An old Artificial Intelligence dream that comes true: Merging language and vision modalities

Raffaella Bernardi

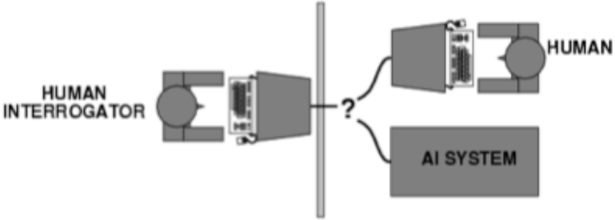
University of Trento

November, 2017

Credits

L. Fei Fei, Tamara Berg, Andrej Karpathy, Angeliki Lazaridou, Elia Bruni, Marco Baroni, Chris McCormick

An old AI dream

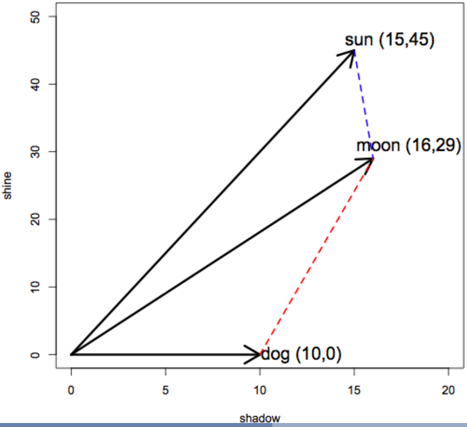


Recall: From words to Meaning Representation

Compute the word distribution

Words can be represented by vectors harvested from a corpus of texts *counting* word co-occurences.

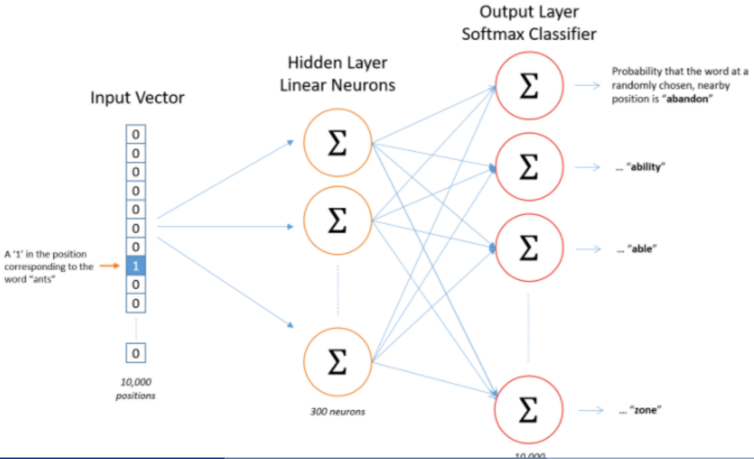
	shadow	shine
moon	16	29
sun	15	45
dog	10	0



From words to Meaning Representation

Predict the context: Word2Vec (Skip-Gram)

Instead counting words co-occurrences, the vector representing a word can be learned by *predicting* its nearby word. (Mikolov et al, 2013)



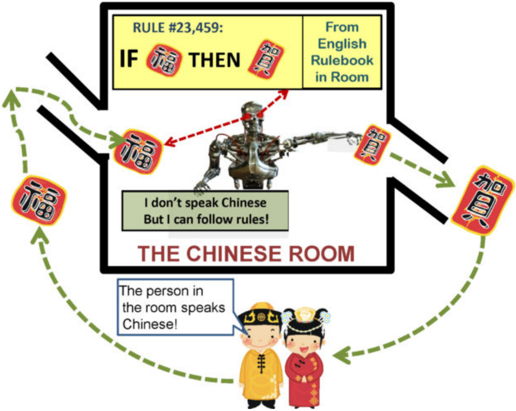
Recall: Vector Representations

Successful

- Lexical meaning
  - Synonyms
  - Concept categorization (eg. car ISA vehicle)
  - Selectional preferences (e.g. eat chocolate vs. *eat sympathy)
  - relation classification (exam-anxiety CAUSE-EFFECT relation)
  - salient properties (car-wheels)
- Compositionality: Phrase and Sentence
  - similairity
  - entailment

Vector Representations

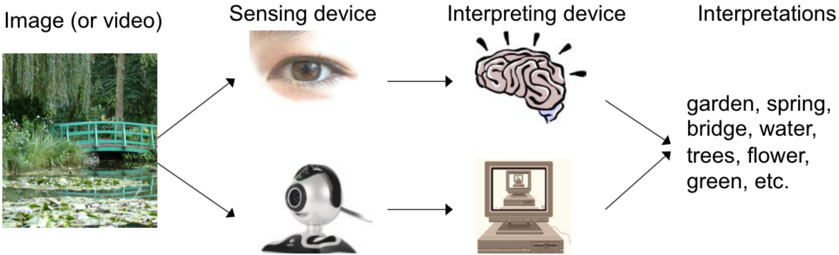
Grounding Problem


RULE #23,459:
IF 福 THEN 賀
From English Rulebook in Room
福
福
賀
加
賀
I don't speak Chinese But I can follow rules!
THE CHINESE ROOM
The person in the room speaks Chinese!

Grounding language representations into the world.

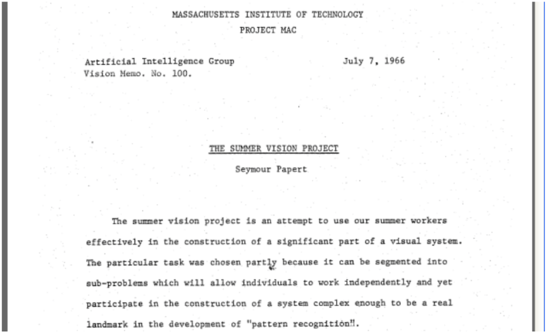
Point to the *reference* of our mental representation.

What is (Computer) Vision



How did it started?

Origins of computer vision: an MIT undergraduate <u>summer project</u>



MASSACHUSETTS INSTITUTE OF TECHNOLOGY
PROJECT MAC

Artificial Intelligence Group July 7, 1966
Vision Memo. No. 100.

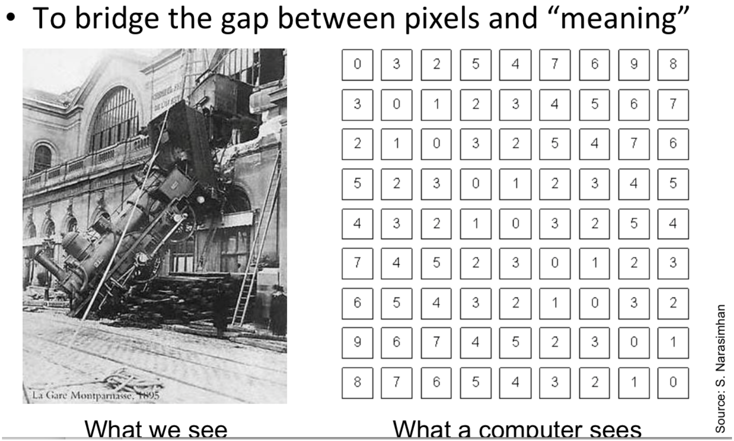
THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

From Pixels to Meaning Representation

Gap



- To bridge the gap between pixels and "meaning"

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What we see

What a computer sees

Source: S. Narasimhan

From Pixels to Meaning

Pixels

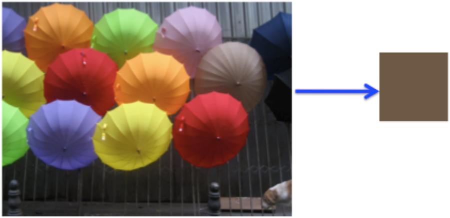
Raw images representation consists of pixels (a pixel is the minimum element of an image).
Pixels, identified by their physical coordinates, are stored as numbers encoding their color intensity. For instance, a black and white image is a 1-D representation of the pixel brightness);
A colored image is a 3-D arity of intensity values:

$$f(x,y) = \left[\begin{array}{l} red(x,y), \\ green(x,y), \\ blue(x,y) \end{array} \right.$$

where color(x,y) is the intensity of that color (x) at position (y).

How to represent an image: Keep all the pixels

How to represent an image: Compute average pixel



How to represent an image: Spatial grid of average pixel colors?

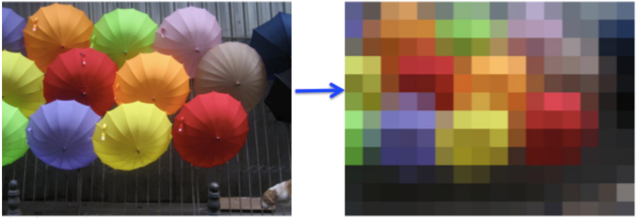
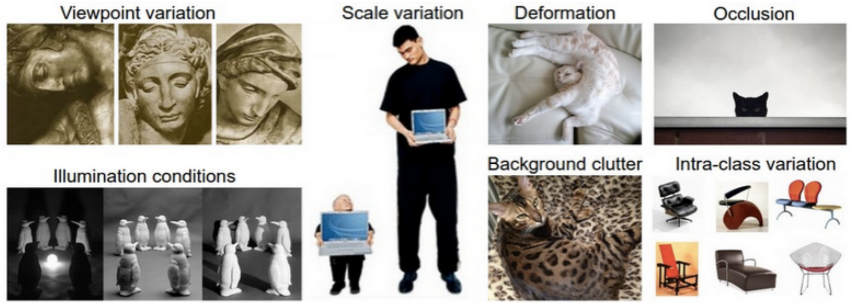


Photo by: marielito

From Pixels to Meaning Representation

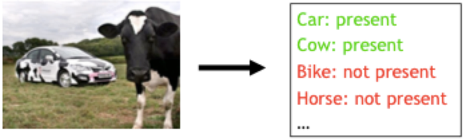
Challanges



Applications: Traditional CV tasks

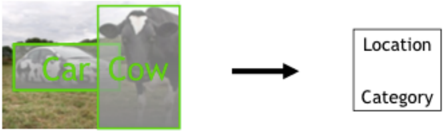
Objects

Image classification: assigning a label to the image.



Car: present
Cow: present
Bike: not present
Horse: not present
...

Object localization: define the location and the category.

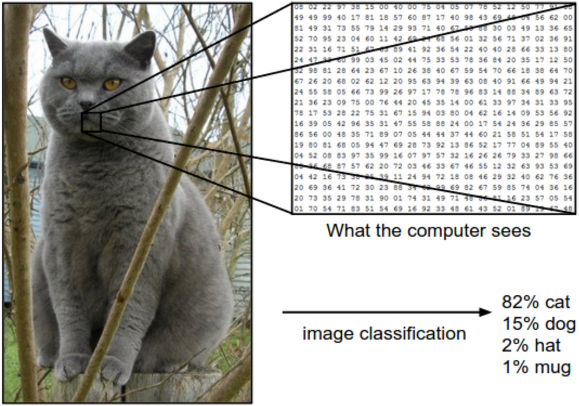


Location

Category

Similarly, scene recognition.

Object Classification

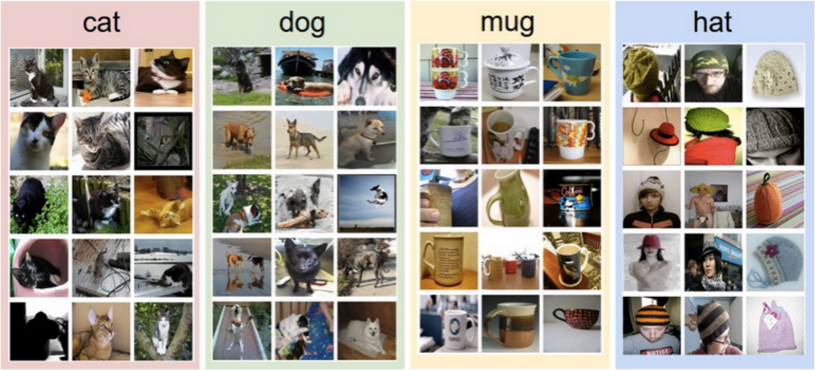


```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

The task in Image Classification is to predict a single label (or a distribution over labels as shown here to indicates our confidence) for a given image. Images are 3-dimensional arrays of integers from 0 to 255, of size Width x Height x 3. The 3 is due to the three color channels Red, Green, Blue.

Data Driven

Data-driven approach: it relies on first accumulating a training dataset of labeled images.



An example training set for four visual categories. In practice we may have thousands of categories and hundreds of thousands of images for each category.

The image classification pipeline

- Input. Our input consists of a set of N images, each labeled with one of K different classes. We refer to this data as the *training set*.
- Learning. Our task is to use the training set to learn what every one of the classes looks like. We refer to this step as *training a classifier*, or learning a model.
- Evaluation. In the end, we evaluate the quality of the classifier by asking it to *predict labels* for a new set of images that it has never seen before (*test set*). We will then compare the true labels of these images to the ones predicted by the classifier. Intuitively, we're hoping that a lot of the predictions match up with the true answers (which we call the ground truth).

Nearest Neighbor Classifier

The nearest neighbor (NN) classifier will

1. take a test image,
2. compare it to every single one of the training images, and
3. predict the label of the closest training image.

Nearest Neighbor examples

In only about 3 out of 10 examples an image of the same class is retrieved, while in the other 7 examples this is not the case. For example, in the 8th row the nearest training image to the horse head is a red car, presumably due to the strong black background. As a result, this image of a horse would in this case be mislabeled as a car.

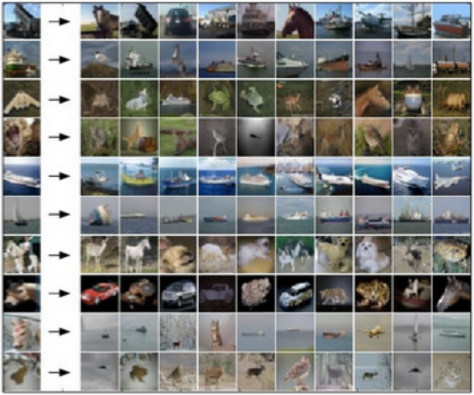
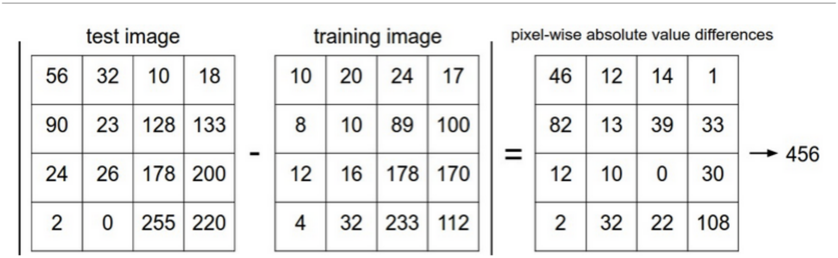


Image distance

The difference (or the familiar cosine similarity)



$$\left| \overset{\text{test image}}{\begin{array}{|c|c|c|c|}\hline 56 & 32 & 10 & 18 \\ \hline 90 & 23 & 128 & 133 \\ \hline 24 & 26 & 178 & 200 \\ \hline 2 & 0 & 255 & 220 \\ \hline\end{array}} - \overset{\text{training image}}{\begin{array}{|c|c|c|c|}\hline 10 & 20 & 24 & 17 \\ \hline 8 & 10 & 89 & 100 \\ \hline 12 & 16 & 178 & 170 \\ \hline 4 & 32 & 233 & 112 \\ \hline\end{array}} \right| = \overset{\text{pixel-wise absolute value differences}}{\begin{array}{|c|c|c|c|}\hline 46 & 12 & 14 & 1 \\ \hline 82 & 13 & 39 & 33 \\ \hline 12 & 10 & 0 & 30 \\ \hline 2 & 32 & 22 & 108 \\ \hline\end{array}} \rightarrow 456$$

An example of using pixel-wise differences to compare two images with L1 distance (for one color channel in this example). Two images are subtracted elementwise and then all differences are added up to a single number. If two images are identical the result will be zero. But if the images are very different the result will be large.

Evaluation

The CIFAR-10 training set of 50,000 images (5,000 images for every one of the labels), and we wish to label the remaining 10,000.
The NN Classifier based on raw pixel representation and the image distance measure above reaches 38.6 % accuracy (vs. upper bound: 94% – human).
State of the art classifier, Convolutional Neural Network reaches 95%.

K-Nearest Neighbor Classifier

You may have noticed that it is strange to only use the label of the nearest image when we wish to make a prediction. Indeed, it is almost always the case that one can do better by using what's called a k-Nearest Neighbor Classifier. The idea is very simple: instead of finding the single closest image in the training set, we will find the top k closest images, and have them vote on the label of the test image.
Which is the best k?
K is an hyperparameter. There are others too.

Validation dataset vs Test dataset

They can be trained/learned.

- Training data-set: to train the classifier.
- Trail data-set (or development dataset or validatation dataset): to tune the parameters.
- Test data-set. To test the classifier.

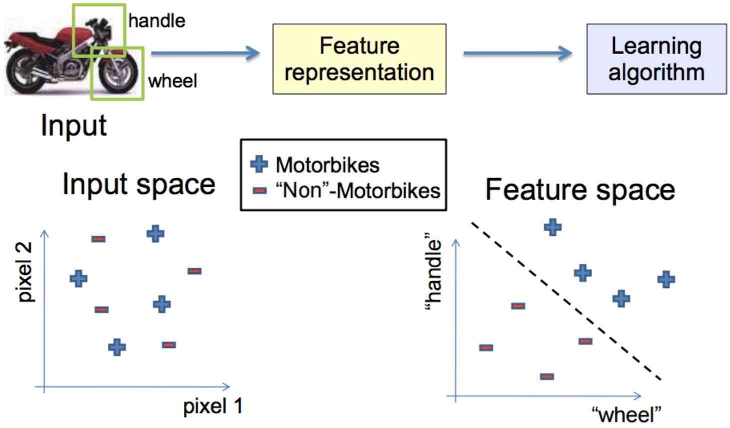
First problem: the Raw Pixel representation

- Using the NN classifier over the raw pixel representation images that are nearby each other are much more a function of the general color distribution of the images, or the type of background rather than their semantic identity.
- For example, a dog can be seen very near a frog since both happen to be on white background.
- Ideally we would like images of all of the 10 classes to form their own clusters, so that images of the same class are nearby to each other regardless of irrelevant characteristics and variations (such as the background).

To get this property we will have to go beyond raw pixels.

From Pixels to Meaning

Abstract Features



Second problem: the classifier

NN Classifier: pro and contra.

- the classifier takes no time to train, since all that is required is to store and possibly index the training data.
- However, we pay that computational cost at test time, since classifying a test example requires a comparison to every single training example.
- This is backwards, since in practice we often care about the test time efficiency much more than the efficiency at training time.

In CV it's better to use other classifiers.
State of the art: Deep neural networks are very expensive to train, but once the training is finished it is very cheap to classify a new test example. This mode of operation is much more desirable in practice.

First Revolution: Big dataset

ImageNet

Image database organized according to the WordNet hierarchy.
Stanford Vision Lab, Stanford University & Princeton University.

- Challenges: 2007-present
- AMT: 48,940 annotators from 167 countries
- 15M images
- 22K categories of objects

From Pixels to Features

Two methods

- Bag of Visual words (BoVW) (Sivic and Zisserman, 2003)
- Convolutional neural network (CNN) (LeCun et al., 1998, Krizhevsky et ali. 2012)

From Pixels to Features: BoVW

Pipeline

Keypoints detectors To locate interesting points/content, various kinds of low-level features detectors exists:

- edge detection: the lines we would draw – encode shape info
- corner detection

Local description The identified interesting points are then described: clustered into regions and transformed into *vectors representing the region*. Several local descriptors exist, e.g:

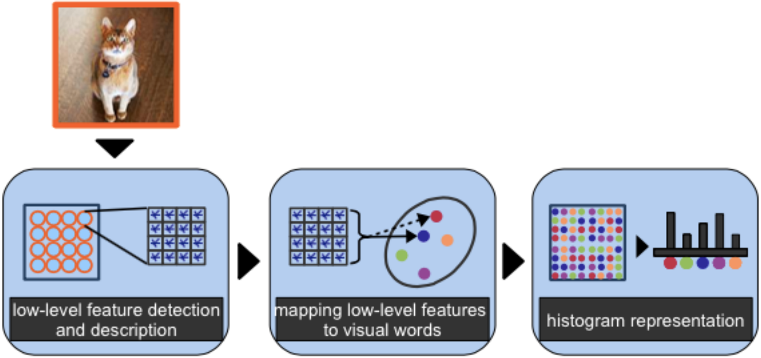
- SIFT: Scale-invariante feature transform (Lowe '99) – edge based features.
- Textons (Leung and Malik '01)
- HoG (Histograms of Oriented Gradients) (Dalal and Triggs '05)

The low-level features can capture eg. Color, Texture, Shape,

Bag of Visual Words The local descriptions are clustered to obtain the Visual Words that are used to obtain the vector representation of the image.

From Pixels to Features: BoVW

BoVW's pipeline


low-level feature detection and description
mapping low-level features to visual words
histogram representation

Second Revolution: End-to-end systems

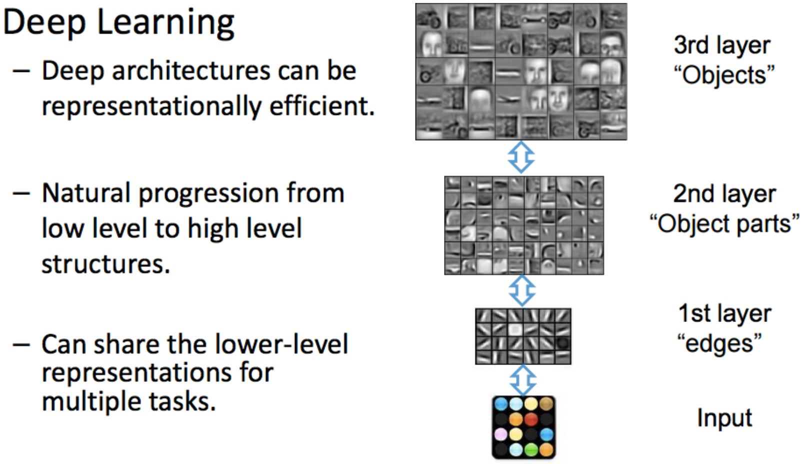
Convolutional Neural Networks

ImageNet Classification with Deep Convolutional Neural Networks Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton, 2012

- 2012: Krizhevsky outperformed the other systems using CNN
- 2013: half of the systems used CNN
- 2014: All of the systems used CNN.

End-to-end Systems

Hierarchy of features

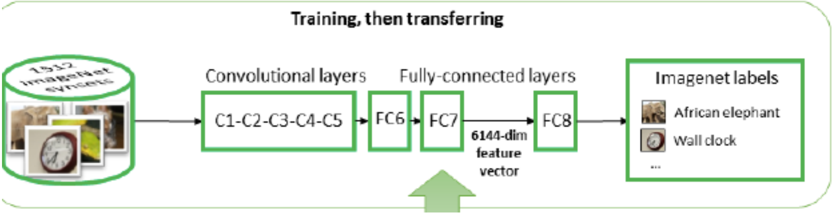


Deep Learning

- Deep architectures can be representationally efficient.
- Natural progression from low level to high level structures.
- Can share the lower-level representations for multiple tasks.

End-to-end systems

CNN: off-the-shelf vector representation

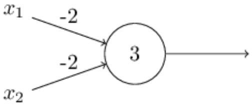


- Train a CNN on a vision task (e.g. AlexNet on ImageNet)
- Do a forward pass given an image input
- Transfer one or more layers (e.g. FC7 or C5)

Neural Networks

Example to compute a logical operator “Not And”

x_1	x_2	“Not and”
0	0	1
0	1	1
1	0	1
1	1	0

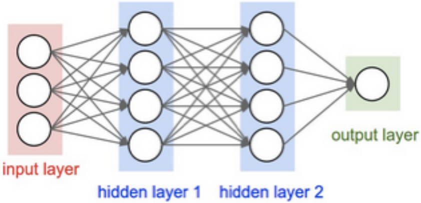


Input 00 produces output 1 (since: (-2)*0 + (-2) * 0 + 3 = 3 is positive) and similarly, 01 and 10; but the input 11 produces output 0 (since: (-2)*1 + (-2) * 1 + 3 = -1 is negative.)

Neural Networks

Neural Networks

It's a composition of functions (neurons) that goes from an n-dimensional vector to class scores.



Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. On the last (fully-connected) layer, they have a loss function (e.g., Softmax).

Neural Networks

Recurrent NN: intuitions

Traditional neural networks cannot use the information “about previous inputs” to inform later ones.

- *Recurrent neural networks* (RNNs) address this issue: They are networks with loops in them, allowing information to persist. They work well with short dependencies.
- *Long Short Term Memory* (LSTM) are a special kind of RNN, capable of learning long-term dependencies.

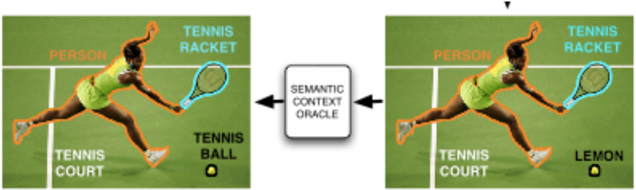
Language and Vision

Language and Visual Space can be combined!

Applications: Traditional CV tasks

Corpora as KB source: Object recognition

A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, S. Belongie (ICCV 2007) Objects in Context.

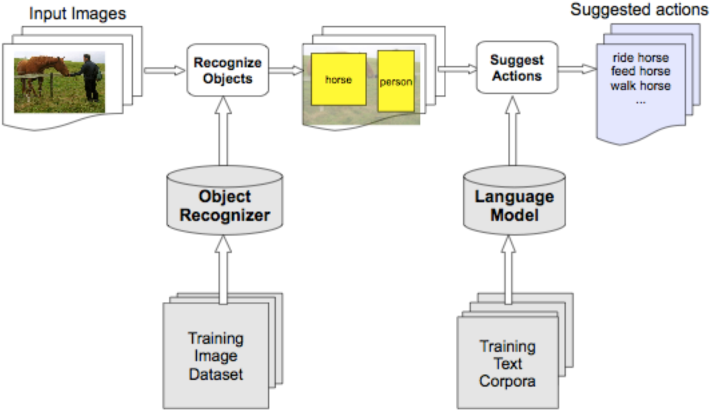


Not a Lemon, it's more probable a Tennis Ball. Info come from a KB (word similarity list, exctracted from internet – Google Sets).

Applications: Traditional CV tasks

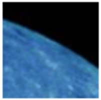
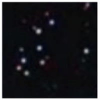
Corpora as KB source: Action recognition

Thu Le Dieu, Jasper Uijlings and R. Bernardi (2010, 2011)

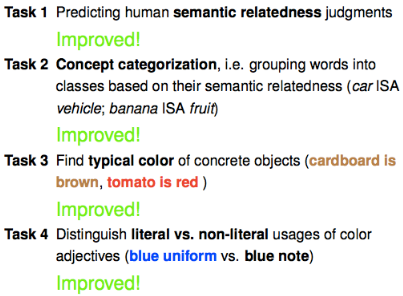


Applications: Traditional NLP tasks

E. Bruni, G.B. Tran and M. Baroni (GEMS 2011, ACL 2012, Journal of AI 2014), E. Bruni, G. Boleda, M. Baroni and N. Tran (ACL 2012)

	planet	night		
moon	10	22	22	0
sun	14	10	15	0
dog	0	4	0	20

Applications: Traditional NLP tasks



Task 1 Predicting human **semantic relatedness** judgments

Improved!

Task 2 **Concept categorization**, i.e. grouping words into classes based on their semantic relatedness (*car* ISA *vehicle*; *banana* ISA *fruit*)

Improved!

Task 3 Find **typical color** of concrete objects (**cardboard is brown**, **tomato is red**)

Improved!

Task 4 Distinguish **literal vs. non-literal** usages of color adjectives (**blue uniform** vs. **blue note**)

Improved!