Typically this is a search for the best solution guided by an optimization algorithm.

7.4.3 Low-level Methods

The simplest methods rely entirely on local image operators and a heuristic grouping of pixels with similar local photometric characteristics. Hence, these methods use only common knowledge of the photometry or geometry. They are therefore considered as *low-level* methods. They consist of the following steps. First, a local image operator may be applied, yielding a new image in which the local features are emphasized. Examples of local image operators are given in Section 1.3.4. Second, pixels with similar local photometric characteristics are grouped. Typical examples of low-level methods are region growing and edge detection.

- ***Region growing*** partitions an image into regions by grouping adjacent pixels with similar gray values, thus creating boundaries between contrasting regions (Figure 7.4). It is often initiated by indicating so-called seed points, which grow by iteratively merging adjacent pixels with similar gray values. Gray value similarity is assessed with simple measures that compare the gray values of neighboring regions. Two adjacent regions are merged if, for example, their mean values differ less than a specified value. In medical imaging, the assumptions on which region growing is based are usually violated. Object intensity is often not homogeneous because of noise and artifacts (see, e.g., Figure 4.45), and adjacent structures may not have sharp boundaries because of poor contrast or insufficient resolution, or because there is simply no clear biological boundary. When applied to medical images region growing then results in regions that are either too small or too large.

  An important positive exception is the segmentation of bony structures in CT images. Because bone is much denser than soft tissue, its CT values are significantly higher, and a simple threshold operation is usually sufficient to separate bone from its surrounding structures (see Figure 7.5). Thresholding is particularly useful for images with a bimodal histogram (see Figure 1.8). The threshold value partitions the image into two classes, which typically correspond to object pixels and background pixels. More than one threshold value can also be used to model one or more different classes, each corresponding to a distinctive interval in the histogram.
- ***Edge detection*** is basically similar to region growing, but instead of grouping pixels, boundary points are linked or tracked. Boundaries are found by first applying a local differential operator, such as the gradient or Laplacian (see Section 1.3.4.1), and subsequently linking those pixels that are most sensitive to this operator. In the ideal case of images with high contrast and without noise, the physical object boundary is found. Medical image data, however, are typically complex. Consequently, the output of a local differential operator does not always reflect the expected meaningful edges, causing an automatic edge linking procedure to get lost (Figure 7.6).

Despite the heuristic nature and the poor performance of these low-level approaches, they are popular in commercial image analysis tools. The reason is that these methods are fast, simple to understand and to implement, and they are generic as they do not assume specific knowledge about the objects to be analyzed. Furthermore, low-level methods are also used to segment the image into subparts or to combine pixels into patterns as input for more sophisticated, model-based methods.[35]

7.4.4 Model-based Methods

Effective image analysis methods must incorporate prior knowledge of the photometry, geometry, and/or context of the considered structures. The nature of these properties can be physical, statistical, and tissue-dependent as well. Such methods rely on a built-in conceptual model for the objects they are looking for. These model-based methods must be able to cope with complex image data. In the remainder of this chapter the basic model-based methods for medical images are discussed. The following two categories are distinguished.

7.4.4.1 Data Classification/Regression

In this data-driven approach, characteristic object features, such as mean gray value or color, area,

35 P. Suetens, P. Fua, and A. J. Hanson. Computational strategies for object recognition. *ACM Computing Surveys*, 24(1):5–61, 1992.

(a)

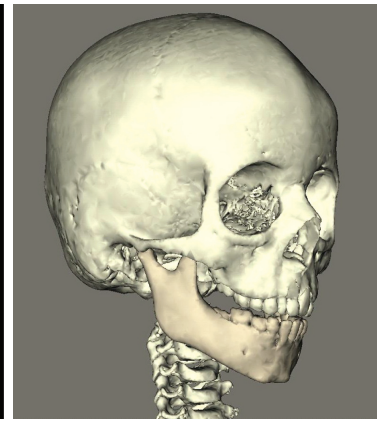
(b)

(c)

Figure 7.4 Segmentation of an empty colon in a CT image of the abdomen for virtual endoscopy. **(a)** Original CT image through the colon. (**b)** A 3D region growing procedure initiated from a seed point in the colon extracts all contiguous pixels with CT intensity similar to air while excluding nonconnected pixels with similar intensity such as the background. Segmented regions that are not connected on this slice are effectively connected in 3D. **(c)** 3D rendering of the segmented colon.

(a)

(b)



(c)

Figure 7.5 Segmentation of the skull and the mandibula in CT images using thresholding. **(a)** Original CT image of the head. **(b)** Result with a threshold value of 276 Hounsfield units. The segmented bony structures are represented in color. **(c)** 3D rendering of the skull shows a congenital growth deficiency of the mandibula in this eight-year-old patient. This information was used preoperatively to plan a repositioning of the mandibula. (Courtesy of Nobel Biocare.)

perimeter, compactness, and so forth, are calculated from the image data and typically represented as a feature vector. For many features it implies that the considered objects have been delineated in the image without any knowledge of the exact model appearance. Based on its descriptive features, an object is then assigned to the most appropriate class from a discrete set of classes (i.e., classification) and/or given the most related value from a continuous range of values (i.e., regression). The feature vector strategy is well established and has proved its usefulness in many industrial applications. It has been described

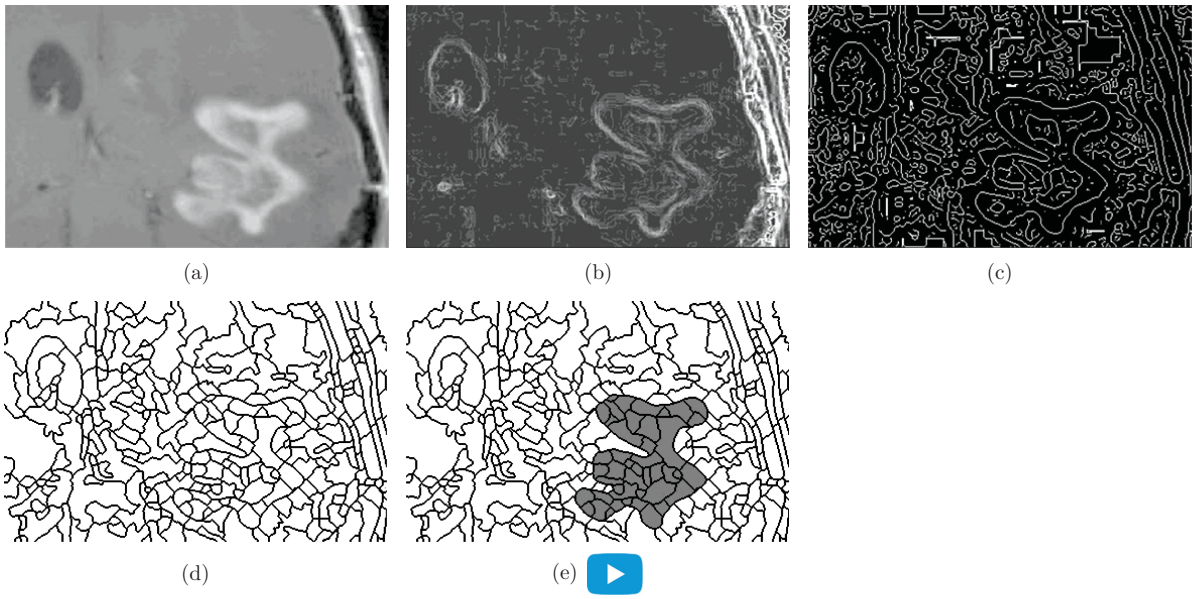


Figure 7.6 Delineation of a brain lesion in a CT image. **(a)** Original image. **(b)** Gradient magnitude image. **(c)** Result of the Canny edge detector, which tracks the local maxima of the gradient magnitude. Closed contours are not guaranteed. **(d)** Edges converted into closed contours by considering the gradient magnitude image as a topographic relief and computing watershed lines. This typically results in oversegmentation of the image into a large number of small regions. **(e)** By interactively merging adjacent regions with similar intensity, only relevant boundaries corresponding to prominent edges remain.

extensively in the literature since the early years of digital imaging.[36]

- ***Pixel labeling***
  In its simplest form this computational strategy assigns individual pixels to the most probable class based on their photometry, geometry, and context.
- ***Pattern recognition***
  More generally, this method does not start from pixels, but from image patches or patterns, i.e., groups of pixels, with photometric, geometric, and contextual features. Characteristic image patterns can be found by image segmentation but this is not an easy task unless the image data are simple. Interactive delineation may be helpful. Generating an oversupply of image patterns, classifying them all and retaining the best set, is an alternative. In the limit features can be investigated for all possible groups of pixels without the need for segmentation. Given the image patterns with their features, the goal then is to assign them to the most probable category from a discrete set (classification) and/or give them a value of a continuous dependent variable (regression).

7.4.4.2 Model Fitting

This model-driven approach generates model instances with varying geometry, photometry, and context and finds the most probable model instance that describes the image data. Typically, this is a search procedure that optimizes a measure expressing the similarity between the model instance and the image data while taking the prior probability of the model instance into account.

- ***Geometric model fitting using a transformation matrix***
  These strategies assume that the geometric variability of the model can be described by a general geometric transformation matrix (see Section 1.3.3), such as a translation, rotation, scaling, shear, affine transformation, and perspective projection. The shape can be represented explicitly as a graph, a curve, surface, or implicitly as an image pattern itself.
- ***Flexible geometric model fitting***
  In many cases, the model needs to be more flexible to take the variability in appearance into account. Flexible geometric models can be

36 R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973. J. T. Tou and R.C. Gonzales. *Pattern Recognition Principles*. Addison-Wesley, Reading, Massachusetts, 1974.

represented as constraints and penalties on the geometric properties of a deformable curve or on a deformable picture or image pattern. Examples of geometric properties are smoothness, curvature, rectilinearity, parallelism, symmetry, elasticity, and rigidity.

7.5 Data Classification/Regression

Data classification/regression assigns pixels or image patterns to a category and/or gives them a value. The pixels or patterns are described by their photometric, geometric, and contextual features, typically represented by a feature vector. Note that groups of pixels or image patterns arranged in a 2D array, graph, curve, or surface are image patterns as well and can also be represented by a linear array of features, making feature vector classification/regression a fairly generally applicable strategy. Image patterns can be obtained by image segmentation but for complex image data this process is often inaccurate. This problem can be overcome by generating an oversupply of candidate image patterns, classifying them all and choosing the best. This principle is implicitly used in Convolutional Neural Networks (Section 7.5.2.2), which investigates all possible image patterns starting from the raw image data.

A well-chosen set of features is discriminative, i.e., pixels or image patterns of the same type have similar feature vectors and contrast with the feature vectors of pixels of a different type. If the feature vectors are represented in a multidimensional feature space, the classification strategy then consists of partitioning the feature space into a number of *classes* (i.e., nonoverlapping regions that separate the different categories).

The boundaries between the regions in the feature space are constructed by means of a decision criterion that is based on prior knowledge. A variety of decision criteria exists to discriminate the classes in the feature space. Popular are methods that learn the *decision boundaries* from examples, such as K-nearest neighbors, neural networks, support vector machines, and random forests. They are extensively studied in the scientific discipline of *machine learning*[37] but details are beyond the scope of this textbook.

37 Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

7.5.1 Pixel Labeling

Pixel classification is a special case of feature vector classification. Pixels can be considered as the smallest possible image patterns and do not need to be outlined. Pixel features can simply be calculated or are directly available as a single value, such as the gray value, or, more generally, as a vector, such as the red-green-blue (RGB) values in color images or (ρ, T_1, T_2) in MRI. Partly because of this simplicity, pixel classification is very popular in medical image computing. Based on their feature vector, pixels can then be assigned, for example, to a vegetation type in aerial images or a particular tissue type in medical images.

Sometimes, in order to account for the partial volume effect, mixture classes are introduced whose intensities are a weighted sum of the intensities of the pure tissue classes. In other cases a rejection class that collects all pixels that cannot be classified into one of the modeled classes can be included to cope, for example, with pathological areas (see Figure 7.45 below).

Generally, the problem can be stated as finding $\arg\max_\Phi p(\Phi|I)$ with $\Phi = \{\phi_k; k = 1, \cdots, N\}$ the tissue labels and $I = \{I_k; k = 1, \cdots, N\}$ the intensities or other features of the pixels $k = 1, \cdots, N$. For a proper application of the classification strategy the tissue labels of neighboring pixels are assumed to be independent. $p(\Phi|I)$ can then be written as

$$p(\Phi|I) = \prod_k p(\phi_k \mid I_k). \tag{7.1}$$

and the problem reduces to finding the maximum of $p(\phi_k \mid I_k)$ for each pixel k independently, i.e., find $\arg\max_{\phi_k} p(\phi_k \mid I_k)$. Hence, each pixel with intensity or feature vector I_k is assigned to the class with the highest probability $p(\phi_k|I_k)$, with ϕ_k the class or tissue label of pixel k. The shape of the decision boundaries in feature space, defined by the transitions where one class becomes more likely than another, can be learned.

7.5.1.1 Supervised Learning

Using supervised learning the probabilities $p(\phi_k|I_k)$ can be learned from a representative set of pixel samples for which the class they belong to is known. This can, for example, be done by manually outlining several regions of pixels, each corresponding to a different class. This process is known as *supervised learning*. After this training phase, the unclassified pixels are assigned to the most probable class ϕ_k,

i.e., the one with the maximal posterior probability $p(\phi_k \mid I_k)$. This way the image is segmented into the different classes. It is also possible to represent the posterior probabilities $p(\phi_k \mid I_k)$ as gray values in each pixel, which yields a so-called *fuzzy segmentation*.

Instead of learning the posterior probability distribution $p(\phi_k|I_k)$ directly, it may also be learned indirectly from the likelihood $p(I_k \mid \phi_k)$ and the prior probability $p(\phi_k)$ using the Bayes' rule

$$p(\phi_k \mid I_k) = \frac{p(I_k \mid \phi_k)\, p(\phi_k)}{p(I_k)}. \tag{7.2}$$

When maximizing $p(\phi_k|I_k)$, the probability $p(I_k)$ is constant and can be ignored. Hence,

$$\arg\max_{\phi_k} p(\phi_k|I_k) = \arg\max_{\phi_k}(p(I_k \mid \phi_k) \cdot p(\phi_k)). \tag{7.3}$$

In this case the prior $p(\phi_k)$ is assumed to be known and can be different for each pixel. If no prior information is available, all classes are equally likely and maximizing the posterior probability $p(\phi_k|I_k)$ does not differ from maximizing the likelihood $p(I_k \mid \phi_k)$, i.e.,

$$\arg\max_{\phi_k} p(\phi_k|I_k) = \arg\max_{\phi_k} p(I_k \mid \phi_k). \tag{7.4}$$

Often, the intensity variations within a given tissue class ϕ_k (e.g., white matter, gray matter, cerebrospinal fluid, and so forth in the case of brain tissue) can be assumed to have a Gaussian distribution, i.e.,

$$p(I_k \mid \phi_k; \mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi}\sigma_k} \cdot \exp\Big(-\frac{(I_k - \mu_k)^2}{2\sigma_k^2}\Big), \tag{7.5}$$

with μ_k and σ_k the unknown mean and standard deviation of class ϕ_k. Because an exponential function is monotonically decreasing

$$\arg\max_{\phi_k} p(I_k|\phi_k) = \arg\min_{\phi_k} \frac{(I_k - \mu_k)^2}{\sigma_k^2}. \tag{7.6}$$

Hence, each pixel k is assigned to the tissue class for which $\frac{(I_k-\mu_k)^2}{\sigma_k^2}$ is minimal. Note that a uniform prior distribution $p(\phi_k)$ is assumed here. Instead, prior knowledge about the spatial distribution of the various tissue classes in the image can be derived from a statistical atlas and used to solve Eq. 7.3 (see Figure 7.7).

If multispectral MR data or multimodal image data are available, the model for each class ϕ_k can be extended to a multivariate distribution with vector mean μ_k and covariance matrix S_k. For the n-dimensional case the likelihood $p(I_k|\phi_k)$ then becomes

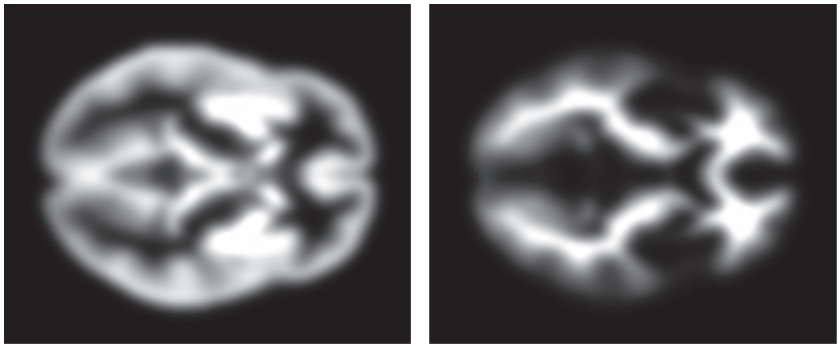
$$p(I_k|\phi_k) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|S_k|}} \cdot \exp\Big(-\frac{1}{2}(\mathbf{I}_k - \mu_k)^{\mathrm{T}} S_k{}^{-1}(\mathbf{I}_k - \mu_k)\Big) \tag{7.7}$$

and the most likely tissue labels

$$\arg\max_{\phi_k} p(I_k \mid \phi_k) = \arg\min_{\phi_k} (\mathbf{I}_k - \mu_k)^{\mathrm{T}} S_k{}^{-1}(\mathbf{I}_k - \mu_k). \tag{7.8}$$

7.5.1.2 Unsupervised Learning

The training phase in supervised learning typically requires user interaction, which may be too cumbersome in clinical practice. For a fully automated procedure, the values of the mean μ_k and standard deviation σ_k of a Gaussian distribution can, for example, be considered as unknown model parameters in the optimization process. This is called *unsupervised learning*. Let $\Theta = \{\mu_j, \sigma_j\}$ be the unknown mean and standard deviation of all the tissue classes. The goal is



(a) (b)

Figure 7.7 Statistical images of **(a)** the gray brain matter and **(b)** the white brain matter. The intensity in each pixel is proportional to its prior probability $p(\phi_k = c_j)$ of belonging to that particular tissue class.

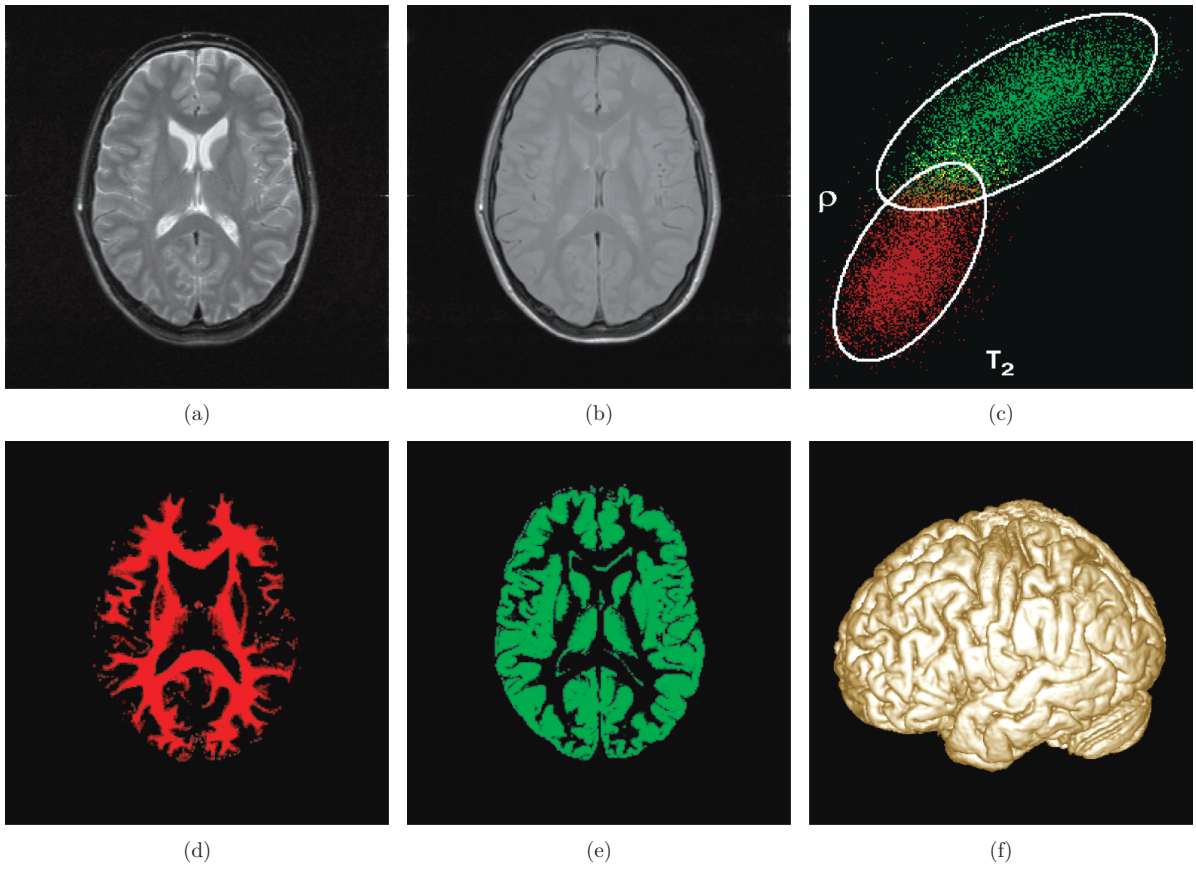


Figure 7.8 Brain tissue segmentation in multispectral MR images using unsupervised pixel classification. (**a** and **b**) Original T_2- and ρ-weighted MR images. **(c** and **d)** Classification of white and gray matter represented in red and green, respectively. **(e)** Scatter plot of the pixels in **(c)** and **(d)** as a function of ρ and T_2, colored according to their tissue type, together with the 0.99 percentile contours of the Gaussian class intensity model that was fitted using the EM algorithm. **(f)** 3D representation of the cortex obtained by volume rendering of the gray matter segmentation **(d)**.

then to find $\{\Phi, \Theta\}$ with the highest probability given the data I, i.e., find $\arg\max_{\Phi,\Theta} p(\Phi, \Theta|I)$. This can be solved iteratively with the expectation-maximization (EM) algorithm. This method iteratively generates hypotheses for the tissue classes Φ and their parameters Θ and matches them against the data I. Details of this optimization algorithm are beyond the scope of this textbook. Figure 7.8 shows an example.

Similarly, the model can be any other probability distribution or function with unknown parameters that must be defined during the pixel labeling. For example, in fMRI, the intensity variation in a time series of fMRI images, acquired during brain stimulation, is modeled as a linear combination of time-dependent functions that represent the stimulation course in the experiment and the low-frequency signal drift over time (Figure 7.9(a)). Voxels with a good match with the stimulation pattern are classified as functional areas that respond to the stimulus (Figure 7.9(b)). Similarly, in perfusion studies, a time-dependent model for the contrast or tracer accumulation in tissue is fitted to the observed intensity changes.

7.5.1.3 Spatial Dependency

Tissue labels of neighboring pixels are not necessarily independent. For example, neighboring pixels can be expected to belong to the same class. Employing this knowledge yields a smooth label image. Similarly, local geometric knowledge can be incorporated in pixel classification. Consider, for example, the problem of blood vessel classification, in which for each pixel the probability that it belongs to a vessel needs to be calculated. Blood vessels are smooth tubular structures. If the labels of neighboring pixels were independent it would be sufficient to apply a local

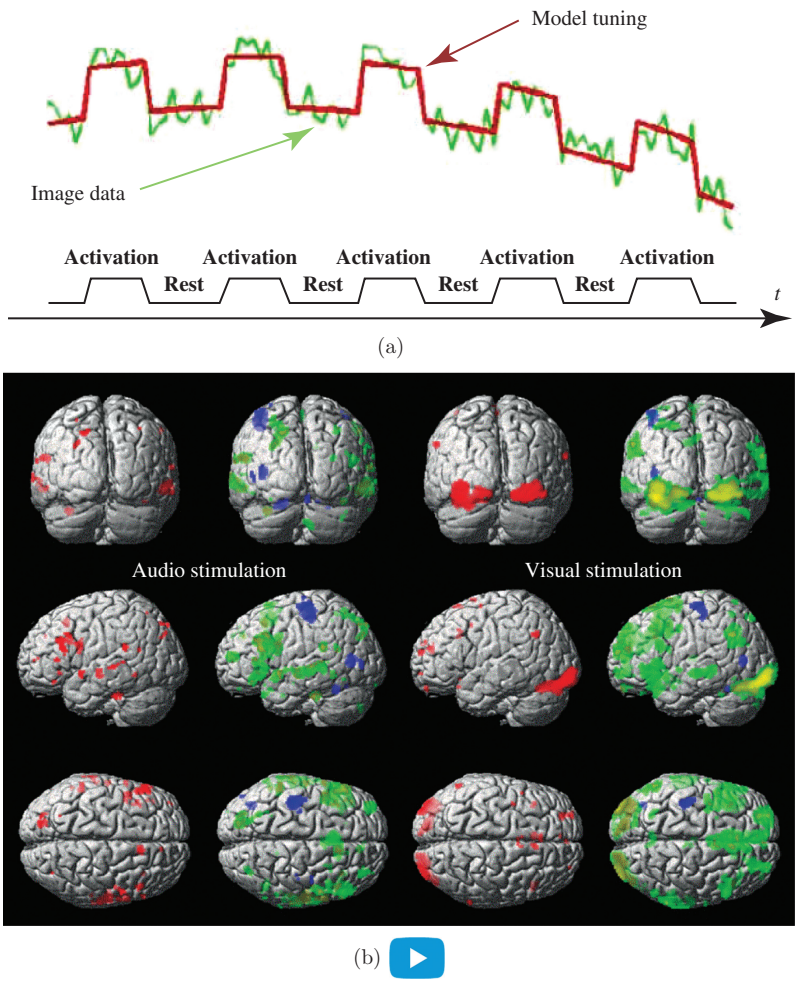


Figure 7.9 Pixel labeling in fMRI. **(a)** The fMRI signal in each voxel (green noisy line) is modeled as a linear combination of functions (red line) that reflect the activation stimuli ("activation-rest" step signal) and the low-frequency signal drift. Clusters of voxels with a significant response to the applied stimulus are classified as brain activation areas. **(b)** Activation areas. Columns 1 and 3: hearing and seeing words (red) during respectively auditory and visual stimulation. Columns 2 and 4: subsequent semantic decision (green) and right-hand response (blue). The yellow color is a mixture of red (perception) and green (interpretation). (Courtesy of Professor S. Sunaert, Department of Radiology.)

image operator sensitive to bar-like primitives and use it as an additional feature in the feature vector. However, tubular structures of neighboring pixels have a similar orientation. This requirement should also be taken into account.

Because the classification of a pixel depends on the labels of its neighboring pixels, iterative methods are typically needed to find the best solution. This way new model instances are iteratively generated and matched with the data. Note that this strategy of generating and matching model instances is to be considered as an example of model fitting, which is explained below (see Section 7.6).

7.5.2 Pattern Recognition

Depending on the geometry of the image patterns, plenty of descriptive features at multiple scales can be generated. Good properties are characteristic for the object they describe and must be able to distinguish different categories and provide compact distinctive clusters in the feature space. Since the emergence of digital image processing in the 1960s research toward better features has always been a focus. For a long time, image features have been hand-crafted. However, this has changed since the emergence of Convolutional Neural Networks (Section 7.5.2.2 below).

7.5.2.1 Dimensionality Reduction

The number of patterns that can be generated in an image is immens. In image analysis tasks it is often not obvious to select an appropriate small set of characteristic and distinctive features. Some of the chosen features may be redundant, interdependent, or insignificant, and too many features increases the computational cost as well as the sparsity of the data

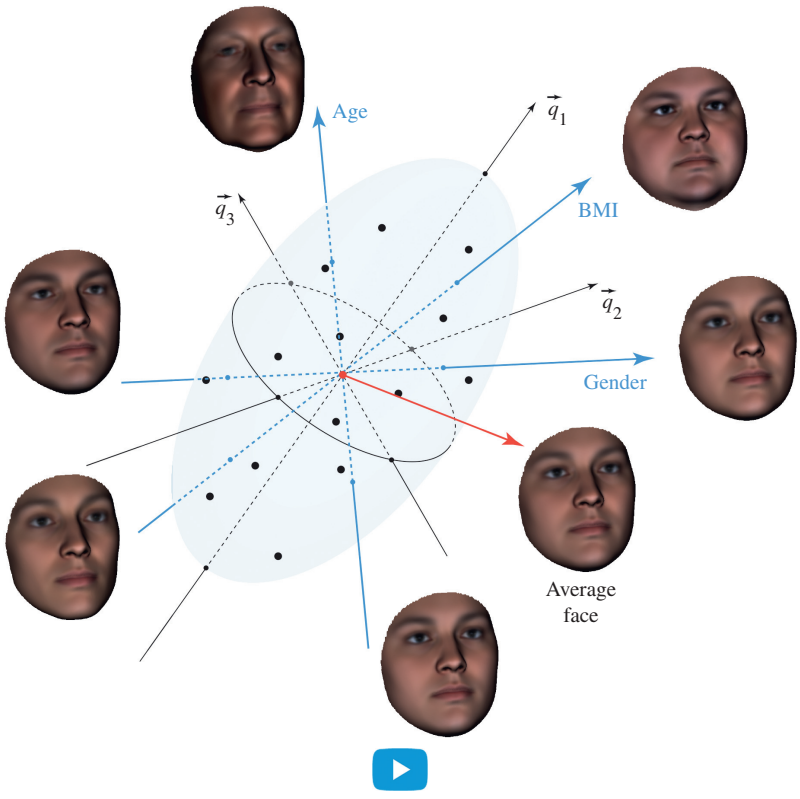


Figure 7.10 Schematic representation of the uncorrelated feature space after dimensionality reduction by PCA. The contextual features, such as age, BMI, and gender, remain linear axes in this new feature space. The faces shown at both ends are extremes within the normal range.

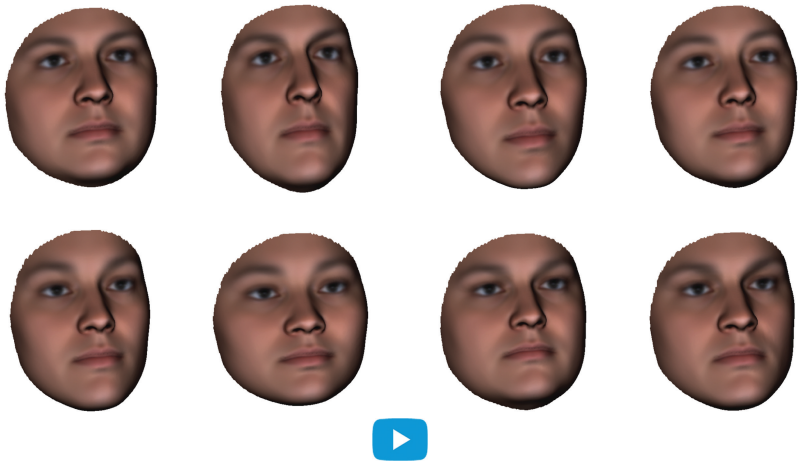


Figure 7.11 Each column from left to right shows two extreme learned faces described by respectively the first four eigenvectors, i.e., $\mathbf{v} = \bar{\mathbf{v}} \pm 3\sqrt{\lambda_k}\mathbf{q}_k$, $k = 1, \cdots, 4$ with $\sqrt{\lambda_k}$ the standard deviation along $\mathbf{q}_k$.

in the feature space. Sparse data give the classifier a higher degree of freedom, which may cause overfitting if the classifier fits the training data too closely and does not generalize well to new data. Overfitting can be avoided by increasing the amount of training data, by modeling a classifier with less degrees of freedom, or by dimensionality reduction. The dimension of the feature space can be reduced by selecting a subset of principal features or by transforming the feature space to a lower dimensional space and projecting the data into that space.

Here is an example. A high-dimensional feature space is shown in Figures 7.10–7.11. Given a database of 3D textured face surfaces, a model is built by representing each complete face as a high-dimensional linear array of landmarks through their 3D surface

coordinates and corresponding skin color. Contextual properties, such as age, gender, bmi, ancestry, etc., are added as additional features to this list. Note that this global face representation implies that no prior segmentation is needed. It is, however, assumed that corresponding face landmarks in different faces have the same vector index. Pose differences of the 3D faces are eliminated by spatially aligning the surfaces (for example, by minimizing the sum of squared differences). Dimensionality reduction is performed using principal component analysis (PCA), yielding a reduction from tens of thousands of features to a few tens of new uncorrelated features, represented by the eigenvectors, also known as eigenfaces.

PCA and ICA. Given a training set of m parametric shapes $\{\mathbf{v}_i, i = 1, 2, \cdots, m\}$. In two dimensions, for example, each shape is written as a column vector of coordinates (x_{ik}, y_{ik}), that is, $\mathbf{v}_i = [x_{i1}\ y_{i1}\ x_{i2}\ y_{i2} \cdots x_{in}\ y_{in}]^{\mathrm{T}}$. The shape variations $\mathbf{v}_i - \bar{\mathbf{v}}$, where $\bar{\mathbf{v}}$ is the mean shape defined as

$$\bar{\mathbf{v}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{v}_i, \tag{7.9}$$

can be represented in a $2n$-dimensional feature space whose axes correspond to the n points along the contour. The variations on different features are not necessarily uncorrelated. To work in an uncorrelated feature space, the theory of *principal component analysis* (PCA) can be applied as follows. The shape variability in the training set is represented by the $2n \times 2n$ covariance matrix S of shape distortions $\mathbf{v}_i - \bar{\mathbf{v}}$ of all the shapes in the set

$$S = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{v}_i - \bar{\mathbf{v}}) \cdot (\mathbf{v}_i - \bar{\mathbf{v}})^{\mathrm{T}}. \tag{7.10}$$

This matrix can also be written as

$$S = Q \cdot \Lambda \cdot Q^{\mathrm{T}}, \tag{7.11}$$

where $Q = [\mathbf{q}_1\ \mathbf{q}_2 \cdots \mathbf{q}_{2n}]$ is the $2n \times 2n$ unitary matrix of eigenvectors $\mathbf{q}_k$ of S, and Λ is the diagonal matrix of corresponding eigenvalues λ_k (with $\lambda_1 \geq \lambda_2 \geq \cdots$). The new axes $\mathbf{q}_k$ in feature space correspond to the new modes of variation, which are mutually uncorrelated and are characteristic for the shape diversity in the training set. $\sqrt{\lambda_k}$ is the standard deviation along $\mathbf{q}_k$ of all the shapes in the learning set. The shape model can then be written as

$$\mathbf{v} = \bar{\mathbf{v}} + \sum_{k}^{2n} c_k \cdot \mathbf{q}_k \tag{7.12}$$

or

$$\mathbf{v} = \bar{\mathbf{v}} + Q \cdot \mathbf{c} \tag{7.13}$$

with

$$\mathbf{c} = [c_1\ \ c_2 \cdots c_{2n}]^{\mathrm{T}} \tag{7.14}$$

and

$$Q = [\mathbf{q}_1\ \ \mathbf{q}_2 \cdots \mathbf{q}_{2n}]. \tag{7.15}$$

Because $Q^{-1} = Q^T$, $\mathbf{c}$ can be calculated from $\mathbf{v}$ as

$$\mathbf{c} = Q^T \cdot (\mathbf{v} - \bar{\mathbf{v}}) \tag{7.16}$$

or

$$c_k = \mathbf{q}_k^T \cdot (\mathbf{v} - \bar{\mathbf{v}}). \tag{7.17}$$

Each eigenvector $\mathbf{q}_k$ has a corresponding eigenvalue λ_k, which is the variance of parameter c_k in the set of training shapes. Because $\lambda_1 \geq \lambda_2 \geq \cdots$ holds, the foremost modes of variation explain most of the variability in the training set. By constraining the model to include only the ν most important modes of variation that explain most of the variability in the training set, the number of degrees of freedom of the model can be significantly reduced without affecting much of its descriptive power.

Any given shape instance $\mathbf{v}^*$ can then be written as the sum of the average shape and a linear mixture of uncorrelated *eigenshapes*, that is,

$$\mathbf{v}^* \approx \bar{\mathbf{v}} + \sum_{k}^{\nu} c_k^* \cdot \mathbf{q}_k \tag{7.18}$$

or

$$\mathbf{v}^* \approx \bar{\mathbf{v}} + Q \cdot \mathbf{c}^* \tag{7.19}$$

with

$$\mathbf{c}^* = [c_1^*\ \ c_2^* \cdots c_\nu^*]^{\mathrm{T}} \tag{7.20}$$

and

$$Q = [\mathbf{q}_1\ \ \mathbf{q}_2 \cdots \mathbf{q}_\nu]. \tag{7.21}$$

The coefficients c_k^* can be calculated from $\mathbf{v}^*$ as

$$c_k^* = \mathbf{q}_k^T \cdot (\mathbf{v}^* - \bar{\mathbf{v}}). \tag{7.22}$$

Obviously, this theory can easily be extended from two-dimensional curves to three-dimensional surfaces or any other set of parametric functions.

For example, spatially aligned images, represented by their gray values, can be written as

$$I = \bar{I} + \sum_{k}^{2n} c_k \cdot I_k, \tag{7.23}$$

where $\bar{I}$ is the average intensity image of the training set and I_k the eigenvectors, i.e., the principal modes of variation, also known also *eigenimages* or *eigenfaces*.

Note that features that are uncorrelated are not necessarily independent. PCA removes only linear dependencies, i.e., *second order* correlations. This is appropriate if the data, i.e., the shapes in the feature space, have a Gaussian distribution. To remove higher-order correlations, statistically independent features can be required, which can be obtained by minimizing the multivariate mutual information. More about the concept of mutual information can be found in Section 7.6.1.1.

Faces are points in the feature space and can be written as

$$\mathbf{v} \approx \bar{\mathbf{v}} + \sum_{k}^{v} c_k \cdot \mathbf{q}_k \tag{7.24}$$

with $\mathbf{q}_k$ the eigenvectors and λ_k the corresponding eigenvalues ($\lambda_1 \geq \lambda_2 \geq \cdots$). $\sqrt{\lambda_k}$ is the standard deviation along $\mathbf{q}_k$ of all the shapes in the learning set (Figure 7.10). Figure 7.11 shows some extreme faces described by, respectively, the first four eigenvectors, i.e., $\mathbf{v} = \bar{\mathbf{v}} \pm 3\sqrt{\lambda_k}\mathbf{q}_k$ ($k = 1, \cdots, 4$).

For any given face $\mathbf{v}^*$ the coefficients c_k^* can be calculated as

$$c_k^* = \mathbf{q}_k^T \cdot (\mathbf{v}^* - \bar{\mathbf{v}}). \tag{7.25}$$

Note that the contextual features, such as age, BMI, and gender, remain linear axes in the new uncorrelated feature space (Figure 7.10).

Pattern classification. Figure 7.12 shows an example of the effect of two extreme SNP variations in one particular gene (SLC35D1). SNPs are variations in a single nucleotide at a specific location in the genome. Mutations in this gene are known to cause Schneckenbecken dysplasia, with superiorly oriented orbits as a typical feature. This facial characteristic can also be noticed in a normal population by looking at the difference between faces with varying SNP in this particular gene. Significant local differences may yield characteristic *biomarkers* for this genetic disorder.

Predicting missing data. Figure 7.13 (top) shows a patient with hemifacial hypertrophy subject to maxillofacial surgery. Let us assume that the affected part of the face was delineated and deleted from the surface description. The most likely normal face that fits the unaffected part of the patient's face while completing the resected part can be calculated using a face model. It is called the normal-equivalent face (Figure 7.13 (middle)) and found as follows. Given the face model and the incomplete data $\mathbf{v}^*$, that is, the unaffected portion of the face, the goal is to find the best model instance $\mathbf{c}^*$, taking into account how likely this model instance a priori is, i.e., $p(\mathbf{c}^*)$, and how similar the model instance and the data are, i.e., $p(\mathbf{v}^*|\mathbf{c}^*)$.

If the parameters c_k have a normal distribution, the model instance $\mathbf{c}^*$ that maximizes $p(\mathbf{c})$ is the one that maximizes

$$\prod_{k=1}^{v} p(c_k) = \frac{1}{\prod_{k=1}^{v}\sqrt{2\pi\lambda_k}} \cdot \exp\left(-\frac{1}{2}\sum_{k=1}^{v}\frac{c_k^2}{\lambda_k}\right). \tag{7.26}$$

Because the exponential function is monotonically decreasing as a function of c_k^2, maximizing $\prod p(c_k)$ yields the same result as minimizing $\sum \frac{{c_k}^2}{\lambda_k}$:

$$\arg\max_{c_k} \prod_{k=1}^{v} p(c_k) = \arg\min_{c_k} \sum_{k=1}^{v} \frac{{c_k}^2}{\lambda_k}. \tag{7.27}$$

In the absence of constraints, the solution of this equation would yield the average face, i.e., $c_k = 0$ for all k. However, in this example, the model instance should maximally resemble the unaffected part $\mathbf{v}^*$ of the patient's face, expressed by the likelihood $p(\mathbf{v}^*|\mathbf{c})$. The normal-equivalent face can be calculated by maximizing the posterior probability, i.e., $\arg\max_{\mathbf{c}}(p(\mathbf{v}^*|\mathbf{c}) \cdot p(\mathbf{c}))$, which can be performed efficiently by iteratively generating model instances $\mathbf{c}^*$ and finding the most probable normal face that fits the unaffected part $\mathbf{v}^*$. Instead of pattern classification, this strategy, used in case of missing data, is to be considered as model fitting, which is discussed in the next section (Section 7.6). The distance map between the patient's face and the calculated normal-equivalent is shown in Figure 7.13 (bottom).

The idea of predicting missing facial parts can be extended to the complete face when only contextual data is available. Besides age, gender, ancestry, and BMI, these data can, for example, be genes responsible for facial development. Figure 7.19 shows the result of face reconstruction from DNA based on a

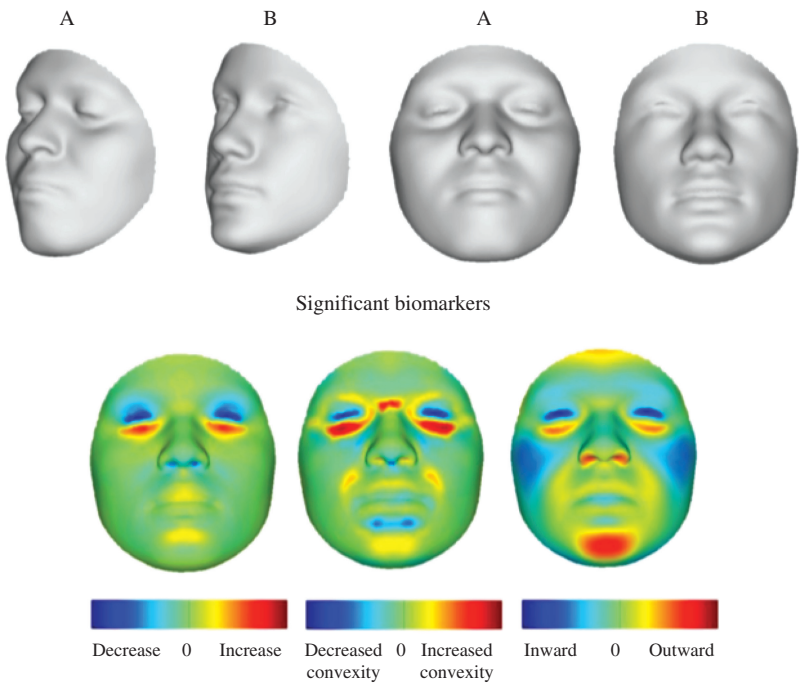


Figure 7.12 Result of a study of a selected set of SNP genotypes in a normal population. Faces A and B show the effect of two extreme SNP variants in gene SLC35D1. Mutations in this gene cause Schneckenbecken dysplasia. The color images show the differences between faces A and B of some local features (from left to right: strain, curvature change, and distance). Significant local differences, such as at the orbits, may define charactistic *biomarkers* for this particular genetic disorder.

database of 3D photographs of an admixed population and gene variants known to be responsible for facial development.

7.5.2.2 Convolutional Neural Networks

The systematic increase of computer power and available training data have triggered methods for automatically learning suitable features from a training set of raw images. Particularly a *Convolutional Neural Network* (CNN or ConvNet) has proven to be a powerful representation to solve classification/regression tasks in computer vision. In addition, CNNs are suited to be efficiently implemented on graphics processing units (GPUs), yielding computationally intensive but highly parallel algorithms. CNNs, to some extent inspired by the functioning of the visual system, exist since several decades. For medical image analysis, early applications include the detection of lung nodules in 2D chest X-rays [38]. In 2012 the adoption of CNNs got a tremendous boost when Krizhevsky *et al.*[39] showed their superior performance on a classification task with 1000 categories and a training set of 1.2 million images. Since then CNNs have become increasingly popular and they are able to surpass human performance in many narrowly defined tasks.

A CNN is a specific type of *Artificial Neural Network* (ANN) and, as such, consists of layers of artificial neurons or nodes (Figure 7.14). In a simple ANN every node in the network assigns weights to its incoming signals and adds them together. This weighted sum then activates the artificial neuron through a nonlinear* function. The output of a neuron serves as input for nodes in the next layer. The weights in the network are iteratively updated via gradient descent to optimize the network for the available training data.

38 S.-C. B. Lo, Y.-S. J. Lin, M. T. Freedman and S. K. Mun. Computer-assisted diagnosis of lung nodule detection using artificial convolution neural network, *SPIE Medical Imaging*, 1898: 859–869, 1993.
S.-C. B. Lo, S.-L. A. Lou Y.-S. J. Lin, M. T. Freedman, M. V. Chien and S. K. Mun. Artificial convolutional neural network techniques and applications for lung nodule detection, *Medical Imaging, IEEE Transactions on*, 14(4):711–718, December 1995.

* Nonlinear activation functions are able to construct decision boundaries (see p. 192) with flexible shapes, as against linear activation functions, whose decision boundaries are linear.

39 A. Krizhevsky, I. Sutskever and G. Hinton. Imagenet classification with deep convolutional neural networks, *Neural Information Processing Systems*, 197–1105, 2012.

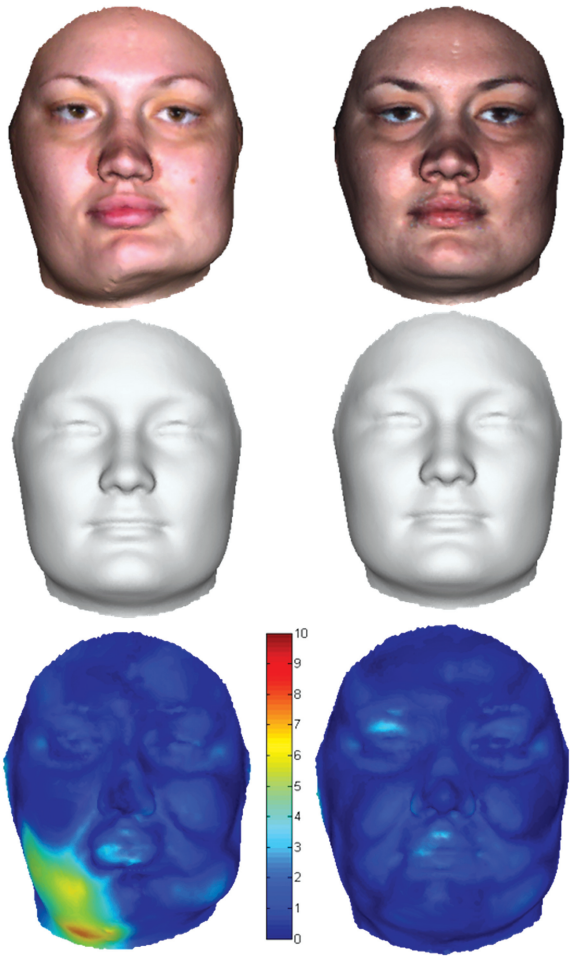


Figure 7.13 Left - before surgery, right - after surgery. Top to bottom: patient with hemifacial hypertrophy; the normal-equivalent face, i.e., the most likely normal face that fits the unaffected part of the patient's face; distance map between the patient's real face and the patient's calculated normal-equivalent face.

An ANN with multiple so-called *hidden layers* layers between the input and output layer can learn increasingly complex functions, however, at the expense of an increasing number of weights with issues of overfitting (see p. 196), and high computation time and memory requirements. Several methods exist to cope with these problems. In Section 7.5.2.1 dimensionality reduction was introduced to avoid overfitting. For neural networks, specific strategies have been developed, such as autoencoders, pooling, regularization, and data augmentation. A detailed discussion is beyond the scope of this textbook.

Networks with multiple hidden layers are often called *deep neural networks* and training them is referred to as *deep learning*.

In a CNN the final layers behave like a traditional ANN that performs a classification or regression task (see p. 192). Typically a CNN has several preceding *convolution layers*, which generate image features by means of discrete convolution operations. In this context, convolution is identical to cross-correlation, see Chapter 1 Section 1.3.4.1. The principle is shown in Figures 7.15–7.16.

A node in a convolution layer is comparable to a node in an ANN, i.e., it performs a nonlinear mapping of a weighted sum of the input signals. But specific is that the weights w_i are the parameters of a convolution mask (filter, kernel, see p. 8), and that a convolution layer consists of one or more images or *feature maps*, which ensure that the spatial relationships are retained. The convolution is applied to one of the images of the preceding layer (Figure 7.15(a)). If the preceding layer consists of k images, a *filter bank* of k convolutions is applied and the convolved images are added together. The result can be written as $\sum_{i=1}^{n} w_i.x_i$, with n the number of weights in one convolution mask (3×3 in this picture) times k, i.e., the number of images in the preceding layer (three in Figure 7.15(b)).

The input layer of a CNN is the original image itself, which can also consist of multiple components, such as red, green, and blue (RGB) in a color image (Figure 7.16). The image components are convolved, each with a different template, and added together. These values pass through the activation function of the nodes in the first hidden *convolution layer* where the results are stored in a feature map (Figure 7.16). Multiple feature maps per layer are typically established by means of different filter banks (Figure 7.16).

Next, the feature maps of the first convolution layer are convolved with new convolution masks to yield new feature maps in the second hidden convolution layer. Each time this process is repeated, a new hidden convolution layer is added to the network. The number of parameters per feature map equals the number of elements of a convolution mask times the number of feature maps k in the preceding layer.

While the nodes in the initial layers are excited by very local features, the nodes in subsequent layers are triggered by more complex and increasingly meaningful features. The CNN by Krizhevsky *et al.*[40]

40 A. Krizhevsky, I. Sutskever and G. Hinton. Imagenet classification with deep convolutional neural networks, *Neural Information Processing Systems*, 197–1105, 2012.

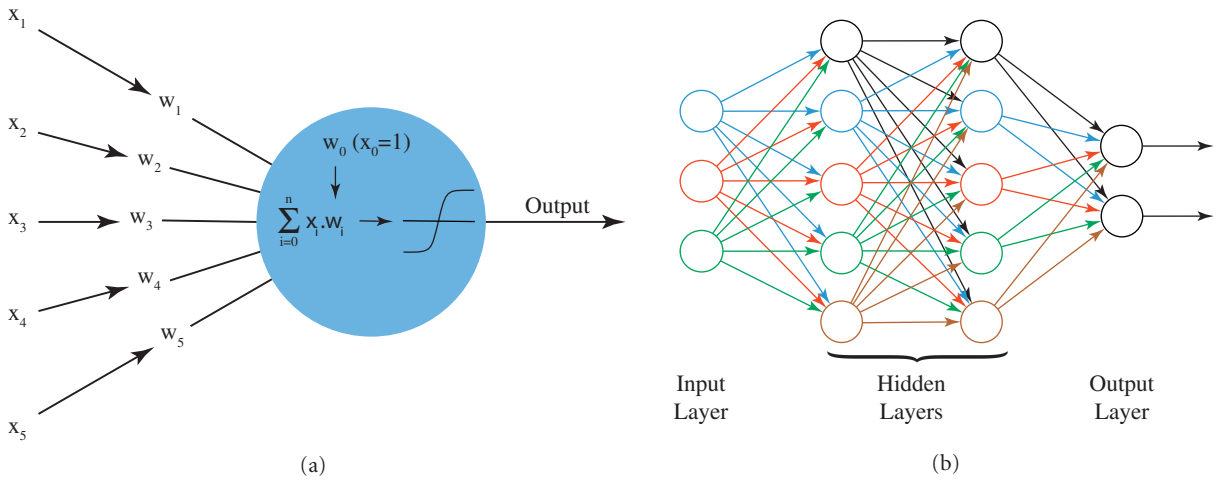


Figure 7.14 **(a)** Schematic representation of an artificial neuron or node, and **(b)** an artificial neural network (ANN). Every node in the network assigns weights w_i to its incoming signals x_i and adds them together, i.e., $\sum_{i=0}^{n} w_i.x_i$. This weighted sum is transformed into an output signal by a nonlinear activation function.

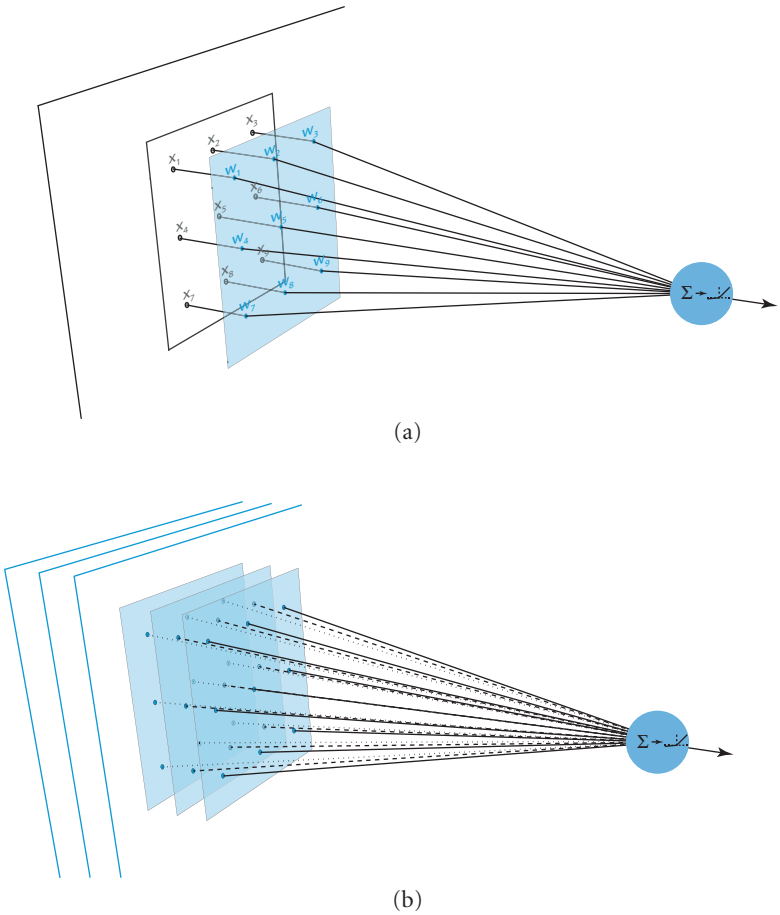


Figure 7.15 Schematic representation of an artificial neuron in a convolution layer of a CNN. **(a)** Compare to Figure 7.14. The weights w_i are the parameters of the convolution mask, which are multiplied with the values x_i of the corresponding pixels in one of the images of the preceding layer. **(b)** If the preceding layer contains k images (three in this picture), k convolution masks are applied and the resulting k values are added together. Together the k convolution masks form a filter bank. The result is $\sum_{i=1}^{n} w_i.x_i$, with n the number of weights in a convolution mask (3×3 in this picture) times k.

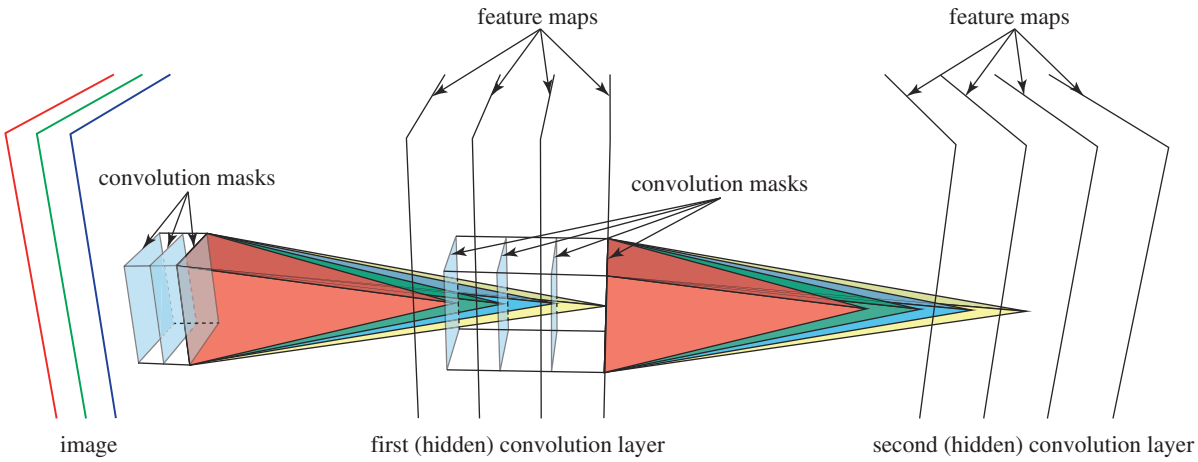


Figure 7.16 Schematic representation of a concatenation of image convolutions in a CNN. Each convolution layer consists of multiple feature maps. A feature map is an image of node outputs obtained from convolving the input values x_i of the preceding layer with the weights w_i as shown in Figure 7.15. Different features are extracted by different filter banks, but all the nodes in one feature map share the same weights w_i. Because the operations are image convolutions (cross-correlations, see Chapter 1 Section 1.3.4.1), the spatial geometry is maintained from the input image through all convolution layers in the network. Each individual node corresponds to a particular characteristic feature and is activated by similar image patches. At lower levels the triggering features are local, while at higher levels they are more global and increasingly meaningful. Note that between subsequent convolution layers other layers are often inserted to perform specific tasks such as pooling and upsampling. However, a detailed discussion about CNN architectures is beyond the scope of this textbook.

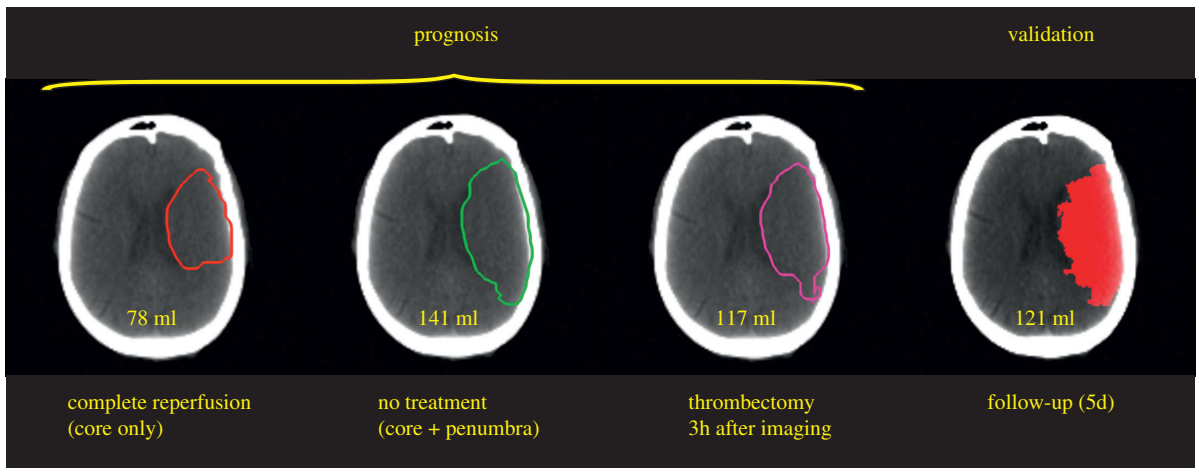


Figure 7.17 Outcome prediction of thrombectomy in acute stroke based on CT perfusion images. The left three images show the prognosis, and the right image shows the lesion in a the follow-up scan five days after the intervention. Three cases are predicted, (left) a complete reperfusion, which corresponds to the core, (middle) the final lesion without treatment, which corresponds to the initial core and penumbra, and (right) the lesion if the thrombectomy takes place three hours after imaging with a presumed and validated mTICI grade 2a (i.e., an antegrade reperfusion of less than half of the occluded target artery). Note the good match with the follow-up scan.

consisted of five convolution layers and about 500,000 nodes. Nowadays, CNNs with dozens of layers and tens of millions of parameters are no exception. Theoretically, an infinite number of different features at many different scales can be assembled and feature engineering is no longer needed, which is an important reason why CNNs have become powerful and attractive. However, increasing the number of layers and nodes is no guarantee for better performance. Too many parameters increase the computation cost and cause overfitting.

CNNs are often seen as a black box. However, in 2014 Zeiler and Fergus[41] contributed to understanding CNNs by visualizing those features that strongly

41 M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks, *European Conference on Computer Vision (ECCV)*, 818-833, 2014.

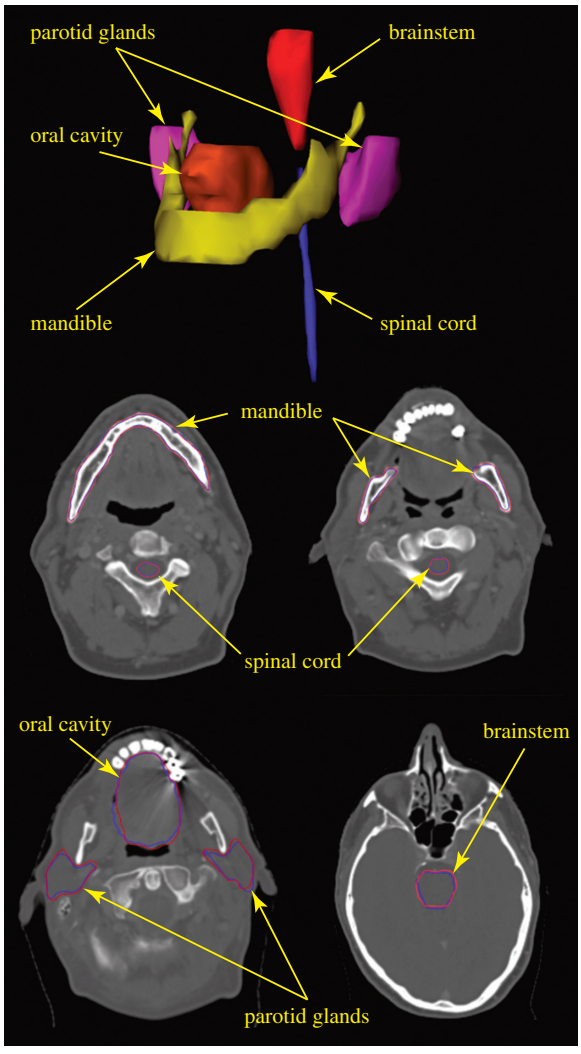


Figure 7.18 Example of automatic delineation of the organs at risk for radiotherapy planning in the head and neck area (blue contour by CNN, red by expert). Sixteen structures were delineated, i.e., the brainstem, left and right cochlea, upper esophagus, glottic larynx, mandible, oral cavity, supraglottic larynx, left and right parotid gland, inferior, mid and superior pharyngeal constrictor muscles (PCM), left and right submandibular gland, and spinal cord. Five of them are visualized here.

activate particular nodes at subsequent levels. At the first convolution level the features are very local, while at higher levels they become more global and meaningful.

Figure 7.17 shows an example of a CNN that predicts the outcome of intra-arterial thrombectomy in acute stroke based on CT perfusion images. The network was trained on the following data: 180 CT perfusion images in the acute phase; whether or not followed by an endovascular treatment, which was the case for about 50% of the training set; the time between imaging and the end of the thrombectomy; whether or not an occlusion was present; and a follow-up CT scan after five days with a delineation of the final lesion by an expert.

Figure 7.18 is an illustration of automatic delineation of the organs at risk for planning proton therapy in the head and neck area. The network was trained with 70 images, Five out of the sixteen delineated structures are visualized.

For a survey about deep learning for medical image analysis the reader is referred to the work of Litjens *et al.*[42].

7.6 Model Fitting

Model-driven approaches search for the best model appearance in the raw, processed or segmented images. During this search the geometric (pose, shape, motion, deformation), photometric, and contextual variability is exploited to generate different model instances while satisfying the geometric and photometric constraints and optimizing a suitable objective function. Examples of optimization algorithms are relaxation, dynamic programming, or gradient descent. For details about optimization theory we refer to Nocedal and Wright.[43]

The objective function expresses how likely the model instance is a priori and how similar the model instance and the data are. The measure can be formulated in terms of a total penalty, cost, or energy that should be minimized, or a probability that should be maximized. To keep the computation time feasible this strategy intrinsically assumes that the number of generated model instances is limited, which implies that a good initial hypothesis is available.

Typically the objective function consists of two components, which can easily be explained using Bayes' rule. If the optimization aims to find the model instance Φ that maximizes its posterior probability given the image data I, Bayes' rule states

42 G. Litjens, T. Kooi, B. E. Bejnordi, A. Arindra, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken and C. I. Sanchez. A survey on deep learning in medical image analysis, *Medical Image Analysis*, 42:60–88, 2017.

43 J. Nocedal and S. Wright. *Numerical Optimization*, volume XXII of *Springer Series in Operations Research and Financial Engineering*. Springer, second edition, 2006.