Pattern recognition

This article is about pattern recognition as a branch of machine learning. For the cognitive process, see Pattern recognition (psychology). For other uses, see Pattern recognition (disambiguation).

Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data, although it is in some cases considered to be nearly synonymous with machine learning.[1] Pattern recognition systems are in many cases trained from labeled "training" data (supervised learning), but when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning).

The terms pattern recognition, machine learning, data mining and knowledge discovery in databases (KDD) are hard to separate, as they largely overlap in their scope. Machine learning is the common term for supervised learning methods and originates from artificial intelligence, whereas KDD and data mining have a larger focus on unsupervised methods and stronger connection to business use. Pattern recognition has its origins in engineering, and the term is popular in the context of computer vision: a leading computer vision conference is named Conference on Computer Vision and Pattern Recognition. In pattern recognition, there may be a higher interest to formalize, explain and visualize the pattern, while machine learning traditionally focuses on maximizing the recognition rates. Yet, all of these domains have evolved substantially from their roots in artificial intelligence, engineering and statistics, and they've become increasingly similar by integrating developments and ideas from each other.

In machine learning, **pattern recognition** is the assignment of a label to a given input value. In statistics, discriminant analysis was introduced for this same purpose in 1936. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of *classes* (for example, determine whether a given email is "spam" or "non-spam"). However, pattern recognition is a more general problem that encompasses other types of output as well. Other examples are regression, which assigns a real-valued output to each input; sequence labeling, which assigns a class to each member of a sequence of values (for example, part of speech tagging, which assigns a part of speech to each word in an input sentence); and parsing, which assigns a parse tree to an input sentence, describing the syntactic structure of the sentence.

Pattern recognition algorithms generally aim to provide a reasonable answer for all possible inputs and to perform "most likely" matching of the inputs, taking into account their statistical variation. This is opposed to *pattern matching* algorithms, which look for exact matches in the input with pre-existing patterns. A common example of a pattern-matching algorithm is regular expression matching, which looks for patterns of a given sort in textual data and is included in the search capabilities of many text editors and word processors. In contrast to pattern recognition, pattern matching is generally not considered a type of machine learning, although pattern-matching algorithms (especially with fairly general, carefully tailored patterns) can sometimes succeed in providing similar-quality output of the sort provided by pattern-recognition algorithms.

1 Overview

Pattern recognition is generally categorized according to the type of learning procedure used to generate the output value. *Supervised learning* assumes that a set of *training data* (the *training set*) has been provided, consisting of a set of instances that have been properly labeled by hand with the correct output. A learning procedure then generates a *model* that attempts to meet two sometimes conflicting objectives: Perform as well as possible on the training data, and generalize as well as possible to new data (usually, this means being as simple as possible, for some technical definition of "simple", in accordance with Occam's Razor, discussed below). Unsupervised learning, on the other hand, assumes training data that has not been hand-labeled, and attempts to find inherent patterns in the data that can then be used to determine the correct output value for new data instances.[2] A combination of the two that has recently been explored is semi-supervised learning, which uses a combination of labeled and unlabeled data (typically a small set of labeled data combined with a large amount of unlabeled data). Note that in cases of unsupervised learning, there may be no training data at all to speak of; in other words, the data to be labeled *is* the training data.

Note that sometimes different terms are used to describe the corresponding supervised and unsupervised learning procedures for the same type of output. For example, the unsupervised equivalent of classification is normally known as *clustering*, based on the common perception of the task as involving no training data to speak of, and

of grouping the input data into *clusters* based on some inherent similarity measure (e.g. the distance between instances, considered as vectors in a multi-dimensional vector space), rather than assigning each input instance into one of a set of pre-defined classes. Note also that in some fields, the terminology is different: For example, in community ecology, the term "classification" is used to refer to what is commonly known as "clustering".

The piece of input data for which an output value is generated is formally termed an *instance*. The instance is formally described by a vector of *features*, which together constitute a description of all known characteristics of the instance. (These feature vectors can be seen as defining points in an appropriate multidimensional space, and methods for manipulating vectors in vector spaces can be correspondingly applied to them, such as computing the dot product or the angle between two vectors.) Typically, features are either categorical (also known as nominal, i.e., consisting of one of a set of unordered items, such as a gender of "male" or "female", or a blood type of "A", "B", "AB" or "O"), ordinal (consisting of one of a set of ordered items, e.g., "large", "medium" or "small"), integer-valued (e.g., a count of the number of occurrences of a particular word in an email) or real-valued (e.g., a measurement of blood pressure). Often, categorical and ordinal data are grouped together; likewise for integer-valued and real-valued data. Furthermore, many algorithms work only in terms of categorical data and require that real-valued or integer-valued data be *discretized* into groups (e.g., less than 5, between 5 and 10, or greater than 10).

1.1 Probabilistic classifiers

Main article: Probabilistic classifier

Many common pattern recognition algorithms are *probabilistic* in nature, in that they use statistical inference to find the best label for a given instance. Unlike other algorithms, which simply output a "best" label, often probabilistic algorithms also output a probability of the instance being described by the given label. In addition, many probabilistic algorithms output a list of the *N*-best labels with associated probabilities, for some value of *N*, instead of simply a single best label. When the number of possible labels is fairly small (e.g., in the case of classification), *N* may be set so that the probability of all possible labels is output. Probabilistic algorithms have many advantages over non-probabilistic algorithms:

- They output a confidence value associated with their choice. (Note that some other algorithms may also output confidence values, but in general, only for probabilistic algorithms is this value mathematically grounded in probability theory. Non-probabilistic confidence values can in general not be given any specific meaning, and only used to compare against other confidence values output by the same algorithm.)
- Correspondingly, they can *abstain* when the confidence of choosing any particular output is too low.
- Because of the probabilities output, probabilistic pattern-recognition algorithms can be more effectively incorporated into larger machine-learning tasks, in a way that partially or completely avoids the problem of *error propagation*.

1.2 Number of important feature variables

Feature selection algorithms attempt to directly prune out redundant or irrelevant features. A general introduction to feature selection which summarizes approaches and challenges, has been given.[3] The complexity of feature-selection is, because of its non-monotonous character, an optimization problem where given a total of n features the powerset consisting of all $2^n - 1$ subsets of features need to be explored. The Branch-and-Bound algorithm[4] does reduce this complexity but is intractable for medium to large values of the number of available features n . For a large-scale comparison of feature-selection algorithms see .[5]

Techniques to transform the raw feature vectors (**feature extraction**) are sometimes used prior to application of the pattern-matching algorithm. For example, feature extraction algorithms attempt to reduce a large-dimensionality feature vector into a smaller-dimensionality vector that is easier to work with and encodes less redundancy, using mathematical techniques such as principal components analysis (PCA). The distinction between **feature selection** and **feature extraction** is that the resulting features after feature extraction has taken place are of a different sort than the original features and may not easily be interpretable, while the features left after feature selection are simply a subset of the original features.

2 Problem statement (supervised version)

Formally, the problem of supervised pattern recognition can be stated as follows: Given an unknown function $g : \mathcal{X} \to \mathcal{Y}$ (the *ground truth*) that maps input instances $\boldsymbol{x} \in \mathcal{X}$ to output labels $y \in \mathcal{Y}$, along with training data $\mathbf{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ assumed to represent accurate examples of the mapping, produce a function $h : \mathcal{X} \to \mathcal{Y}$ that approximates as closely as possible the correct mapping g . (For example, if the problem is filtering spam, then $\boldsymbol{x}_i$ is some representation of an email and y is either "spam" or "non-spam"). In order for this to be a well-defined problem, "approximates as closely as possible" needs to be defined rigorously. In decision theory,

this is defined by specifying a loss function that assigns a specific value to "loss" resulting from producing an incorrect label. The goal then is to minimize the expected loss, with the expectation taken over the probability distribution of $\mathcal{X}$. In practice, neither the distribution of $\mathcal{X}$ nor the ground truth function $g : \mathcal{X} \to \mathcal{Y}$ are known exactly, but can be computed only empirically by collecting a large number of samples of $\mathcal{X}$ and hand-labeling them using the correct value of $\mathcal{Y}$ (a time-consuming process, which is typically the limiting factor in the amount of data of this sort that can be collected). The particular loss function depends on the type of label being predicted. For example, in the case of classification, the simple zero-one loss function is often sufficient. This corresponds simply to assigning a loss of 1 to any incorrect labeling and implies that the optimal classifier minimizes the error rate on independent test data (i.e. counting up the fraction of instances that the learned function $h : \mathcal{X} \to \mathcal{Y}$ labels wrongly, which is equivalent to maximizing the number of correctly classified instances). The goal of the learning procedure is then to minimize the error rate (maximize the correctness) on a "typical" test set.

For a probabilistic pattern recognizer, the problem is instead to estimate the probability of each possible output label given a particular input instance, i.e., to estimate a function of the form

$$p(\text{label}|\boldsymbol{x}, \boldsymbol{\theta}) = f\left(\boldsymbol{x}; \boldsymbol{\theta}\right)$$

where the feature vector input is $\boldsymbol{x}$, and the function f is typically parameterized by some parameters $\boldsymbol{\theta}$.[6] In a discriminative approach to the problem, f is estimated directly. In a generative approach, however, the inverse probability $p(\boldsymbol{x}|\text{label})$ is instead estimated and combined with the prior probability $p(\text{label}|\boldsymbol{\theta})$ using Bayes' rule, as follows:

$$p(\text{label}|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}|\text{label}, \boldsymbol{\theta})p(\text{label}|\boldsymbol{\theta})}{\sum_{L \in \text{labels all}} p(\boldsymbol{x}|L)p(L|\boldsymbol{\theta})}.$$

When the labels are continuously distributed (e.g., in regression analysis), the denominator involves integration rather than summation:

$$p(\text{label}|\boldsymbol{x}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{x}|\text{label}, \boldsymbol{\theta})p(\text{label}|\boldsymbol{\theta})}{\int_{L \in \text{labels all}} p(\boldsymbol{x}|L)p(L|\boldsymbol{\theta}) \,\mathrm{d}\, L}.$$

The value of $\boldsymbol{\theta}$ is typically learned using maximum a posteriori (MAP) estimation. This finds the best value that simultaneously meets two conflicting objects: To perform as well as possible on the training data (smallest error-rate) and to find the simplest possible model. Essentially, this combines maximum likelihood estimation with a regularization procedure that favors simpler models over more complex models. In a Bayesian context, the regularization procedure can be viewed as placing a prior probability $p(\boldsymbol{\theta})$ on different values of $\boldsymbol{\theta}$. Mathematically:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathbf{D})$$

where $\boldsymbol{\theta}^*$ is the value used for $\boldsymbol{\theta}$ in the subsequent evaluation procedure, and $p(\boldsymbol{\theta}|\mathbf{D})$, the posterior probability of $\boldsymbol{\theta}$, is given by

$$p(\boldsymbol{\theta}|\mathbf{D}) = \left[\prod_{i=1}^{n} p(y_i|\boldsymbol{x}_i, \boldsymbol{\theta})\right] p(\boldsymbol{\theta}).$$

In the Bayesian approach to this problem, instead of choosing a single parameter vector $\boldsymbol{\theta}^*$, the probability of a given label for a new instance $\boldsymbol{x}$ is computed by integrating over all possible values of $\boldsymbol{\theta}$, weighted according to the posterior probability:

$$p(\text{label}|\boldsymbol{x}) = \int p(\text{label}|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{D}) \,\mathrm{d}\, \boldsymbol{\theta}.$$

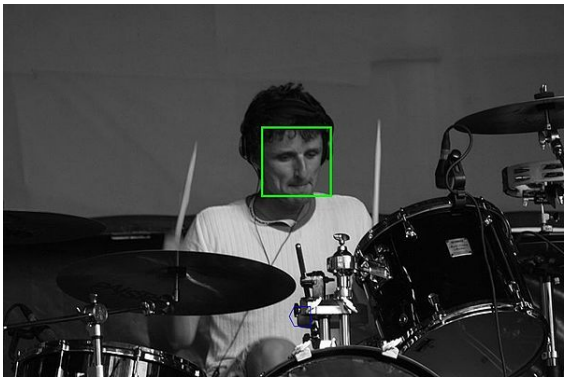
2.1 Frequentist or Bayesian approach to pattern recognition

The first pattern classifier – the linear discriminant presented by Fisher – was developed in the Frequentist tradition. The frequentist approach entails that the model parameters are considered unknown, but objective. The parameters are then computed (estimated) from the collected data. For the linear discriminant, these parameters are precisely the mean vectors and the Covariance matrix. Also the probability of each class $p(\text{label}|\boldsymbol{\theta})$ is estimated from the collected dataset. Note that the usage of 'Bayes rule' in a pattern classifier does not make the classification approach Bayesian.

Bayesian statistics has its origin in Greek philosophy where a distinction was already made between the 'a priori' and the 'a posteriori' knowledge. Later Kant defined his distinction between what is a priori known – before observation – and the empirical knowledge gained from observations. In a Bayesian pattern classifier, the class probabilities $p(\text{label}|\boldsymbol{\theta})$ can be chosen by the user, which are then a priori. Moreover, experience quantified as a priori parameter values can be weighted with empirical observations – using e.g., the Beta- (conjugate prior) and Dirichlet-distributions. The Bayesian approach facilitates a seamless intermixing between expert knowledge in the form of subjective probabilities, and objective observations.

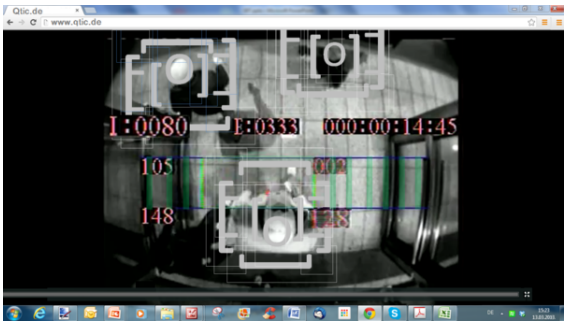
Probabilistic pattern classifiers can be used according to a frequentist or a Bayesian approach.

3 Uses



The face was automatically detected by special software.

Within medical science, pattern recognition is the basis for computer-aided diagnosis (CAD) systems. CAD describes a procedure that supports the doctor's interpretations and findings.



Pattern & Shape Recognition Technology (SRT) in a people counter system

Other typical applications of pattern recognition techniques are automatic speech recognition, classification of text into several categories (e.g., spam/non-spam email messages), the automatic recognition of handwritten postal codes on postal envelopes, automatic recognition of images of human faces, or handwriting image extraction from medical forms.[7] The last two examples form the subtopic image analysis of pattern recognition that deals with digital images as input to pattern recognition systems.[8][9]

Optical character recognition is a classic example of the application of a pattern classifier, see OCR-example. The method of signing one's name was captured with stylus and overlay starting in 1990. The strokes, speed, relative min, relative max, acceleration and pressure is used to uniquely identify and confirm identity. Banks were first offered this technology, but were content to collect from the FDIC for any bank fraud and did not want to inconvenience customers..

Artificial neural networks (neural net classifiers) and Deep Learning have many real-world applications in image processing, a few examples:

- identification and authentication: e.g., license plate recognition,[10] fingerprint analysis and face detection/verification;[11]
- medical diagnosis: e.g., screening for cervical cancer (Papnet)[12] or breast tumors;
- defence: various navigation and guidance systems, target recognition systems, shape recognition technology etc.

For a discussion of the aforementioned applications of neural networks in image processing, see e.g.[13]

In psychology, pattern recognition (making sense of and identifying objects) is closely related to perception, which explains how the sensory inputs humans receive are made meaningful. Pattern recognition can be thought of in two different ways: the first being template matching and the second being feature detection. A template is a pattern used to produce items of the same proportions. The template-matching hypothesis suggests that incoming stimuli are compared with templates in the long term memory. If there is a match, the stimulus is identified. Feature detection models, such as the Pandemonium system for classifying letters (Selfridge, 1959), suggest that the stimuli are broken down into their component parts for identification. For example, a capital E has three horizontal lines and one vertical line.[14]

4 Algorithms

Algorithms for pattern recognition depend on the type of label output, on whether learning is supervised or unsupervised, and on whether the algorithm is statistical or non-statistical in nature. Statistical algorithms can further be categorized as generative or discriminative.

4.1 Classification algorithms (supervised algorithms predicting categorical labels)

Main article: Statistical classification

Parametric:[15]

- Linear discriminant analysis
- Quadratic discriminant analysis
- Maximum entropy classifier (aka logistic regression, multinomial logistic regression): Note that logistic regression is an algorithm for classification, despite its name. (The name comes from the fact that logistic regression uses an extension of a linear regression model to model the probability of an input being in a particular class.)

Nonparametric:[16]

- Decision trees, decision lists
- Kernel estimation and K-nearest-neighbor algorithms
- Naive Bayes classifier
- Neural networks (multi-layer perceptrons)
- Perceptrons
- Support vector machines
- Gene expression programming

4.2 Clustering algorithms (unsupervised algorithms predicting categorical labels)

Main article: Cluster analysis

- Categorical mixture models
- Deep learning methods
- Hierarchical clustering (agglomerative or divisive)
- K-means clustering
- Correlation clustering
- Kernel principal component analysis (Kernel PCA)

4.3 Ensemble learning algorithms (supervised meta-algorithms for combining multiple learning algorithms together)

Main article: Ensemble learning

- Boosting (meta-algorithm)
- Bootstrap aggregating ("bagging")
- Ensemble averaging
- Mixture of experts, hierarchical mixture of experts

4.4 General algorithms for predicting arbitrarily-structured (sets of) labels

- Bayesian networks
- Markov random fields

4.5 Multilinear subspace learning algorithms (predicting labels of multidimensional data using tensor representations)

Unsupervised:

- Multilinear principal component analysis (MPCA)

4.6 Real-valued sequence labeling algorithms (predicting sequences of real-valued labels)

Main article: sequence labeling

Supervised (?):

- Kalman filters
- Particle filters

4.7 Regression algorithms (predicting real-valued labels)

Main article: Regression analysis

Supervised:

- Gaussian process regression (kriging)
- Linear regression and extensions
- Neural networks and Deep learning methods

Unsupervised:

- Independent component analysis (ICA)
- Principal components analysis (PCA)

4.8 Sequence labeling algorithms (predicting sequences of categorical labels)

Supervised:

- Conditional random fields (CRFs)
- Hidden Markov models (HMMs)
- Maximum entropy Markov models (MEMMs)
- Recurrent neural networks

Unsupervised:

- Hidden Markov models (HMMs)

5 See also

- Adaptive resonance theory
- Black box
- Cache language model
- Compound term processing
- Computer-aided diagnosis
- Data mining
- Deep Learning
- List of numerical analysis software
- List of numerical libraries
- Machine learning
- Multilinear subspace learning
- Neocognitron
- Perception
- Perceptual learning
- Predictive analytics
- Prior knowledge for pattern recognition
- Sequence mining
- Template matching
- Contextual image classification
- List of datasets for machine learning research

6 References

This article is based on material taken from the Free Online Dictionary of Computing prior to 1 November 2008 and incorporated under the "relicensing" terms of the GFDL, version 1.3 or later.

[1] Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer. p. vii. Pattern recognition has its origins in engineering, whereas machine learning grew out of computer science. However, these activities can be viewed as two facets of the same field, and together they have undergone substantial development over the past ten years.

[2] Carvalko, J.R., Preston K. (1972). "On Determining Optimum Simple Golay Marking Transforms for Binary Image Processing". *IEEE Transactions on Computers* **21**: 1430–33. doi:10.1109/T-C.1972.223519..

[3] Isabelle Guyon Clopinet, André Elisseeff (2003). *An Introduction to Variable and Feature Selection*. The Journal of Machine Learning Research, Vol. 3, 1157-1182. Link

[4] Iman Foroutan; Jack Sklansky (1987). "Feature Selection for Automatic Classification of Non-Gaussian Data". *IEEE Transactions on Systems, Man and Cybernetics* **17** (2): 187–198. doi:10.1109/TSMC.1987.4309029..

[5] Mineichi Kudo; Jack Sklansky (2000). "Comparison of algorithms that select features for pattern classifiers". *Pattern Recognition* **33** (1): 25–41. doi:10.1016/S0031-3203(99)00041-2..

[6] For linear discriminant analysis the parameter vector $\boldsymbol{\theta}$ consists of the two mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ and the common covariance matrix $\boldsymbol{\Sigma}$.

[7] Milewski, Robert; Govindaraju, Venu (31 March 2008). "Binarization and cleanup of handwritten text from carbon copy medical form images". *Pattern Recognition* **41** (4): 1308–1315. doi:10.1016/j.patcog.2007.08.018.

[8] Richard O. Duda, Peter E. Hart, David G. Stork (2001) *Pattern classification* (2nd edition), Wiley, New York, ISBN 0-471-05669-3

[9] R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*, Wiley, ISBN 978-0-470-51706-2, 2009

[10] THE AUTOMATIC NUMBER PLATE RECOGNITION TUTORIAL http://anpr-tutorial.com/

[11] Neural Networks for Face Recognition Companion to Chapter 4 of the textbook Machine Learning.

[12] PAPNET For Cervical Screening http://health-asia.org/papnet-for-cervical-screening/

[13] Egmont-Petersen, M., de Ridder, D., Handels, H. (2002). "Image processing with neural networks - a review". *Pattern Recognition* **35** (10): 2279–2301. doi:10.1016/S0031-3203(01)00178-9.

[14] "A-level Psychology Attention Revision - Pattern recognition | S-cool, the revision website". S-cool.co.uk. Retrieved 2012-09-17.

[15] Assuming known distributional shape of feature distributions per class, such as the Gaussian shape.

[16] No distributional assumption regarding shape of feature distributions per class.

7 Further reading

- Fukunaga, Keinosuke (1990). *Introduction to Statistical Pattern Recognition* (2nd ed.). Boston: Academic Press. ISBN 0-12-269851-7.
- Koutroumbas, Konstantinos; Theodoridis, Sergios (2008). *Pattern Recognition* (4th ed.). Boston: Academic Press. ISBN 1-59749-272-8.
- Hornegger, Joachim; Paulus, Dietrich W. R. (1999). *Applied Pattern Recognition: A Practical Introduction to Image and Speech Processing in C++* (2nd ed.). San Francisco: Morgan Kaufmann Publishers. ISBN 3-528-15558-2.

- Schuermann, Juergen (1996). *Pattern Classification: A Unified View of Statistical and Neural Approaches*. New York: Wiley. ISBN 0-471-13534-8.
- Godfried T. Toussaint, ed. (1988). *Computational Morphology*. Amsterdam: North-Holland Publishing Company.
- Kulikowski, Casimir A.; Weiss, Sholom M. (1991). *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Machine Learning. San Francisco: Morgan Kaufmann Publishers. ISBN 1-55860-065-5.
- Jain, Anil.K.; Duin, Robert.P.W.; Mao, Jianchang (2000). "Statistical pattern recognition: a review". *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (1): 4–37. doi:10.1109/34.824819.
- An introductory tutorial to classifiers (introducing the basic terms, with numeric example)

8 External links

- The International Association for Pattern Recognition
- List of Pattern Recognition web sites
- Journal of Pattern Recognition Research
- Pattern Recognition Info
- Pattern Recognition (Journal of the Pattern Recognition Society)
- International Journal of Pattern Recognition and Artificial Intelligence
- International Journal of Applied Pattern Recognition
- Open Pattern Recognition Project, intended to be an open source platform for sharing algorithms of pattern recognition

9 Text and image sources, contributors, and licenses

9.1 Text

- **Pattern recognition** *Source:* https://en.wikipedia.org/wiki/Pattern_recognition?oldid=723738699 *Contributors:* Mav, The Anome, B4hand, Patrick, Michael Hardy, Lexor, Kku, Zeno Gantner, Karada, Rethunk, Pagingmrherman, DavidWBrooks, Hike395, Novum, Guaka, Dysprosia, Pir, Finlay McWalter, Robbot, Hobbes~enwiki, Benwing, Peak, Yosri, KellyCoinGuy, Ojigiri~enwiki, Ramir, Giftlite, Crimson30, Msm, Dratman, Utcursch, LiDaobing, Gene s, Xezbeth, Antaeus Feldspar, Violetriga, Vdm, Sludge, Basilwhite, Storm Rider, Denoir, Tabor, Dirac1933, Recury, Mcsee, Ruud Koot, Bluemoose, Mandarax, Qwertyus, Quiddity, TheRingess, Nandesuka, Intgr, Predictor, Kri, Chobot, Bobdc, Bgwhite, Vyroglyph, YurikBot, Wavelength, RobotE, Rsrikanth05, Wiki alf, Daniel Mietchen, Who-is-me, Mebden, Masroor, SmackBot, Offput, Madwyn, CommodiCast, Eskimbot, Trebor, Fuzzform, DHN-bot~enwiki, Dfletter, JonHarder, Memming, Cybercobra, Special-T, Feureau, Dicklyon, Izik~enwiki, Hu12, Mrdthree, GerryWolff, CmdrObot, Peptidefarmer, Pgr94, Krauss, Dancter, Tuxide, Thijs!bot, Eco84, Electron9, Gioto, AnAj, Shoejar, SHCarter, Swpb, TARBOT, Robotman1974, David Eppstein, Pmbhagat, Matthias Röder, Lauerfab, MartinBot, Anaxial, Maurice Carbonaro, Mikael Häggström, AntiSpamBot, Kraftlos, RJASE1, Redgecko, Oshwah, Rvencio, Guillaume2303, Echauz, Wikipedie, Sebastjanmm, SieBot, BotMultichill, ToePeu.bot, Cmbishop, Jbmurray, Sanya3, Maelgwnbot, Melcombe, Martarius, ClueBot, Justin W Smith, M4gnum0n, Wprlh, Carriearchdale, Ireny, Qwfp, Kamoshirenai, WikHead, P.r.newman, Addbot, Fgnievinski, MrOllie, Jarble, Movado73, Luckas-bot, Yobot, AnomieBOT, 1exec1, Jim1138, Materialscientist, Devantheryv, ArthurBot, Gtfjbl, DSisyphBot, Omnipaedista, Shirik, Krabeert, FrescoBot, Mycotoxin, Pinethicket, Ironman28, Tanja-Else, Lotje, Fastilysock, GodfriedToussaint, EmausBot, Dappe~enwiki, Blueshifting, Primefac, Starcheerspeaksnewslostwars, Slightsmile, Radshashi, Dcirovic, Fæ, Compvision, Oktie, Іванко1, Itss ieee, ExDxV, Tot12, Thacheezinator, ClueBot NG, Tillander, Aiwing, Kimberleyporter, NormBograham, Snotbot, MiroBrada, Atiqahad, WikiMSL, Paul.mather, BG19bot, Lisasolomonsalford, KimberlyWHoyt, Kpthai, Ligu, J.Davis314, Oritnk, Spearmanm, Me, Myself, and I are Here, Phamnhatkhanh, Rexgraham, Phleg1, Yes deeper, Monkbot, QueSera4710, YdJ, Annaelison, KasparBot, Datakeeper and Anonymous: 128

9.2 Images

- **File:800px-Cool_Kids_of_Death_Off_Festival_p_146-face_selected.jpg** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/d9/800px-Cool_Kids_of_Death_Off_Festival_p_146-face_selected.jpg *License:* CC BY-SA 3.0 *Contributors:*
- Cool_Kids_of_Death_Off_Festival_p_146.jpg *Original artist:*
- derivative work: MaGIc2laNTern (talk)
- **File:Ambox_important.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/b4/Ambox_important.svg *License:* Public domain *Contributors:* Own work, based off of Image:Ambox scales.svg *Original artist:* Dsmurat (talk · contribs)
- **File:Edit-clear.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/f/f2/Edit-clear.svg *License:* Public domain *Contributors:* The *Tango! Desktop Project*. *Original artist:*

  The people from the Tango! project. And according to the meta-data in the file, specifically: “Andreas Nilsson, and Jakub Steiner (although minimally).”
- **File:Gnome-searchtool.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1e/Gnome-searchtool.svg *License:* LGPL *Contributors:* http://ftp.gnome.org/pub/GNOME/sources/gnome-themes-extras/0.9/gnome-themes-extras-0.9.0.tar.gz *Original artist:* David Vignoni
- **File:Portal-puzzle.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/f/fd/Portal-puzzle.svg *License:* Public domain *Contributors:* ? *Original artist:* ?
- **File:Question_book-new.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/9/99/Question_book-new.svg *License:* Cc-by-sa-3.0 *Contributors:*

  Created from scratch in Adobe Illustrator. Based on Image:Question book.png created by User:Equazcion *Original artist:* Tkgd2007
- **File:SRT_Shape_Recognition_Technology_Principles.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/d/d0/SRT_Shape_Recognition_Technology_Principles.png *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* YdJ
- **File:Split-arrows.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/a/a7/Split-arrows.svg *License:* Public domain *Contributors:* ? *Original artist:* ?

9.3 Content license

- Creative Commons Attribution-Share Alike 3.0