Chapter 30. Natural Languages and Natural Scenes

The Quest for Artificial Intelligence, Nilsson, N. J., 2009.

Lecture Notes on Artificial Intelligence, Spring 2012

Summarized by Heo, Min-Oh and Lee, Sangwoo

Biointelligence Laboratory
School of Computer Science and Engineering
Seoul National Univertisy

http://bi.snu.ac.kr

Contents

Overview of Chapter 30

- Difficult problem to deal with natural languages and natural scenes
  - AI complete: generally as intelligent as humans
- Approaches for NLP
  - Introducing grammars and parsing algorithms for NLP
  - Introducing probability on parse tree
    - probabilistic context-free grammars (PCFGs)
- Computer vision for natural scenes
  - Recovering surfaces and depth information
  - Tracking moving objects using particle filtering
  - Introducing hierarchical model for images
  - Building stochastic grammars of images

Chapter 30. Natural Languages and Natural Scenes

30.1 Natural Language Processing

Natural Language Processing (NLP)

- Difficult problem to deal with natural languages and natural scenes
  - AI complete: generally as intelligent as humans
- Some technologies for NLP toward AI complete
  - Grammars and parsing algorithms (For detail→ )
    - Use parsing algorithms to search among candidate “parse trees” with additional semantic and pragmatic information
      - Taking into account the context
      - Taking into account common-sense knowledge
  - Statistical NLP: probabilistic context-free grammars (PCFGs)
    - context-free grammars that assigned probabilities to the rules used to define a grammar

Statistical NLP

- Probabilistic Context-Free Grammars (PCFGs)
  - Not possible to provide an exact and complete characterization of well-formed utterances and ill-formed utterances
  - People always stretch and bend the ‘rules’ to meet their communicative needs
    - Example) “John shot elephants in pajamas”
      - John (while in pajamas) shot elephants.
      - John shot elephants (which were in pajamas).
  - In 1969, the automata theorist Taylor L. Booth
    - Proposed a variation on context-free grammars that assigned probabilities to the rules used to define a grammar

Probabilistic context-free grammars (PCFGs)

- Context-Free Rules with Probabilities

S -- NP VP (1.0)
PP -- P NP (1.0)
VP -- VP NP (0.7)
VP -- VP PP (0.3)
P -- *in* (1.0)
V -- *shot* (1.0)

NP -- NP PP (0.4)
NP -- *John* (0.1)
NP -- *pajamas* (0.18)
NP -- *shot* (0.04)
NP -- *elephants* (0.18)
NP -- *uniforms* (0.1)

Probability of the rule

- Probabilities of Parse Trees

$$Prob_{left} = 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0009072$$
$$Prob_{right} = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0006804$$

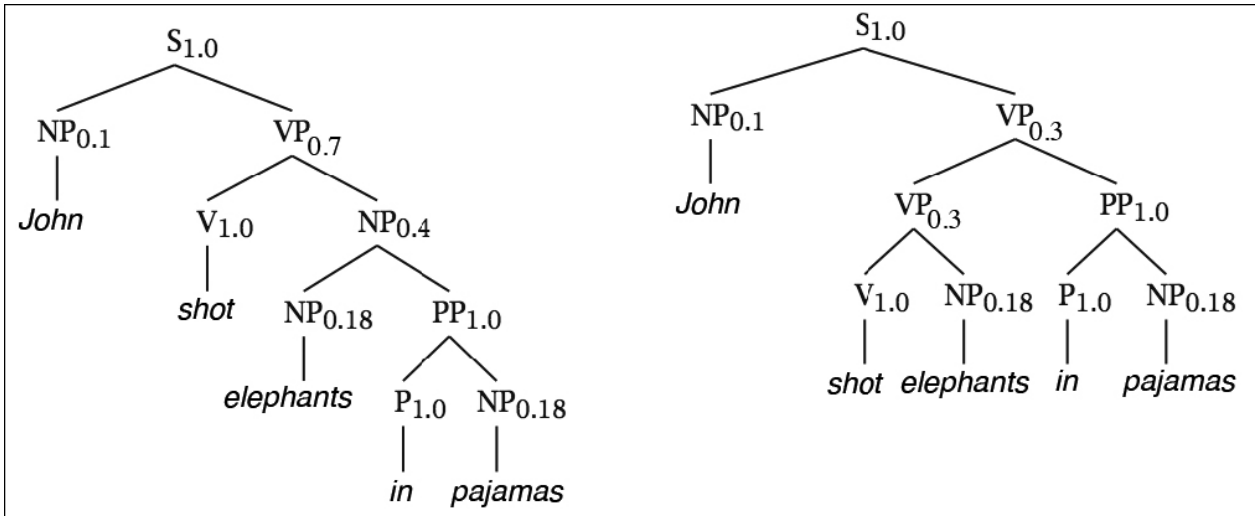


Figure 30.1: Two parse trees for “John shot elephants in pajamas."

Probabilistic context-free grammars (PCFGs)

- Learning PCFGs
  - Case of using tree bank
    - The parse trees use rules all of the form *l* -> *r*
      - *l* : left-hand side of rule (such as VP)
      - *r* : right-hand side of rule (such as VP NP).
      - Probability : # (*l* -> *r*) / # (*l*)
  - Case of using ordinary (nonprobabilistic) context-free parser
    - Expectation Maximization (EM) algorithm
      1. Convert the original CFG into a PCFG with equal rule probabilities.
      2. Parse the sentences with this PCFG, computing a probability for each ambiguous parse.
      3. Count the rules in each parse for each sentence, and weight the count by the probability of that parse.
      4. Use these weighted counts to compute new probabilities for the rules and thus a new PCFG.
      5. Repeat this process until the rule probabilities cease to change (which will happen eventually).

Chapter 30. Natural Languages and Natural Scenes

30.2 Computer Vision

Computer vision for natural scenes

- Recovering surfaces and depth information
  - Classifying surfaces of images into support, vertical (porous, solid) and sky surfaces (Hoiem)

    - Learning models from hand-labeled data

  - Extracting actual depth information and scene-structure information from monocular images
    - Andrew Ng's work

Computer vision for natural scenes

- Tracking moving objects using particle filtering
  - CONDENSATION 
    - Track outlines and features of foreground objects, modeled as curves, as they move in substantial clutter at almost video frame-rate
  - “particle filter” in action 
    - Tracking of a changing number of people using a *moving* robot’s laser range finder

Computer vision for natural scenes

- Hierarchical models
  - The raw pixels are aggregated spatially to form higher level grouping
  - Lower-level grouping are aggregated again into somewhat higher level components
  - Models based on Machine Leaning
    - Lee-Mumford model 
    - Deep belief network 
    - Hierarchical temporal memory (HTM) 
      - Developed based on Jeff Hawkins's idea
        - Layers in the neocortex store increasingly abstract representation of sensory input sequence
        - To make increasingly detailed predictions of future experience
        - This procedure also used for other sensory modalities (not only vision)
    - Deep architectures & Energy-based models (EBMs)
      - composed of multiple layers of trainable nonlinear modules

- Image grammars
  - Attempt to use grammars and syntactic analyses for processing pictures and images
  - Stochastic grammars of images 

Chapter 30. Natural Languages and Natural Scenes

Appendix

Natural Language Processing

- Growing need for Natural Language Process(NLP) systems
  - able to deal with massive data
  - Performing summarizing text, answering queries, translating languages

- Hard to achieve human level NLP
  - AI Complete
    - Generally as intelligent as humans
    - Being able to reason and to solve problems as well as humans do those things.

30.1.1 Grammars and Parsing Algorithms

- Grammars
  - Use parsing algorithms to search among candidate "parse trees"
    - Accept legal sentences or reject
    - Many possible parses in one sentence, each convey different meaning
      - Example

        "One morning I Shot an elephant in my pajamas.
        How he got into my pajamas I don't know."

    - For disambiguation, use semantic and pragmatic analyses
      - Taking into account the context
      - Taking into account common-sense knowledge

30.1.1 Grammars and Parsing Algorithms

- Grammars (Con'd)
  - Advance syntactic structure (introduced before)
    - Context-free grammars (CFGs)
    - Definite clause grammars (DCGs)
    - Systemic grammars
    - Transition network grammars
    - DIAGRAM
  - More advance syntactic structure
    - Lexical functional grammars (LFGs)
    - Tree adjoining grammars (TAGs)
    - Dependency grammars
    - Head-driven phrase structure grammars (HPSGs)
    - Government and binding
    - Categorical grammars

30.1.1 Grammars and Parsing Algorithms

- Parsing Algorithms
  - Breadth-first search and depth-first search need too much cost
  - Employ charts and other constructs to store already computed parses of segments of sentences for possible reuse
  - Chart-like parsers
    - Martin Kay's chart parser
    - Earley parser
    - Cocke-Younger-Kasami(CYK) algorithm

- Usable Corpora
  - Large files of sentences
  - Newspaper article, literary texts, and other materials
  - Notable Web Database
    - Web sites at Stanford
    - "Linguistic Data Consortium"
    - "Penn Treebanks"
  - Use statistically based machine learning techniques

30.1.2 Statistical NLP

- Probabilistic Context-Free Grammars (PCFGs)
  - Not possible to provide an exact and complete characterization of well-formed utterances and ill-formed utterances
  - People always stretch and bend the ‘rules’ to meet their communicative needs
    - Example) “John shot elephants in pajamas”
      - John (while in pajamas) shot elephants.
      - John shot elephants (which were in pajamas).
  - In 1969, the automata theorist Taylor L. Booth
    - Proposed a variation on context-free grammars that assigned probabilities to the rules used to define a grammar

30.1.2 Statistical NLP

- Context-Free Rules with Probabilities

S -- NP VP (1.0)
PP -- P NP (1.0)
VP -- VP NP (0.7)
VP -- VP PP (0.3)
P -- *in* (1.0)
V -- *shot* (1.0)

Probability of the rule

NP -- NP PP (0.4)
NP -- *John* (0.1)
NP -- *pajamas* (0.18)
NP -- *shot* (0.04)
NP -- *elephants* (0.18)
NP -- *uniforms* (0.1)

- Probabilities of Parse Trees

$$Prob_{left} = 1.0 \times 0.1 \times 0.7 \times 1.0 \times 0.4 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0009072$$
$$Prob_{right} = 1.0 \times 0.1 \times 0.3 \times 0.7 \times 1.0 \times 0.18 \times 1.0 \times 1.0 \times 0.18 = 0.0006804$$

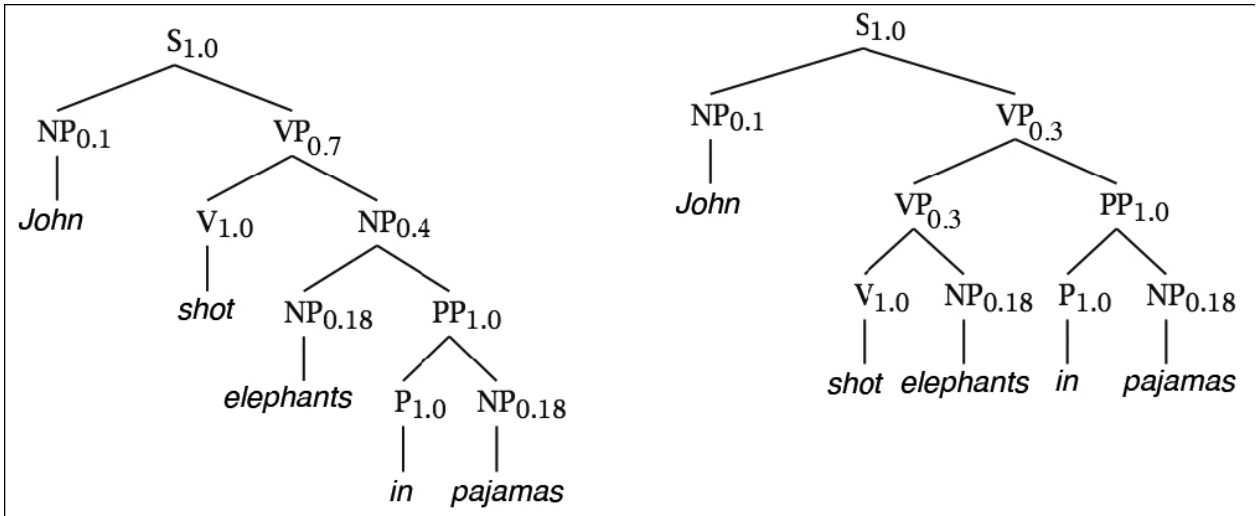


Figure 30.1: Two parse trees for “John shot elephants in pajamas."

30.1.2 Statistical NLP

- Learning PCFGs
  - Case of using tree bank
    - The parse trees use rules all of the form *l* -> *r*
      - *l* : left-hand side of rule (such as VP)
      - *r* : right-hand side of rule (such as VP NP).
      - Probability : # (*l* -> *r*) / # (*l*)
  - Case of using ordinary (nonprobabilistic) context-free parser
    - Expectation Maximization (EM) algorithm
      1. Convert the original CFG into a PCFG with equal rule probabilities.
      2. Parse the sentences with this PCFG, computing a probability for each ambiguous parse.
      3. Count the rules in each parse for each sentence, and weight the count by the probability of that parse.
      4. Use these weighted counts to compute new probabilities for the rules and thus a new PCFG.
      5. Repeat this process until the rule probabilities cease to change (which will happen eventually).

30.1.2 Statistical NLP

- Learning PCFGs (Con'd)
  - Data-Oriented Parsing (DOP)
    - Based on the idea
      - "human language perception and production work with representations of concrete language experiences, rather than with abstract grammatical rules"
  - Lexical Functional Grammars (LFGs):
    - Enhance grammar by using DOP ideas
  - Kaplan and his group's method
    - Trying to learn how to assign probability orderings to the multiple parse trees of a sentence
  - n-grams
    - Using data about how frequently certain combinations of words occur in various text sources.
    - "just now" – 2-gram
    - "put it on the shelf" – 5-gram

Modern Computer Vision

- Ideas three decades ago, usability of large amounts of data and computational resources can be applied now
  - Region and boundary descriptors
  - Superpixels
  - Combining bottom-up and top-down processing
  - Bayesian formulation
  - Feature selection
- Contributions from several other fields
  - Optics, mathematics, computer graphics, electrical engineering, physics, neuroscience, and statistics
  - Machine learning

30.2.1 Recovering Surface and Depth Information

- Hoiem, Efros and Hebert's work at CMU
  - Classify regions of an image into one of three major surface categories
    - Support surface
      - parallel to the ground and could support an object: road, lawns, lakes…
    - Vertical surfaces
      - steep support surface: walls, cliffs, the curb sides, people, trees or cows…
    - Sky surfaces
      - open air and clouds
  - Classify each vertical surface into one of the following subclasses
    - Planar surfaces
      - Facing to the 'left,' 'center,' or 'right'
      - Vertical surfaces that are roughly planar.
    - Nonplanar surfaces
      - 'porous' or 'solid'
      - Porous surface: Not have a solid continuous surface: tree leaves, shrubs, telephone wires, chain link fences, …
  - Learning
    - Groups of adjacent pixels in each of the training images in this set were assembled into nearly uniform regions (superpixels)
    - Assigning labels on superpixels or segments manually

Hoiem, Efros and Hebert's work at CMU

Figure 30.2: Typical outdoor images.

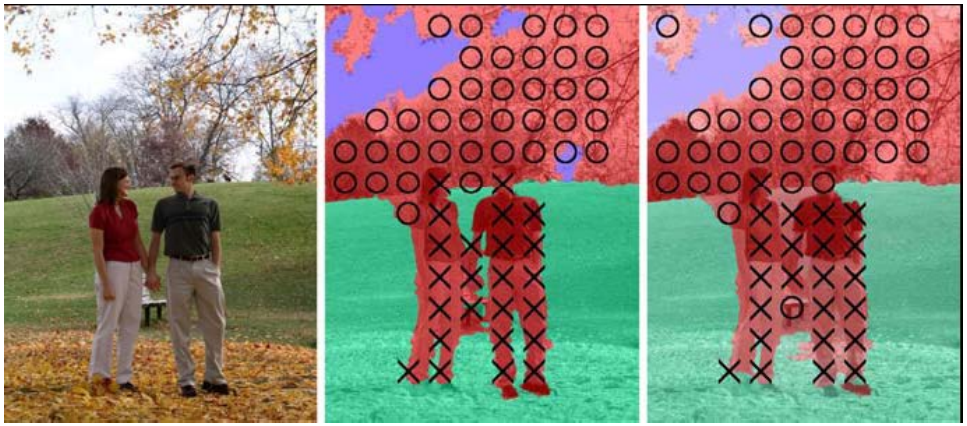


Figure 30.3: Original image (left), hand-labeled image (center), and system's output (right).
"O" for porous surfaces, and "X" for solid surfaces.

30.2.1 Recovering Surface and Depth Information

- Andrew Ng's work at Stanford
  - Extracting actual depth information and scene-structure information from monocular images
  - Outline
    - Dataset
      - 'Ground truth' depth information for a set of training images by a 3-D laser scanner
    - Learning
      - match its estimates of depth against ground-truth depth using several image features
    - Model: multiscale Markov random field network
      - to represent the relationships between the depth of an image patch and the depths of neighboring patches
    - Features
      - Texture variations, texture gradients, color, and occlusion information
      - Because depth information about close objects is captured at larger scales than that of distant objects, features are extracted at multiple image scales.

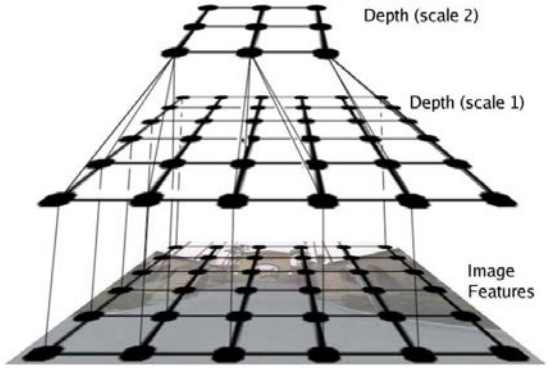


Figure 30.4: A multiscale Markov random field network

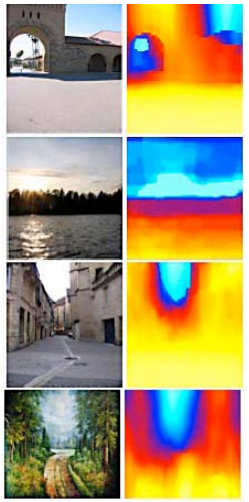


Figure 30.5: Typical images (left) and predicted depth maps (right)

30.2.2 Tracking Moving Objects

- CONDENSATION (CONditional DENSity PropagATION)
  - Able to track outlines and features of foreground objects, modeled as curves, as they move in substantial clutter at almost video frame-rate
  - Model: dynamic Bayesian network
  - Particle filtering
    - represents the probability of an outlining curve by a large set of weighted samples (particles) of outlines

Figure 30.7: Tracking a leaf in the wind.

Michael Isard (left)
Andrew Blake (right)

Look the movie at
http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/ISARD1/images/leafmv.mpg.

30.2.2 Tracking Moving Objects

- "particle filter" in action (D. Fox and colleagues)
  - Tracking of a changing number of people using a *moving* robot's laser range finder
  - Using particle filter

Figure 30.8: Tracking multiple people.
Look the movie at
http://www.cs.washington.edu/ai/Mobile_Robotics/mcl/animations/floor3D.avi

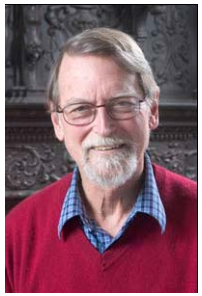
30.2.2 Tracking Moving Objects

- Dickmanns' work
  - The 1st significant real world application of computer vision
  - Vision and control system for driverless automobile
  - "4-D" approach
    - Able to detect and track adjacent vehicles using spatio-temporal models of object motion
  - The first group to use Kalman filtering for visual object tracking
- Malik's object-tracking research
  - Tracking soccer players and automobiles
- Leeds' work
  - Goal
    - Improving object-tracking accuracy by reasoning about "fundamental constraints on the spatio-temporal continuity of objects."

30.2.3 Hierarchical Models

- One of the potentially most promising development in computer vision (and maybe in AI)
- Procedure
  - The raw pixels are aggregated spatially to form higher level grouping
  - Lower-level grouping are aggregated again into somewhat higher level components
- Features
  - The aggregation at the various levels are learned using massive data sets – not predesigned by hand
  - Occurrences of aggregations at each level are qualified by probabilities with probabilistic graphical models
  - The probabilities of aggregations at one level can affect not only same level but other level
- Motivated mainly by attempts to model the storage and inference mechanisms in the visual cortex of humans and primates

30.2.3 Hierarchical Models



Tai Sing Lee (left)
David Mumford (right)

- Lee-Mumford Model
  - Process
    - “bottom-up” visual observation
    - “top-down” hypotheses
    - Feed-forward-feedback process
  - Problem
    - None of levels can be completely sure of its interpretation there might be multiple high-probability global interpretations
    - They didn’t implement their model
      - choice of model: unconstrained
      - simulation result: provides only weak support for a high-level hypothesis
    - Just cite neurophysiological and psychophysical evidence supporting model

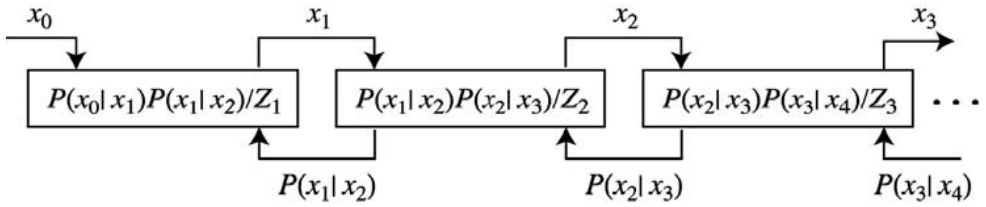


Figure 30.9. (bottom): Their layers of visual processing

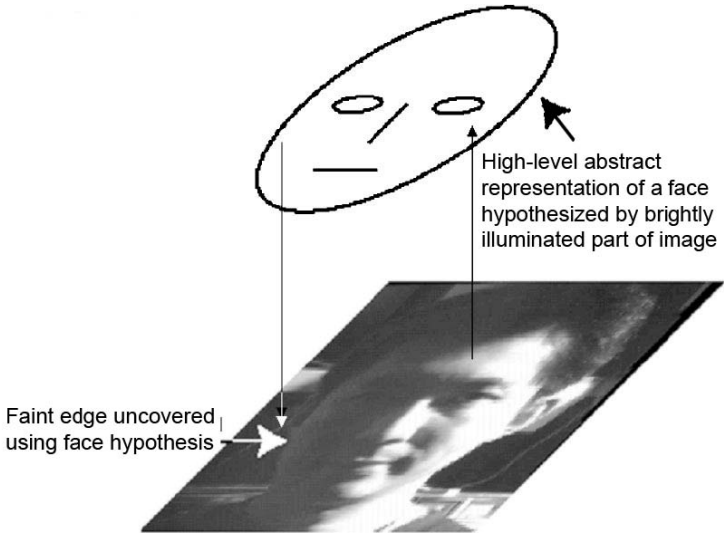


Figure 30.10:
Seeing a face more clearly

30.2.3 Hierarchical Models

Geoffrey Hinton

- Deep Belief Network (Hinton, Osindero, Teh)
  - Overall structure: a layered neural network
  - Greed method of training
    - Training proceeds from the bottom in steps, level by level
    - As each level is trained, its weights are 'frozen'
    - The results are used as inputs for training the next higher level
  - Handwritten digits experiment
    - Successful discrimination ability
    - Successful generation ability

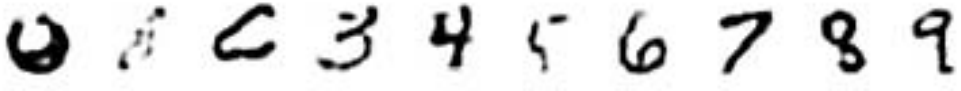


Figure 30.12: Some images generated by the trained network

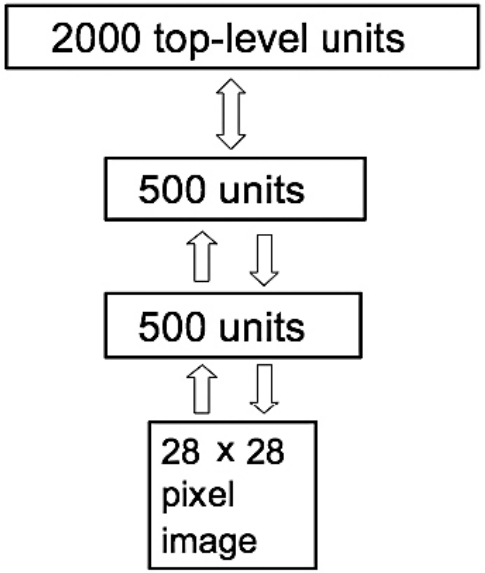


Figure 30.11. (right) : The deep belief network

30.2.3 Hierarchical Models

- Hierarchical Temporal Memory (HTM)
  - Jeff Hawkins' idea
    - neocortex is a hierarchical temporal memory
    - whose layers store increasingly abstract representation of sensory input sequence
    - To make increasingly detailed predictions of future experience.
    - Hierarchical memory and its learning procedures are used not only for visual input but for other sensory modalities as well.
  - HTM by above idea
    - Developed by Dileep George
    - Procedure
      - The nodes in each layer are trained to recognize commonly occurring sequences in its receptive field in the layer below.
      - After training, the probabilities of the sequences represented at each level are conditioned by feedback from above.

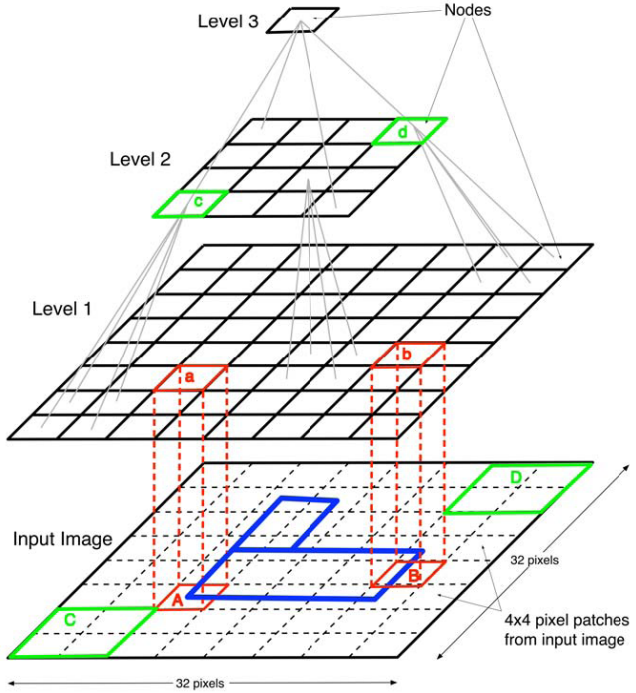


Figure 30.13: The HTM model

30.2.3 Hierarchical Models

- Other works
  - Poggio's work
    - Apply mathematical and statistical learning mechanisms to help model how the brain learns to recognize visual objects.
  - LeCun's work
    - Study "Deep architectures" and Energy-based models (EBMs)
      - Composed of multiple layers of trainable nonlinear modules

30.2.4 Image Grammars

- Attempt to use grammars and syntactic analyses for processing pictures and images

- Works
  - Mimicking painter's drawing (Russell Kirsch)
    - Develop a grammar for analyzing (and producing) pictures
  - Picture grammars (Azriel Rosenfeld)

Stochastic grammars of images

- Song-Chun Zhu
- decomposing an image into component parts
- representing as a parse tree

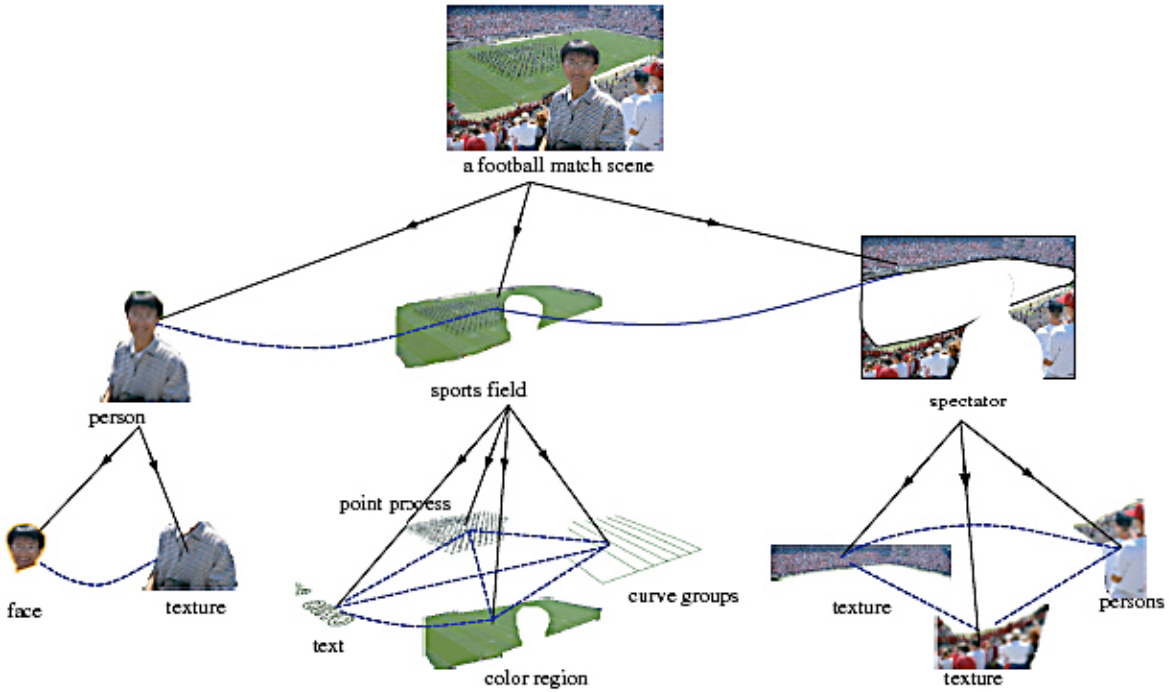


Figure 30.14: Parsing an image.