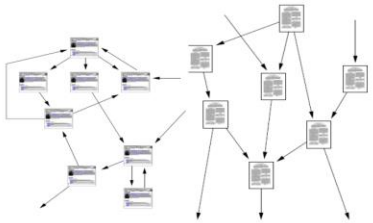
Social networks



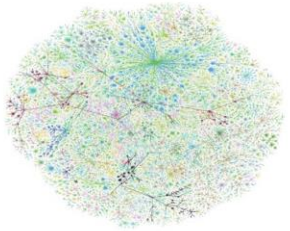
Economic networks



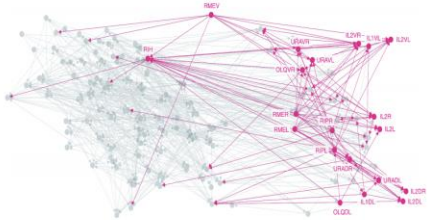
Biomedical networks



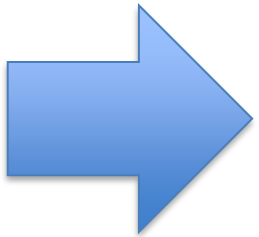
Information networks:
Web & citations



Internet



Networks of neurons



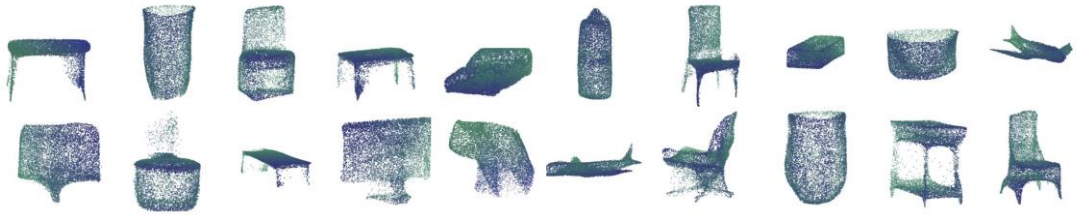
Tasks on graphs:
Node classification
Link prediction
…

Using graphs:
Knowledge graphs
for QA
Social network for
sentiment analysis
…

Hamilton and Tang, Tutorial on Graph Representation Learning. AAAI 2019

Unimodal Representation – Sets

Sets



Point clouds

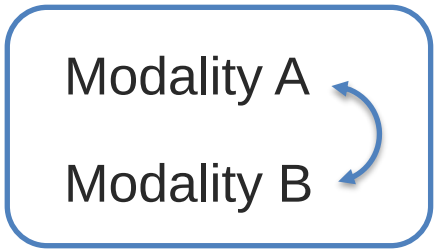
- Set anomaly detection
- Set expansion
- Set completion
- Point cloud classification
- Point cloud generation

Zaheer et al., DeepSets. NeurIPS 2017, Li et al., Point Cloud GAN. arxiv 2018

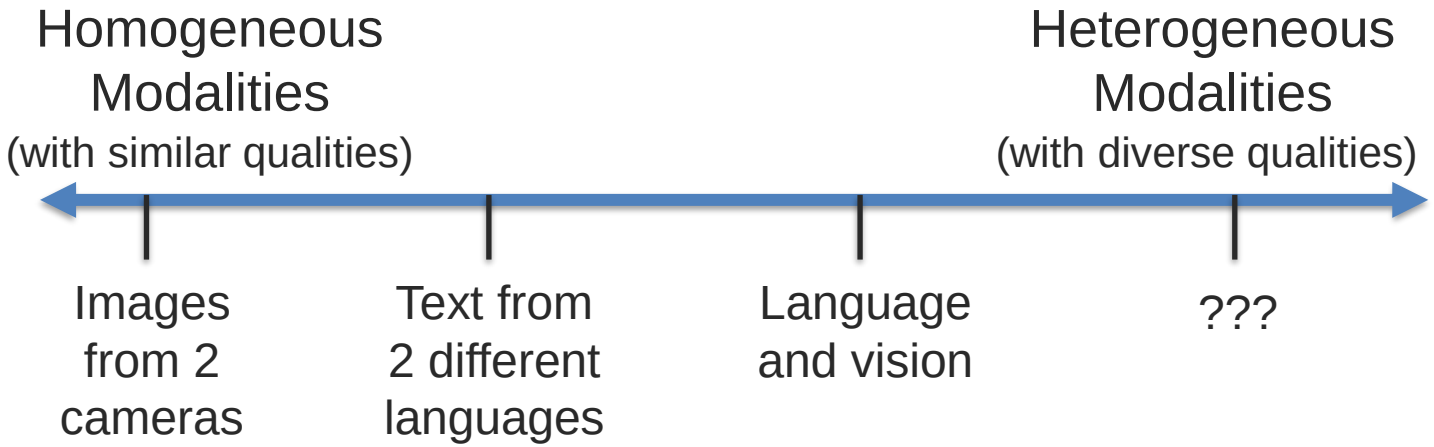
Dimensions of Heterogeneity

Heterogeneous Modalities

Information present in different modalities will often show diverse qualities, structures and representations.



Examples:



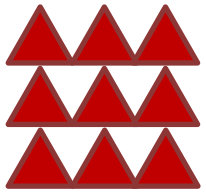
Dimensions of Heterogeneity

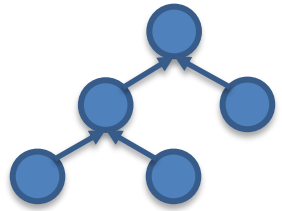
Modality A ⟷ Modality B

1. **Element representations:**
   Discrete, continuous, granularity

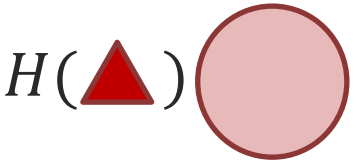
2. **Element distributions:**
   Density, frequency

3. **Structure:**
   Temporal, spatial, latent, explicit





4. **Information:**
   Abstraction, entropy

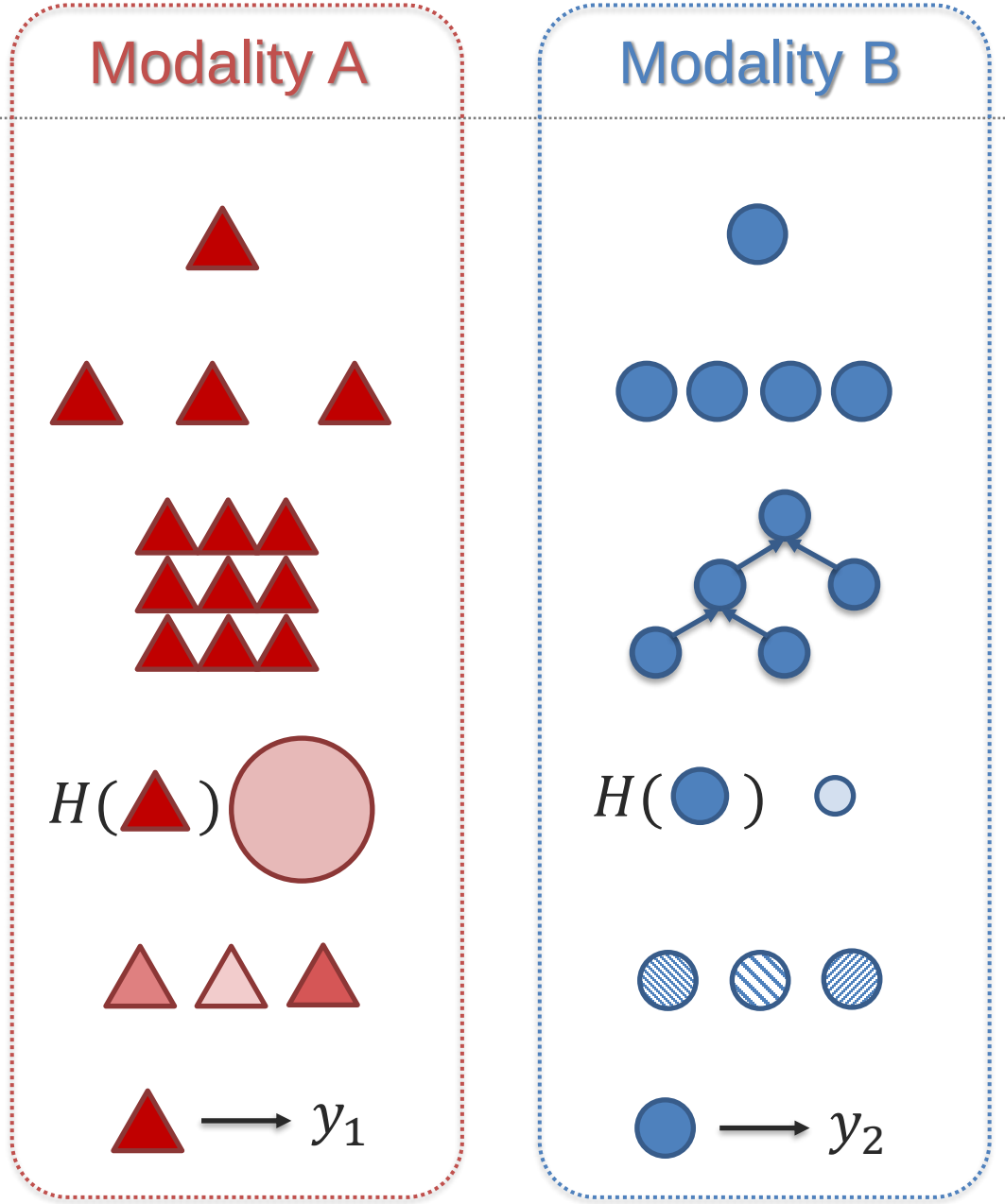


5. **Noise:**
   Uncertainty, noise, missing data

6. **Relevance:**
   Task, context dependence

Modality Profile

1. **Element representations:**
   Discrete, continuous, granularity
2. **Element distributions:**
   Density, frequency
3. **Structure:**
   Temporal, spatial, latent, explicit
4. **Information:**
   Abstraction, entropy
5. **Noise:**
   Uncertainty, noise, missing data
6. **Relevance:**
   Task, context dependence



Modality Profile

1. **Element representations:**
   Discrete, continuous, granularity
2. **Element distributions:**
   Density, frequency
3. **Structure:**
   Temporal, spatial, latent, explicit
4. **Information:**
   Abstraction, entropy
5. **Noise:**
   Uncertainty, noise, missing data
6. **Relevance:**
   Task, context dependence

Visual Image Modality

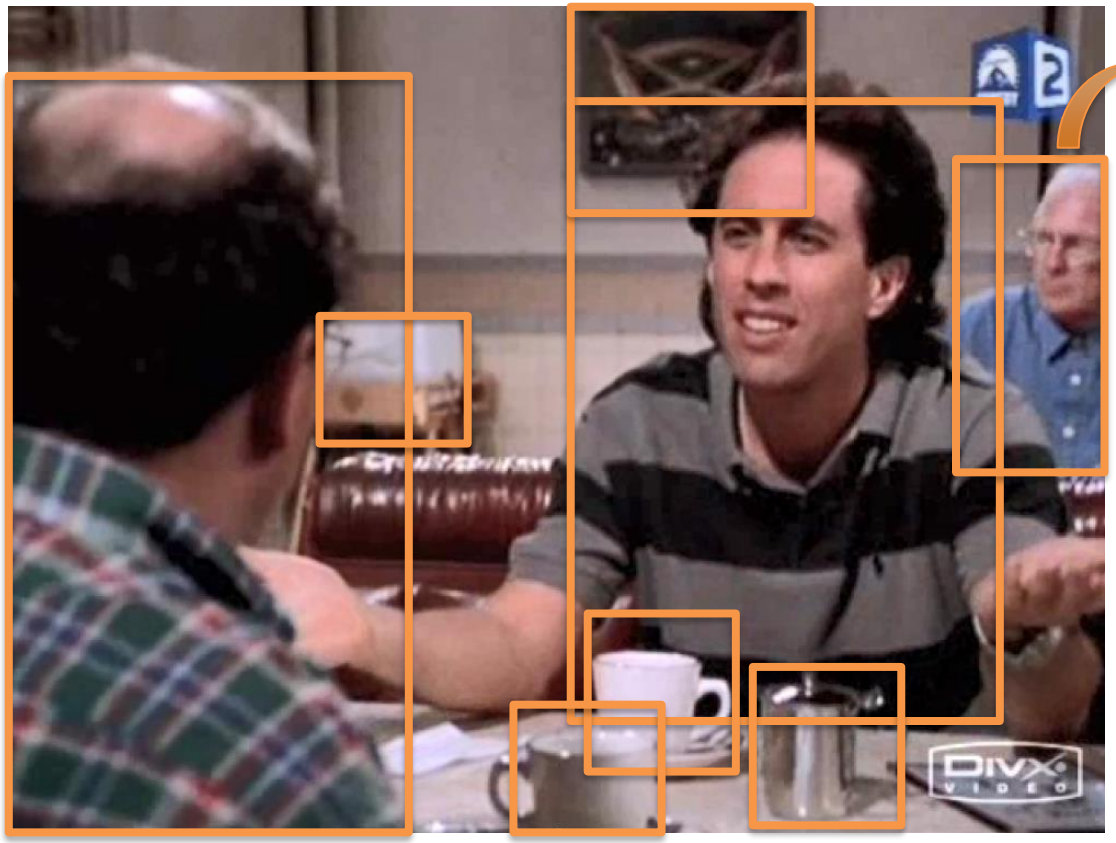
Image Representations

How Would You Describe This Image?

?

88
88
85
38
20
22
24
21
23
82
80
79
35
25
26
28
22
22
84
78
80
⋮

Object-Based Visual Representation



"person" label

+

Appearance descriptor

- ❑ Age
- ❑ Expression
- ❑ Clothes
- ❑ …

Feature vector

Object Descriptors

How to represent and detect an object?

Many approaches over the years…

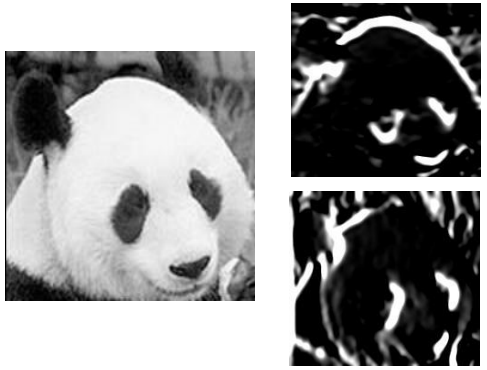
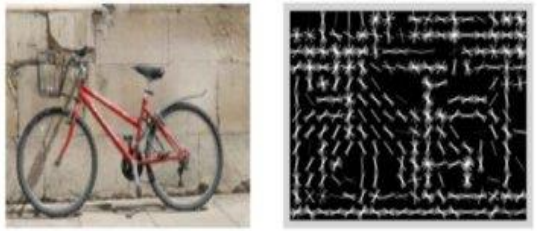


Image gradient

Edge detection



Histograms of Oriented Gradients

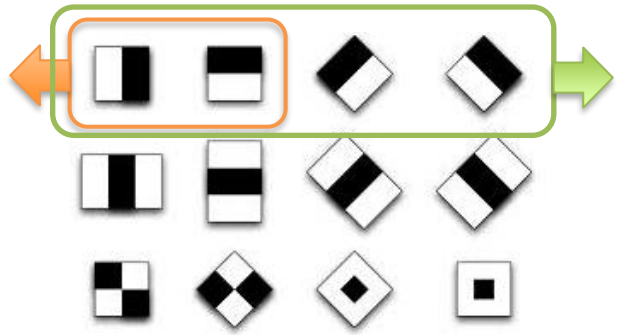
Optical Flow

Object Descriptors

How to represent and detect an object?

Many approaches over the years…

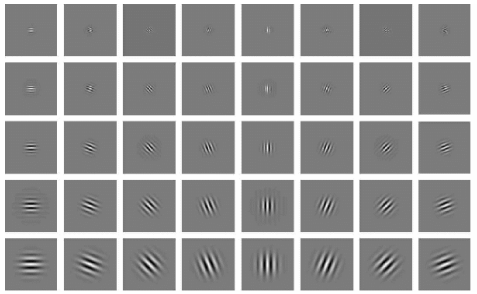
Horizontal and vertical gradients



Oriented gradients

Haar Wavelets

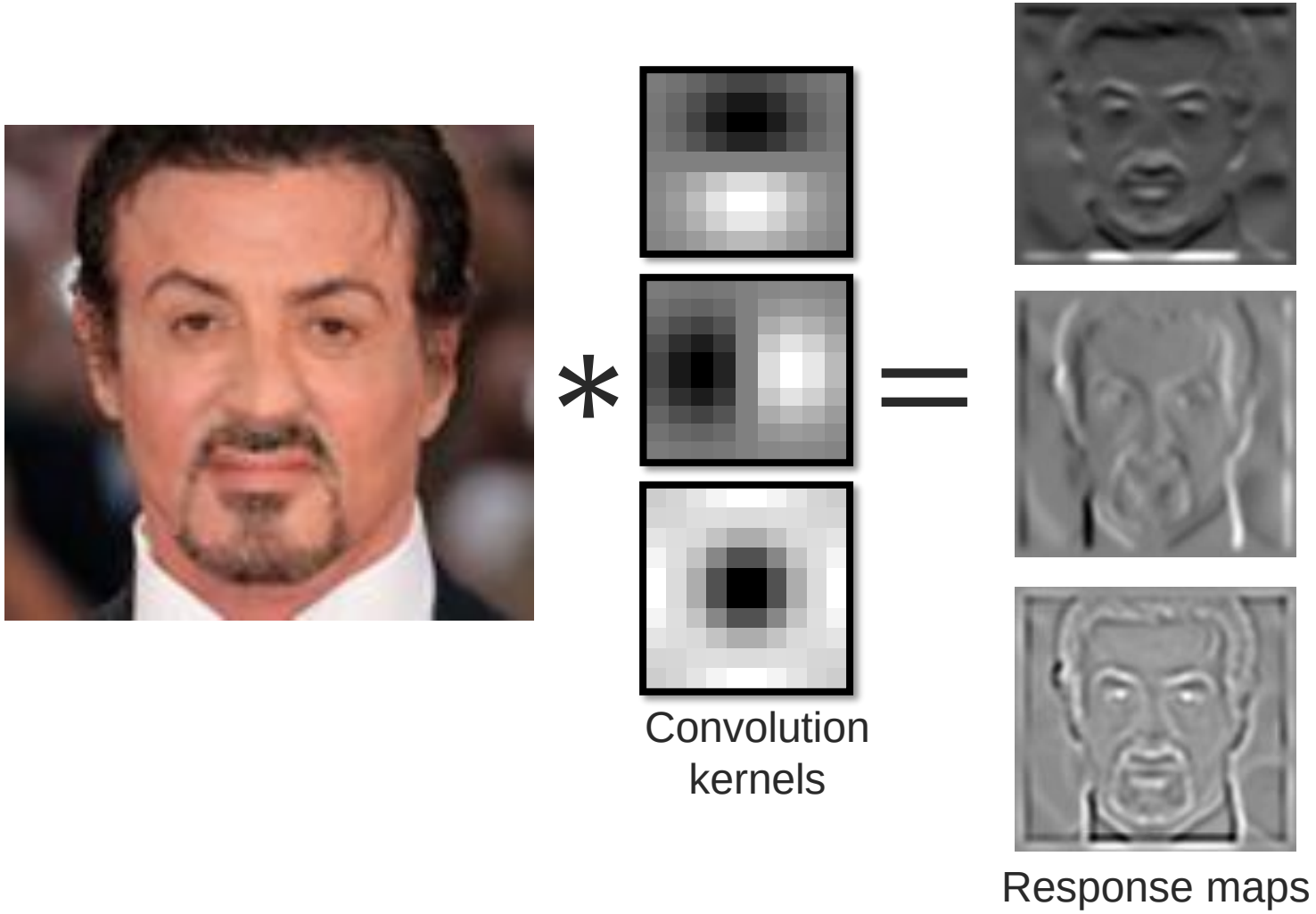
Templates tested on the image *(i.e., convolution kernels)*



Gabor filters

Inspired by visual cortex

Convolution Kernels

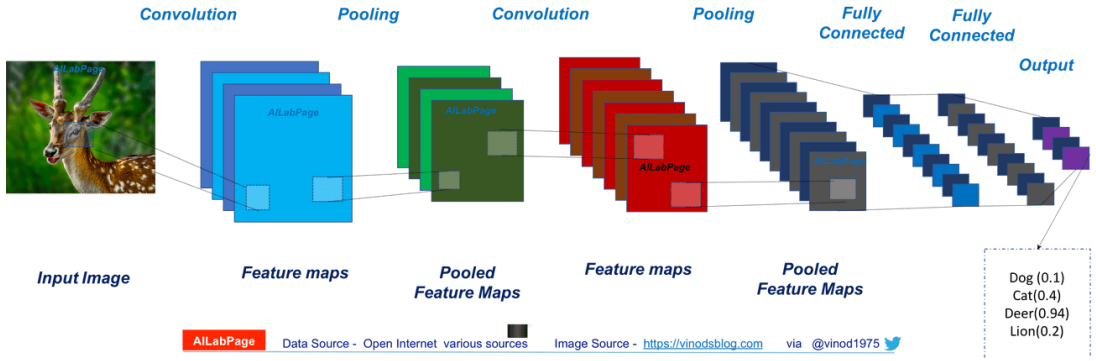


Object Descriptors

How to represent and detect an object?

Many approaches over the years…

Convolutional Neural Network (CNN)



More details about CNNs is coming…

… and we will also talk about visual transformers in coming weeks…

And images are more than a list of objects!

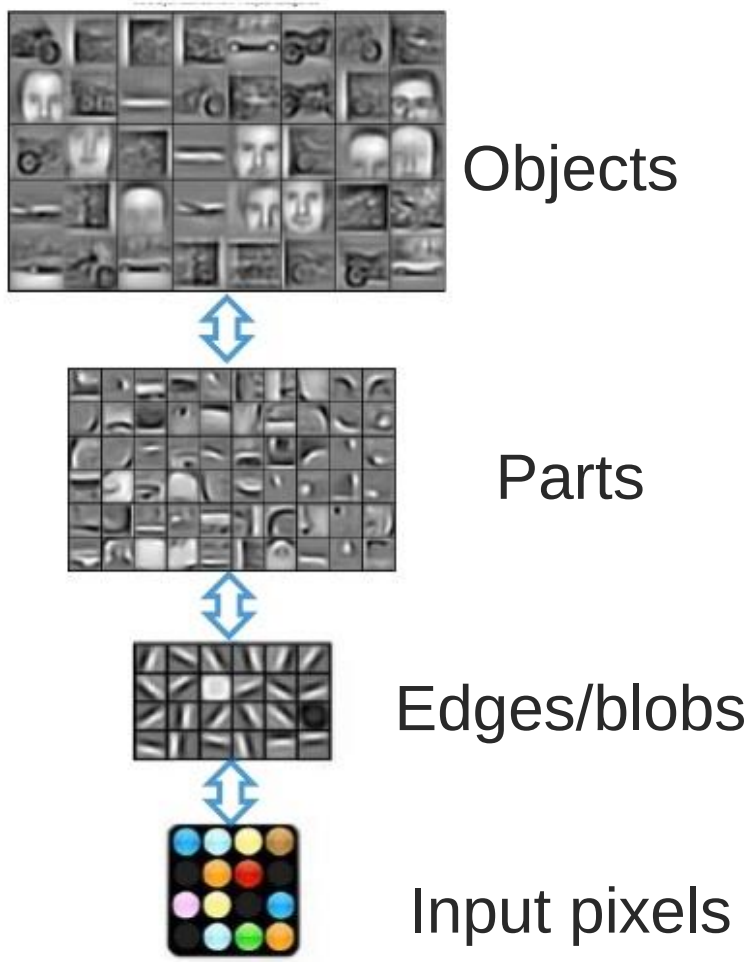
Convolutional Neural Networks

Why using Convolutional Neural Networks?

Goal: building more abstract, hierarchical visual representations

Key advantages:

1) Inspired from visual cortex
2) Encourages visual abstraction
3) Exploits *translation invariance*
4) Kernels/templates are learned
5) Fewer parameters than MLP



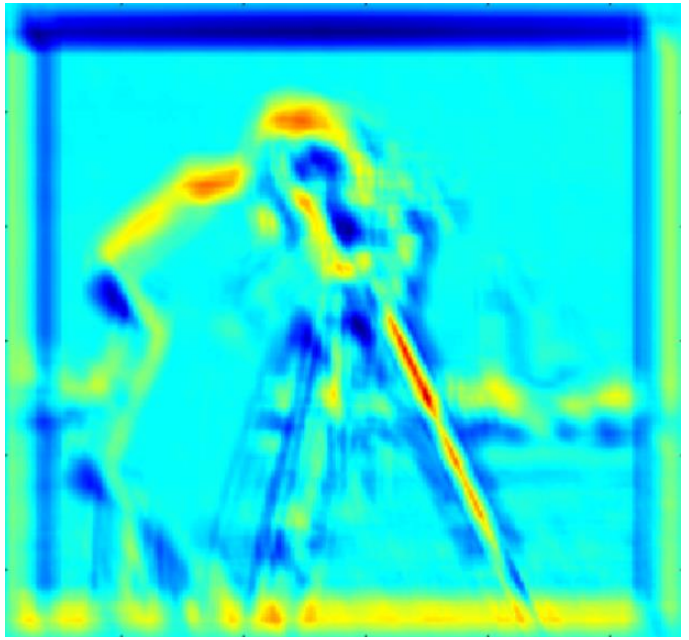
Convolution in 2D – Example

Input image

*

Convolution kernel

=



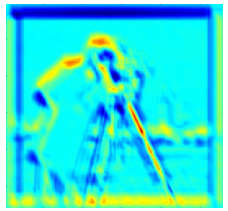
Response map

Convolution as a Fully-Connected Network

Input

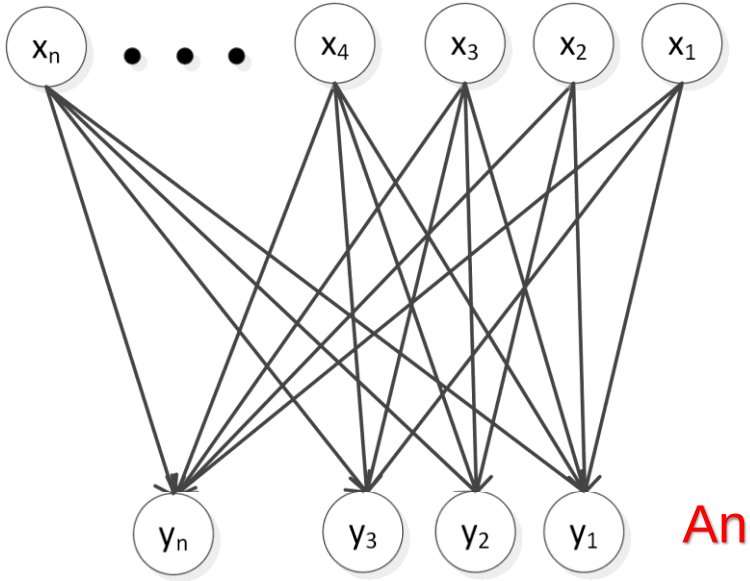
(image)

Output



(response map)

Input: all pixels



Output: kernel responses

Not efficient!

200×200 image
requires
$40{,}000 \times n$ parameters

(where n is size of kernel)

And it may learn different kernels for different pixel positions

➡ Not translation invariant

Convolutional Neural Layer

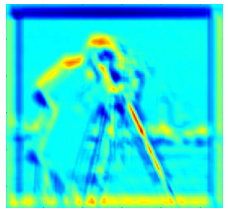
Input

(image)

Weighted sum

$\boldsymbol{Wx}$

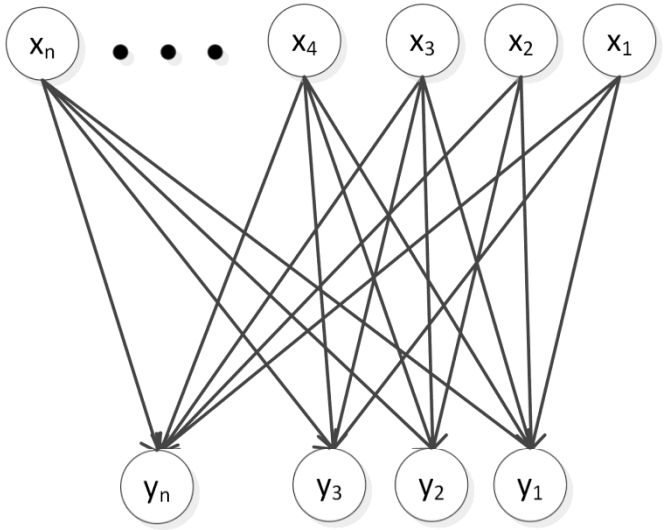
Output



(response map)

$y = Wx$

Input: all pixels



Output: kernel responses

Example with 1D kernel:

w_1	w_2	w_3

Convolution kernel

Convolutional Neural Layer

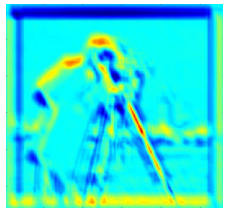
Input

(image)

Weighted sum

Wx

Output

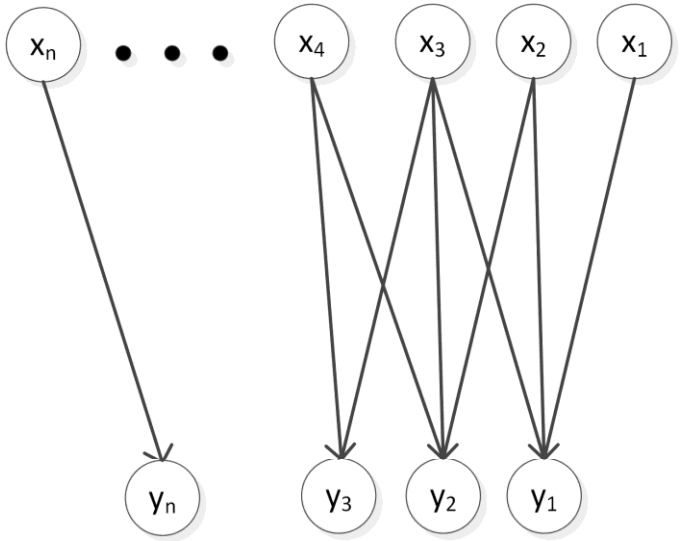


(response map)

$y = Wx$

Modification 1: Sliding window – Only apply the kernel to a small region

Input: all pixels



Output: kernel responses

Example with 1D kernel:

w_1	w_2	w_3

Convolutional Neural Layer

Modification 2: Same kernel applied to all sliding windows

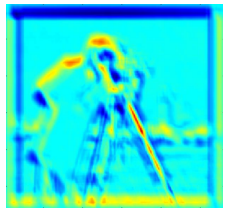
Input

(image)

Weighted sum

Wx

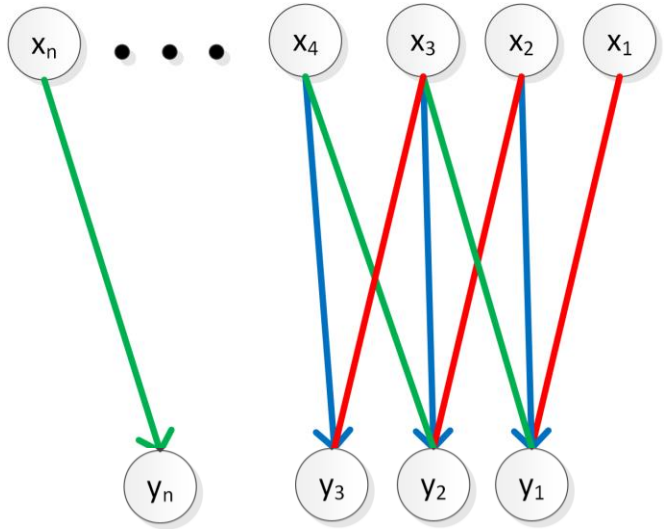
Output



(response map)

$y = Wx$

Input: all pixels



Output: kernel responses

Example with 1D kernel:

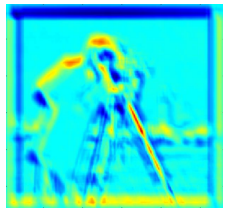
Convolutional Neural Layer

Modification 2: Same kernel applied to all sliding windows

Input

(image)

Output



(response map)

$$y = Wx$$

$$W = \begin{pmatrix} w_1 & w_2 & w_3 & & 0 & 0 & 0 \\ 0 & w_1 & w_2 & \cdots & 0 & 0 & 0 \\ 0 & 0 & w_1 & & 0 & 0 & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & 0 & 0 & & w_3 & 0 & 0 \\ 0 & 0 & 0 & \cdots & w_2 & w_3 & 0 \\ 0 & 0 & 0 & & w_1 & w_2 & w_3 \end{pmatrix}$$

Example with 1D kernel:

➡ Can be implemented efficiently on GPUs

➡ *W* will be 3D: 3rd dimension allows for multiple kernels

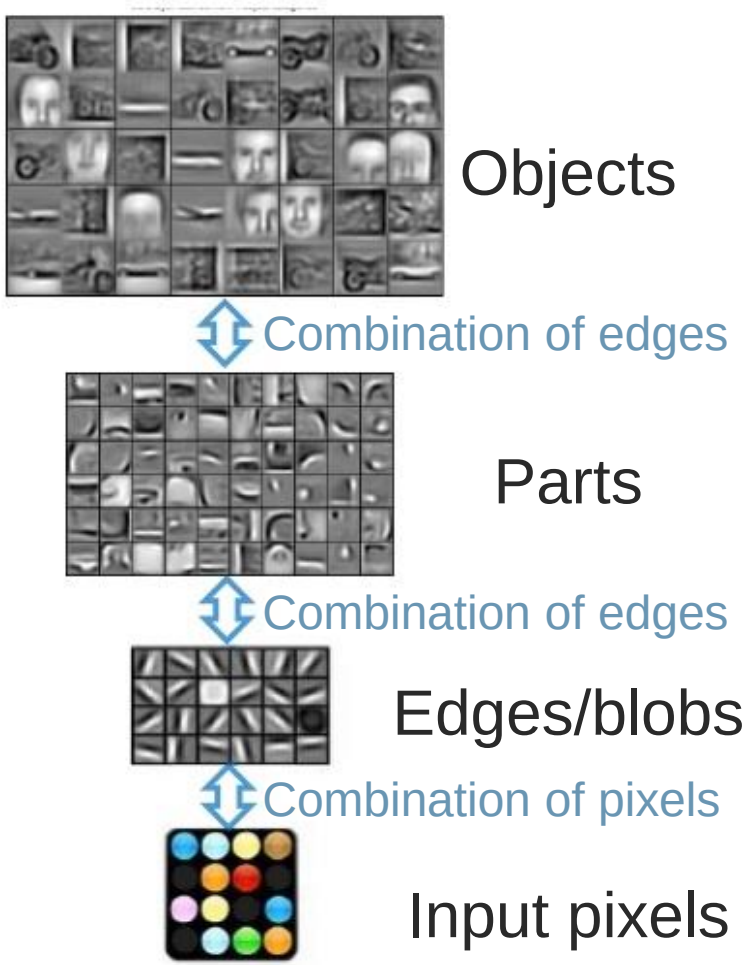
Convolutional Neural Network

Multiple convolutional layers

➡ Allows the network to learn combinations of sub-parts, to increase complexity

but how to encourage abstraction and summarization?

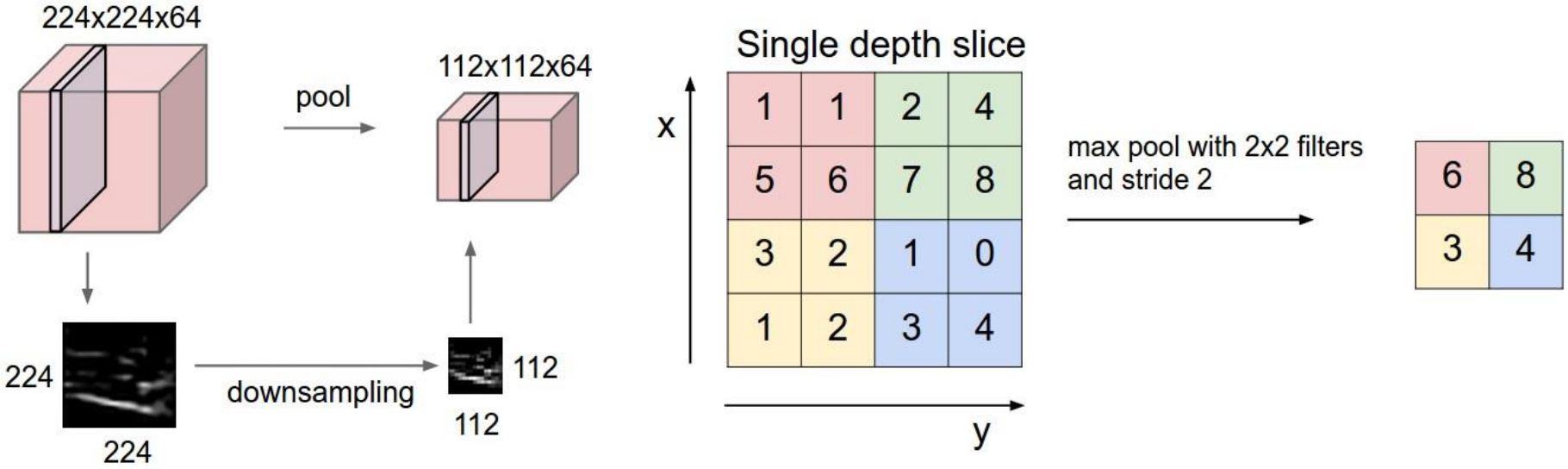
Answer: Pooling layers



Pooling Layer

Response map subsampling:

Allows summarization of the responses

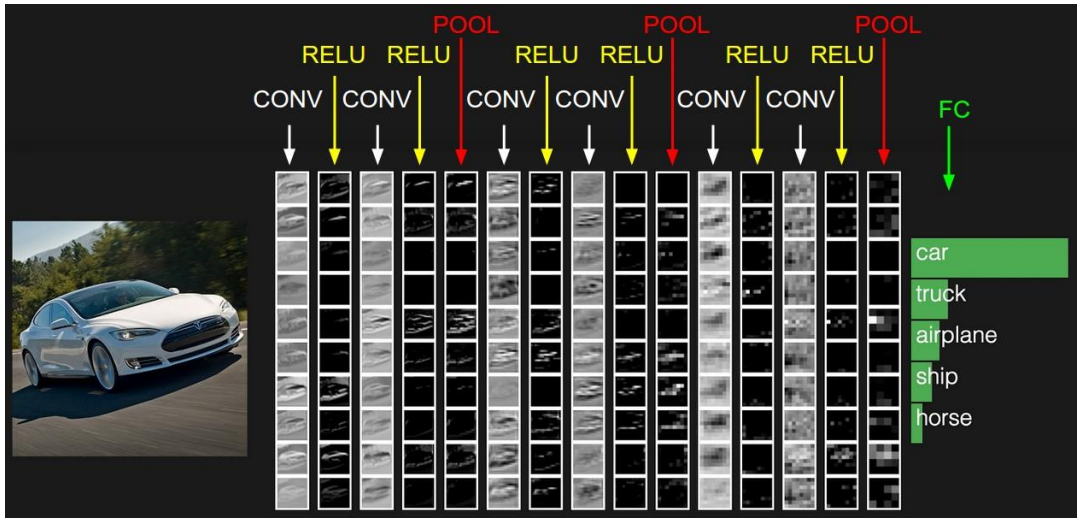


Common architectures

Repeat several times:

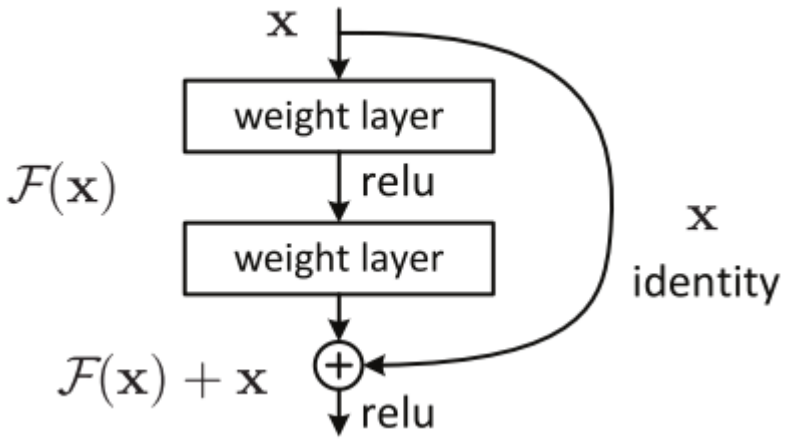
- Start with a convolutional layer
- Followed by non-linear activation and pooling

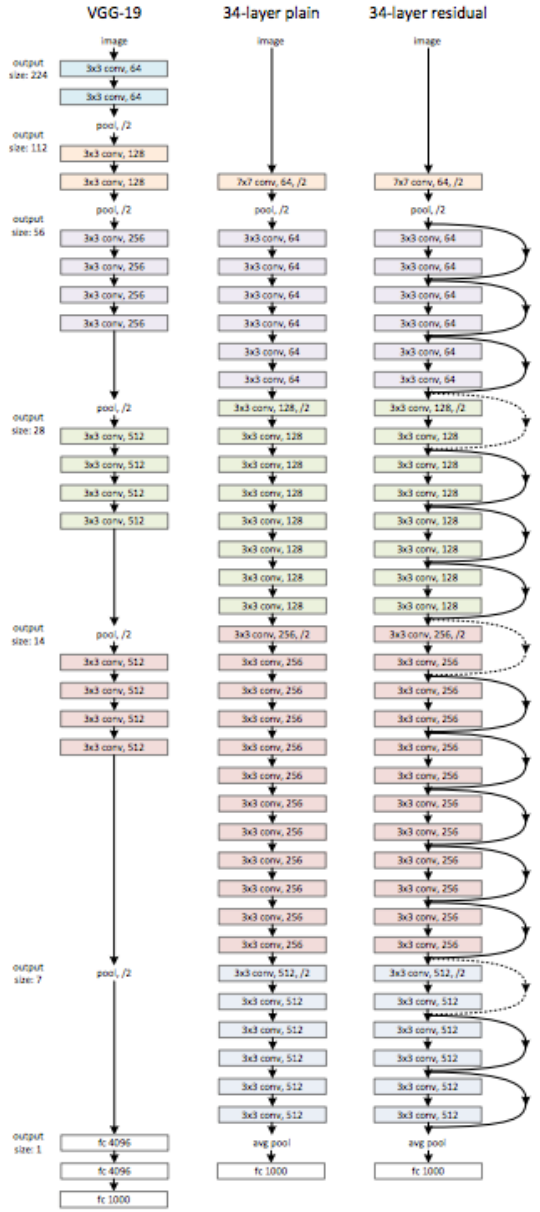
End with a fully connected (MLP) layer



Residual Networks (ResNet)

Adding residual connections





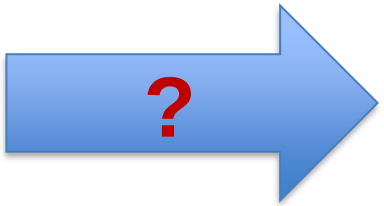
ResNet (He et al., 2015)

- Up to 152 layers!

Region-based CNNs

Object Detection (and Segmentation)

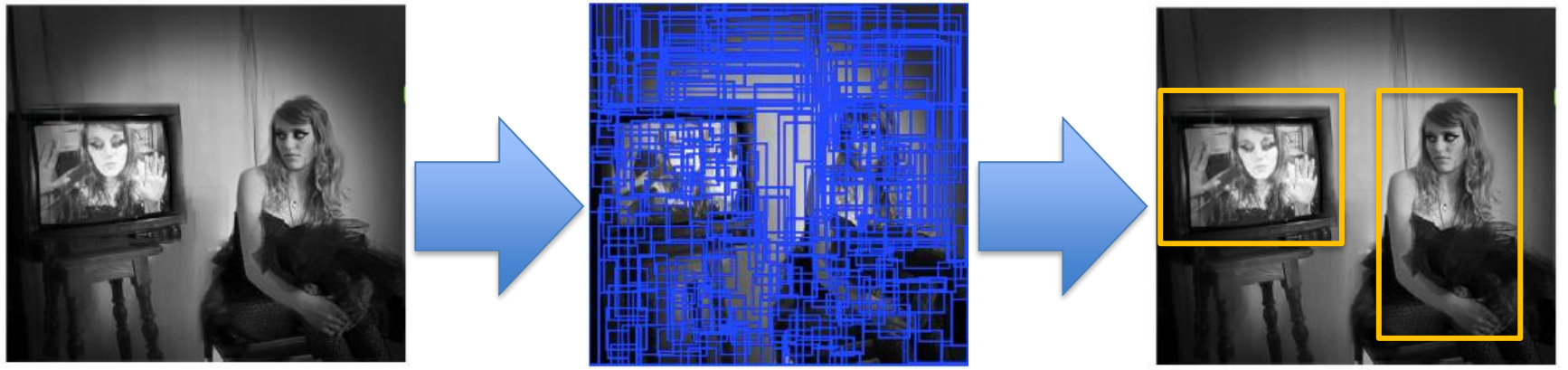
Input image



Detected Objects

One option: Sliding window

Object Detection (and Segmentation)

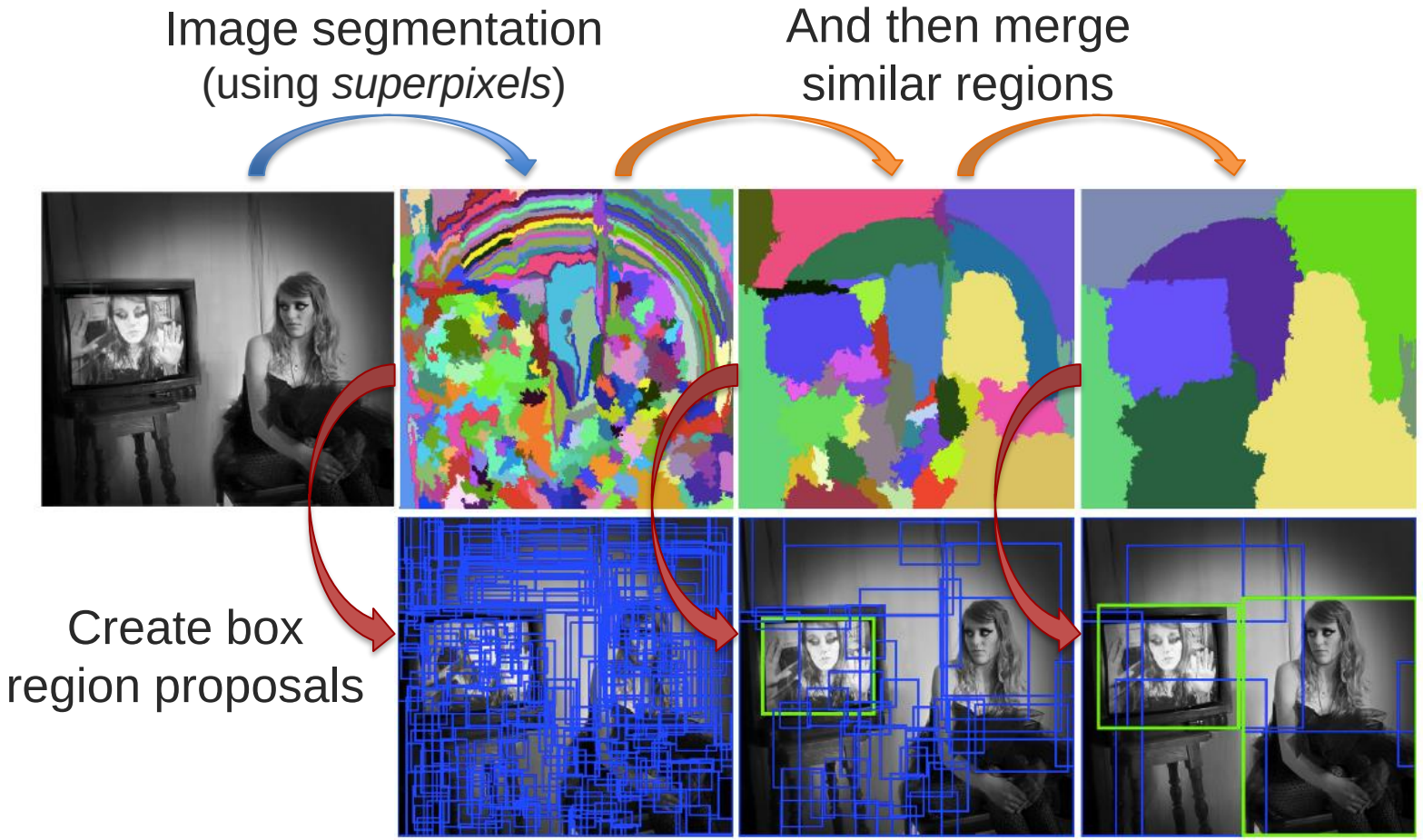


Input image Region Proposals Detected Objects

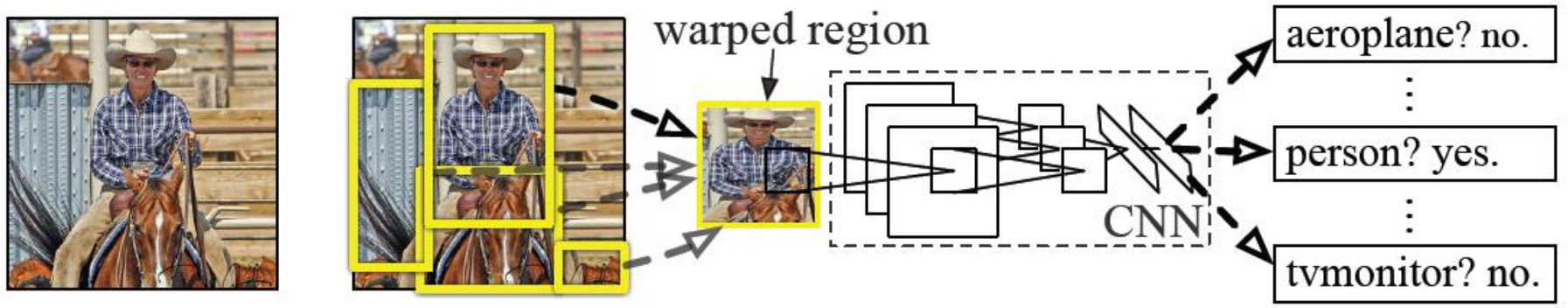
A better option: Start by Identifying hundreds of region proposals and then apply our CNN object detector

How to efficiently identify region proposals?

Selective Search [Uijlings et al., IJCV 2013]



R-CNN [Girshick et al., CVPR 2014]



- Select ~2000 region proposals ⇒ Time consuming!
- Warp each region
- Apply CNN to each region ⇒ Time consuming!

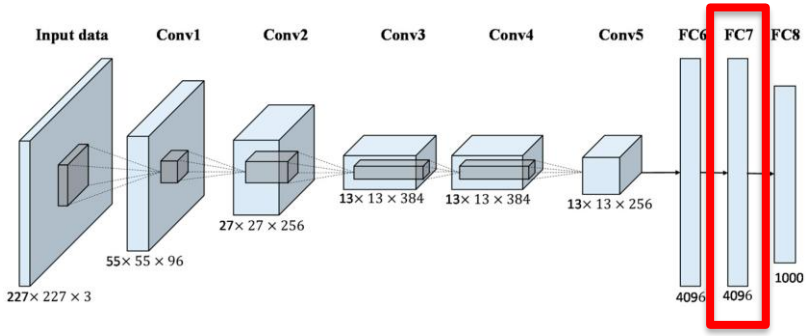
Fast R-CNN: Applies CNN only once, and then extracts regions

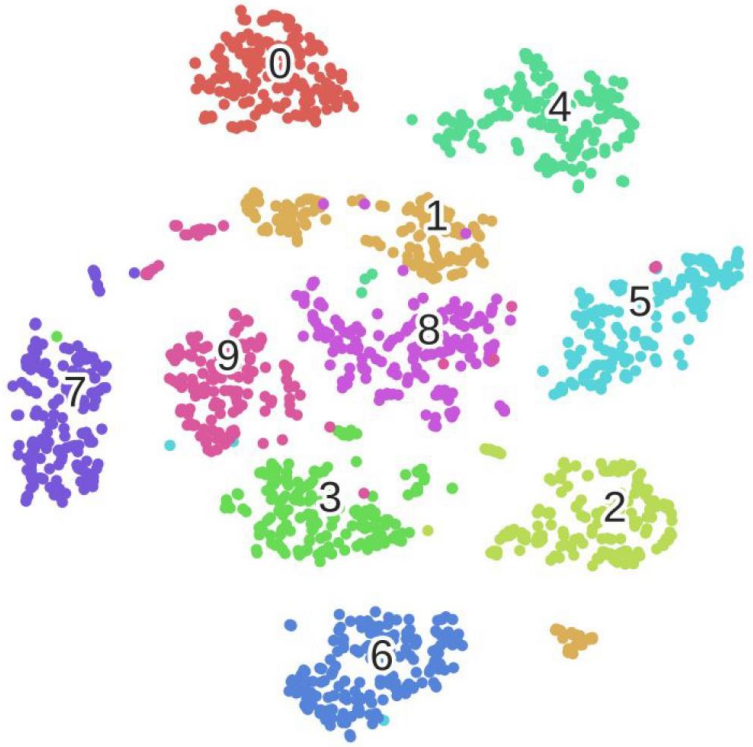
Faster R-CNN: Region selection on the Conv5 response map

Visualizing CNNs

Visualizing the Last CNN Layer: t-sne

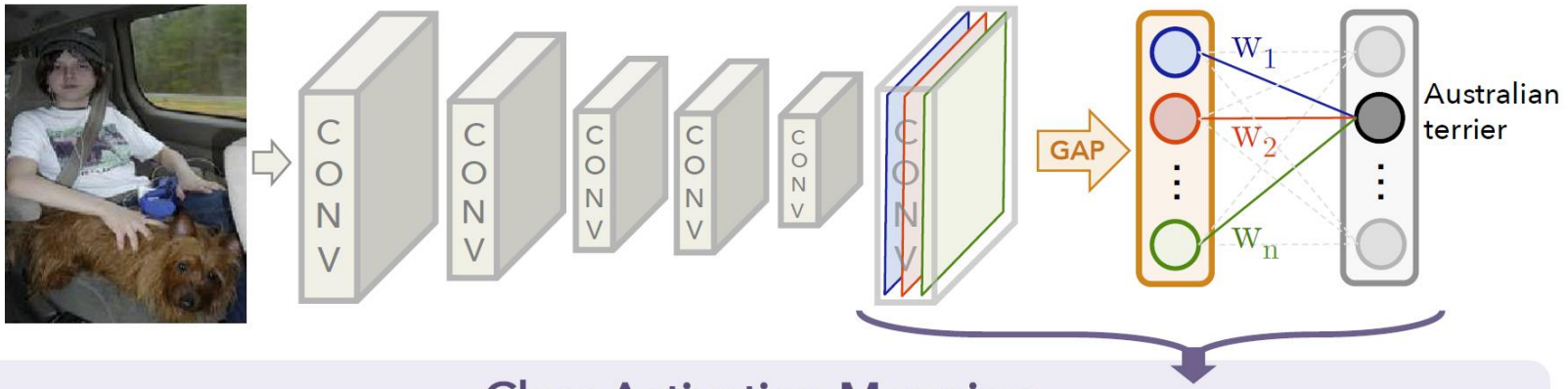
Alex Net

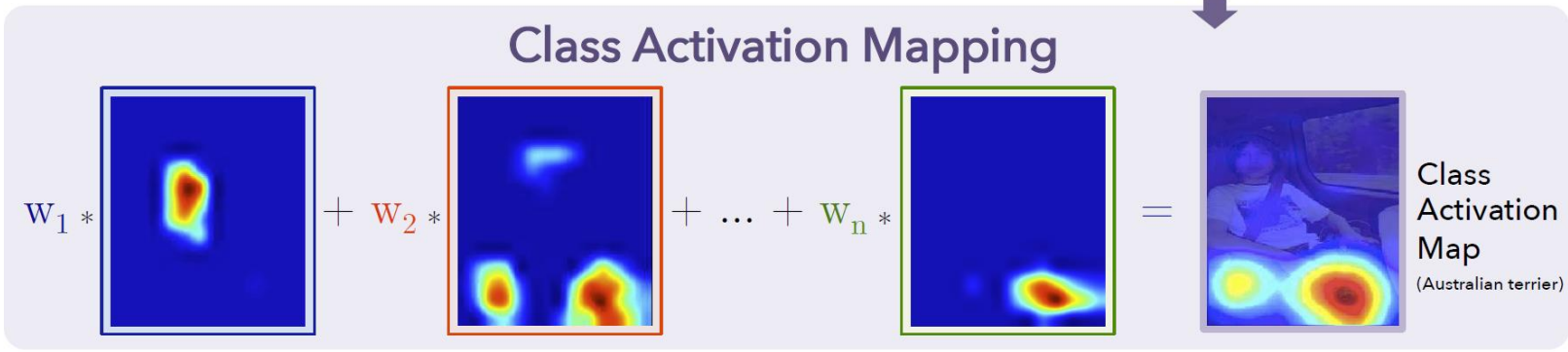




Embed high dimensional data points (i.e. feature codes) so that pairwise distances are conserved in local neighborhoods.

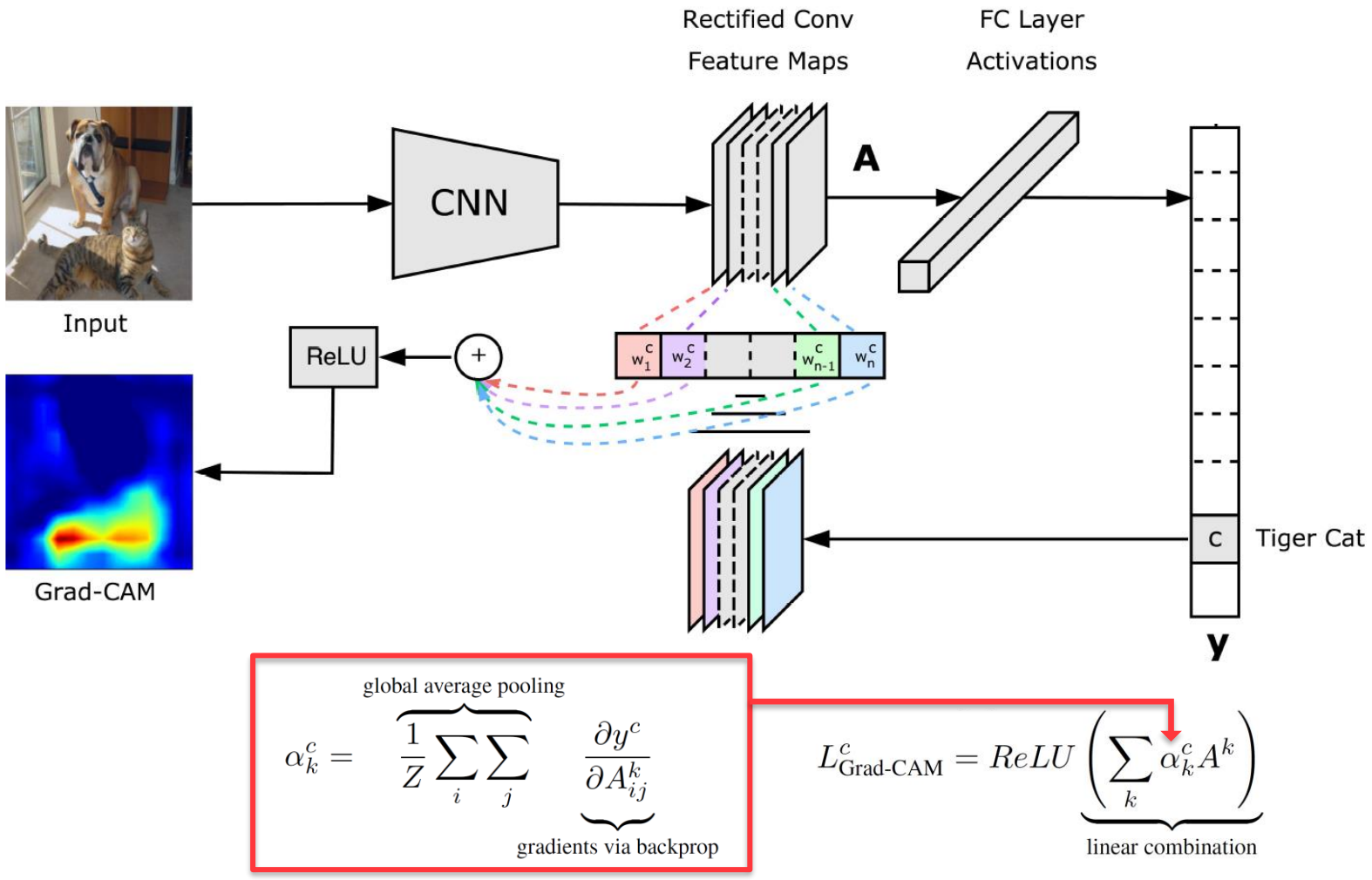
CAM: Class Activation Mapping [CVPR 2016]





$$L^c_{\text{CAM}} = \underbrace{\sum_k w^c_k A^k}_{\text{linear combination}}$$

Grad-CAM [ICCV 2017]



$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}}$$

$$L^c_{\text{Grad-CAM}} = ReLU\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}}$$

Are CNNs over?

- Eesearch indicates that CNNs are still alive
- They are more efficient, easier to use, great baselines