Chapter 2
Statistical Classification of Audiovisual Data

Abstract In this chapter we explore a mathematical model for representation of audiovisual data as a sequence of independent segments. Each segment is associated with a sample of independent identically distributed primitive features. Based on this model the classification task is reduced to a problem of complex hypothesis testing of segment homogeneity. According to this approach, several nearest neighbor criteria are implemented. The well-known special cases are emphasized for some of them, e.g., the probabilistic neural network and the minimum Jensen–Shannon divergence principle. An experimental study in the face recognition problem is presented. It is shown that the segment homogeneity testing improves the accuracy when compared with the contemporary classification methods.

2.1 Mathematical Model of the Piecewise-Regular Object

Following the description from Chap. 1, we detect an object of interest X in the audiovisual stream by any known algorithm. For instance, the Viola–Jones cascade classifier [26] can be used in image processing. For clarity, let us define the new observation X as the M-dimensional vector $[x_1, \ldots, x_M]$. Suppose $\mathbf{X} \subset \mathbb{R}^M$ is the finite nonempty set of analyzed objects. Next, the segmentation procedure is applied to the object X. The observation X is described with $K \geq 1$ segments $\{X(k)|k \in \{1, \ldots, K\}\}$. Every kth segment is defined by its boundaries $(m_1(k), m_2(k))$, i.e., the kth segment is represented as a vector $[x_{m_1(k)}, \ldots, x_{m_2(k)}]$. We assume that the segments are non-overlapped and several points of the original vector can be missed, i.e., $1 \leq m_1(1) < m_2(1) < m_1(2) < \ldots < m_1(k) < m_2(k) < \ldots < m_2(K) \leq M$. The most simple way to divide X into a set of frames is as follows: $m_1(k) = 1 + \lfloor (M \cdot (k-1))/K \rfloor, m_2(k) = \lfloor (M \cdot k)/K \rfloor$, where $\lfloor \cdot \rfloor$ is the floor round function. In fact, more complex segmentation algorithms can be applied. For example, the region growing, the clustering, and the histogram-based methods are widely used in image processing [23].

Next, we extract in each segment $X(k)$ a set $\{\mathbf{x}_j(k)|j \in \{1, \ldots, n(k)\}\}$ of $n(k) \gg 1$ feature vectors $\mathbf{x}_j(k)$ with the fixed dimension p. In the simplest case, these features are equal to the original representation of the segment, i.e., $p = 1, n(k) = m_2(k) - m_1(k) + 1, \mathbf{x}_j(k) = [x_{m_1(k)+j-1}]$.

A.V. Savchenko, *Search Techniques in Intelligent Classification Systems*,
SpringerBriefs in Optimization, DOI 10.1007/978-3-319-30515-8_2

Similarly, every reference object X_r is segmented into a sequence of $K_r \geq 1$ regular parts $\{X_r(k)|k \in \{1,\ldots,K_r\}\}$, and the kth segment is associated with a set $\{\mathbf{x}_j^{(r)}(k)|j \in \{1,\ldots,n_r(k)\}\}$ of $n_r(k)$ feature vectors $\mathbf{x}_j^{(r)}(k)$ with the same dimension p.

Let us assume [21, 22] that:

1. Vectors $\mathbf{x}_j(k)$ and $\mathbf{x}_j^{(r)}(k)$ are *random vectors*.
2. Segments $X(k)$ and $X_r(k)$ are simple random samples of i.i.d. feature vectors $\mathbf{x}_j(k)$ and $\mathbf{x}_j^{(r)}(k)$, respectively.
3. The distributions of the feature vectors are different for various instances of the same class.

As the segmentation procedure is inaccurate, every segment $X(k)$ should be compared with a set $N_r(k)$ of numbers of segments of the r-th instance that are closed to the segment k. This neighborhood is determined for a specific task individually. The alignment of the segments is illustrated in Fig. 2.1. Here the kth segment in the input image is matched not only with the corresponding kth segment of the reference image, but also with a set of the segments $N_r(k)$. If the segmentation procedure is always correct, one can assume that $N_r(k) = \begin{cases} \{k\}, & K = K_r \\ \oslash & K \neq K_r \end{cases}$.

There are two possible ways to estimate unknown class densities, namely, the *parametric* and the *nonparametric* approaches [17, 25]. Let us discover both of them in detail.

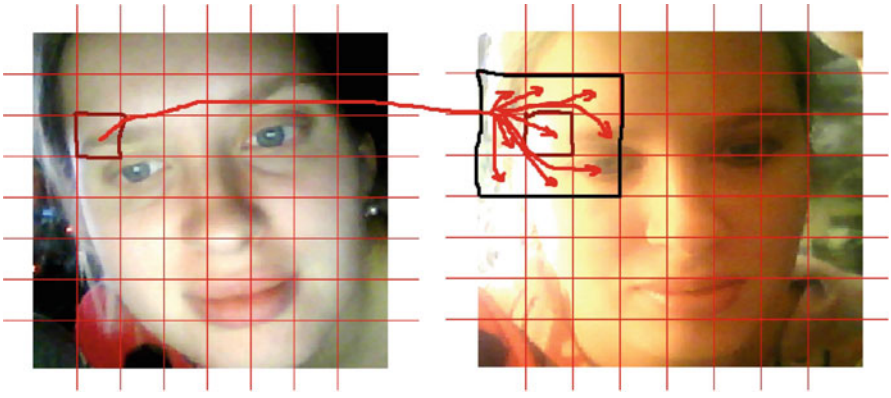

Fig. 2.1 The typical alignment of the segment of the input image (*left*) in the neighborhood of the corresponding segment in a reference image (*right*)

2.1.1 Exponential Family of Distributions

In this section it is assumed that the distributions of all segments from all classes are of multivariate exponential type generated by a fixed (for all classes) function [22]. Hence, each kth segment of each r-th instance is determined by its parameter vector $\boldsymbol{\theta}_r(k)$. The latter is estimated by using the observed (given) sample $X_r(k)$. At first, let us repeat the definition of exponential family with the notation from [8].

Definition. Let $\widetilde{X}$ be the random sample of independent identically distributed vectors $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n$. Their joint probability density $f_{\boldsymbol{\theta};n}$ is of *exponential type* generated by the function $f_0(\widetilde{X})$, if

$$f_{\boldsymbol{\theta};n}(\widetilde{X}) = \exp(\boldsymbol{\tau}(\boldsymbol{\theta}) \cdot \hat{\boldsymbol{\theta}}(\widetilde{X})) \cdot f_0(\widetilde{X}) / M(\boldsymbol{\tau}), \tag{2.1}$$

$$M(\boldsymbol{\tau}) = \int \exp(\boldsymbol{\tau}(\boldsymbol{\theta}) \cdot \hat{\boldsymbol{\theta}}(\widetilde{X})) \cdot f_0(\widetilde{X}) \cdot d\widetilde{X}, \tag{2.2}$$

where $\boldsymbol{\theta}$ is the J-dimensional parameter vector, $\hat{\boldsymbol{\theta}}(\widetilde{X})$ is an estimation of the parameter $\boldsymbol{\theta}$ using the available data (random sample) $\widetilde{X}$, and $\tau(\boldsymbol{\theta})$ is a normalizing function (the J-dimensional parameter vector), defined by the following equation

$$\int \hat{\boldsymbol{\theta}}(\widetilde{X}) \cdot f_{\boldsymbol{\theta};n}(\widetilde{X}) \cdot d\widetilde{X} \equiv \frac{d}{d\tau(\boldsymbol{\theta})} \ln M(\boldsymbol{\tau}) = \boldsymbol{\theta}, \tag{2.3}$$

if the parameter estimation $\hat{\boldsymbol{\theta}}(\widetilde{X})$ is unbiased.

The exponential family covers wide range of known distributions, e.g., polynomial, normal, etc. [8, 9].

Example. Let $\widetilde{X}$ be a sample of n independent yes/no experiments. If $\tilde{x} \in \{0, \ldots, n\}$ is the number of successes ("yes" values) in n experiments, and p is the probability of "yes," we can denote $\boldsymbol{\theta} = [n \cdot p]$, $\hat{\boldsymbol{\theta}}(\widetilde{X}) = [\tilde{x}]$, and $f_0(\widetilde{X}) = \binom{n}{\tilde{x}}$. Following (2.2) for a discrete case, one can notice that $M(\boldsymbol{\tau}) = \sum_{\tilde{x}=0}^{n} \binom{n}{\tilde{x}} \exp(\tau \cdot \tilde{x}) = (1 + \exp(\tau))^n$. Equation (2.3) can be written as $\frac{d}{d\tau(\boldsymbol{\theta})} \ln M(\boldsymbol{\tau}) = \frac{n \cdot \exp(\tau)}{1+\exp(\tau)} = n \cdot p$. Hence, $\tau = \ln(p/(1-p))$. Substituting this value to (2.1), we finally obtain the well-known *binomial* distribution $f(\tilde{x}) = \binom{n}{\tilde{x}} \cdot p^{\tilde{x}} \cdot (1-p)^{n-\tilde{x}}$.

In this section we focus on the case of full prior uncertainty and assume that the prior probabilities of each class are equal. If the classification task is reduced to a problem of statistical testing of **simple** hypothesis, the Bayesian approach [3] will be equivalent to the maximum likelihood criterion [1]. For our task, every segment is recognized with the following rule

$$\max_{r \in \{1,\ldots,R\}} \max_{k_r \in N_r(k)} f_{\hat{\boldsymbol{\theta}}(X_r(k));n(k)}(X(k)). \tag{2.4}$$

It can be shown that Eq. (2.4) is equivalent to the Kullback–Leibler minimum information discrimination rule [8]

$$\min_{r\in\{1,\ldots,R\}}\sum_{k=1}^{K}\min_{k_r\in N_r(k)}\widehat{I}(*:f_{\hat{\boldsymbol{\theta}}(X_r(k_r));n(k)};X(k)), \tag{2.5}$$

where

$$\widehat{I}(*:f_{\hat{\boldsymbol{\theta}}(X_r(k_r));n(k)};X(k))=\int f_{\hat{\boldsymbol{\theta}}(X(k));n(k)}(\widetilde{X})\cdot\ln\frac{f_{\hat{\boldsymbol{\theta}}(X(k));n(k)}(\widetilde{X})}{f_{\hat{\boldsymbol{\theta}}(X_r(k_r));n(k)}(\widetilde{X})}\cdot d\widetilde{X}. \tag{2.6}$$

2.1.2 Nonparametric Estimates of Probability Density

The major assumption of the previous section is about the exponential family of distributions (2.1)–(2.3). This assumption is known to be inappropriate for arbitrary objects. Hence, another, nonparametric approach is more popular nowadays [17]. The conditional probability density $f(X(k)|W_r(k_r))$ is usually estimated by the given training set with a kernel trick:

$$f(X(k)|W_r(k_r))=\frac{1}{(n_r(k_r))^{n(k)}}\prod_{j=1}^{n(k)}\sum_{j_r=1}^{n_r(k_r)}K(\mathbf{x}_j(k),\mathbf{x}_{j_r}^{(r)}(k_r)), \tag{2.7}$$

where $W_r(k_r)$ is the hypothesis that the distribution of the segment $X(k)$ is identical to the distribution, estimated on the basis of the training sample $X_r(k_r)$, and $K(\mathbf{x}_j(k),\mathbf{x}_{j_r}^{(r)}(k_r))$ is a kernel function [6]. For example, the Gaussian–Parzen kernel is widely used [17, 24].

$$K(\mathbf{x}_j(k),\mathbf{x}_{j_r}^{(r)}(k_r))=\frac{1}{(2\pi\sigma^2)^{p/2}}\exp\left(-\frac{1}{2\sigma^2}\cdot||\mathbf{x}_j(k)-\mathbf{x}_{j_r}^{(r)}(k_r)||^2\right), \tag{2.8}$$

where σ is the smoothing parameter, and $||\mathbf{x}_j(k)-\mathbf{x}_{j_r}^{(r)}(k_r)||$ is the Euclidean distance between p-dimensional vectors $\mathbf{x}_j(k)$ and $\mathbf{x}_{j_r}^{(r)}(k_r)$.

Hence, if the prior probabilities of all classes are equal, then the criterion

$$\max_{r\in\{1,\ldots,R\}}\sum_{k=1}^{K}\max_{k_r\in N_r(k)}\frac{1}{(n_r(k_r))^{n(k)}}\cdot\prod_{j=1}^{n(k)}\sum_{j_r=1}^{n_r(k_r)}K(\mathbf{x}_j(k),\mathbf{x}_{j_r}^{(r)}(k_r)) \tag{2.9}$$

is equivalent to the maximum likelihood rule for the statistical testing of **simple** hypothesis about distributions of the segments.

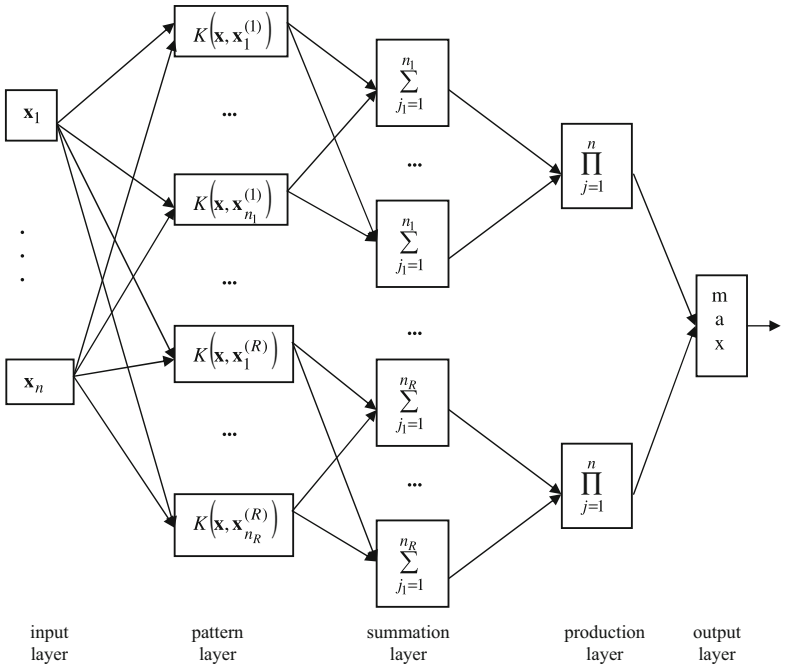

Fig. 2.2 Probabilistic neural network in the segment recognition task. The figure is reprinted from [19] with the permission of Elsevier

The criterion (2.9) corresponds the PNN [24] for statistical recognition of a segment. Its implementation is shown in Fig. 2.2. Here we do not show the segments indexes k and k_r for simplicity. In contrast with the conventional four-layered PNN with the input, pattern, summation, and output layers, which classifies one feature vector, the network (Fig. 2.2) contains additional, production layer [18] to classify the sample $X(k)$.

2.2 Classification with the Homogeneity Testing

The criteria (2.5) and (2.9) are known to be incorrect, because the true densities of segments of each class are unknown, and unbiased estimates of the parameters $\boldsymbol{\theta}_r(k)$ are used [1]. In fact, the pattern recognition problem should be reduced to a statistical testing of **complex** hypothesis of segments homogeneity [18]:

$$W_{k;k_r}^{(r)} : X(k) \textit{ and } X_r(k_r) \textit{ are homogeneous}. \tag{2.10}$$

In such case, the maximal likelihood decision

$$\max_{r\in\{1,\dots,R\}}\max_{k_r\in N_r(k)}\sup_{\overline{\boldsymbol{\theta}}(k),\overline{\boldsymbol{\theta}}_j(k_r),j=\overline{1,R}} f(\{X(k),X_1(k_r),\dots,X_R(k_r)\}|W^{(r)}_{k;k_r}) \tag{2.11}$$

is known to be asymptotically equivalent to the minimax criterion [1], if the size of the segment, i.e., the image resolution or the phoneme duration, is large. Here $\overline{\boldsymbol{\theta}}(k)$ are the possible parameters of $X(k)$, $\overline{\boldsymbol{\theta}}_j(k_r)$ are the possible parameters of the reference segment $X_j(k_r)$, and $f(\{X(k),X_1(k_r),\dots,X_R(k_r)\}|W^{(r)}_{k;k_r})$ is the joint probability density of the united sample $\{X(k),X_1(k_r),\dots,X_R(k_r)\}$, if the hypothesis $W^{(r)}_{k;k_r}$ is true.

2.2.1 Parametric Density Estimates

If we assume the exponential family of the segment distribution, then the following theorem holds [22].

Theorem 2.1. *If $\hat{\boldsymbol{\theta}}(\widetilde{X})$ is unbiased maximum likelihood estimate of the parameter vector $\boldsymbol{\theta}$ in the distribution of exponential type, then*

$$\min_{r\in\{1,\dots,R\}}\sum_{k=1}^{K}\min_{k_r\in N_r(k)}(\widehat{I}(*:f_{\hat{\boldsymbol{\theta}}(X^{(r)}_{k;k_r});n(k)};X(k))+\widehat{I}(*:f_{\hat{\boldsymbol{\theta}}(X^{(r)}_{k;k_r});n_r(k_r)};X_r(k_r))) \tag{2.12}$$

is the asymptotically minimax criterion of testing the hypothesis of the segments homogeneity (2.10), where $X^{(r)}_{k;k_r}=\{X(k),X_r(k_r)\}$ is the united sample of the segments $X(k)$ and $X_r(k_r)$.

Proof. As all vectors in the set $\{X(k),X_1(k_r),\dots,X_R(k_r)\}$ are independent, the likelihood function in (2.11) can be written as follows

$$\begin{aligned}&\sup_{\overline{\boldsymbol{\theta}}(k),\ \overline{\boldsymbol{\theta}}_j(k_r),j=\overline{1,R}} f(\{X(k),X_1(k_r),\dots,X_R(k_r)\}|W^{(r)}_{k;k_r})\\ &\quad=\sup_{\overline{\boldsymbol{\theta}}(k)}f(X(k)|W^{(r)}_{k;k_r})\cdot\textstyle\prod_{j=1}^{R}\sup_{\overline{\boldsymbol{\theta}}_j(k_r)}f(X_j(k_r)|W^{(r)}_{k;k_r}).\end{aligned} \tag{2.13}$$

If the hypothesis $W^{(r)}_{k;k_r}$ is true, i.e., the segments $X(k)$ and $X_r(k_r)$ are homogeneous, then the conditional density of $X_j(k_r)$ does not depend on the r-th instance in the case of $j\neq r$. Hence, Eq. (2.13) can be presented as

$$\sup_{\overline{\boldsymbol{\theta}}(k),\ \overline{\boldsymbol{\theta}}_j(k_r),j=\overline{1,R}} f(\{X(k),X_1(k_r),\dots,X_R(k_r)\}|W^{(r)}_{k;k_r})$$

$$= \sup_{\overline{\boldsymbol{\theta}}(k)} f(X(k)|W^{(r)}_{k;k_r}) \cdot \sup_{\overline{\boldsymbol{\theta}}_r(k_r)} f(X_r(k_r)|W^{(r)}_{k;k_r}) \cdot \prod_{\substack{j=1 \\ j\neq r}}^{R} \sup_{\overline{\boldsymbol{\theta}}_j(k_r)} f_{\overline{\boldsymbol{\theta}}_j(k_r);n_j(k_r)}(X_j(k_r))$$

$$= \frac{\sup_{\overline{\boldsymbol{\theta}}(k)} f(X(k)|W^{(r)}_{k;k_r}) \cdot \sup_{\overline{\boldsymbol{\theta}}_r(k_r)} f(X_r(k_r)|W^{(r)}_{k;k_r})}{\sup_{\overline{\boldsymbol{\theta}}_r(k_r)} f_{\overline{\boldsymbol{\theta}}_r(k_r);n_r(k_r)}(X_r(k_r))} \cdot \prod_{j=1}^{R} \sup_{\overline{\boldsymbol{\theta}}_j(k_r)} f_{\overline{\boldsymbol{\theta}}_j(k_r);n_j(k_r)}(X_j(k_r)). \tag{2.14}$$

It is possible to divide (2.14) on $\sup_{\overline{\boldsymbol{\theta}}(k)} f_{\overline{\boldsymbol{\theta}}(k);n(k)}(X(k)) \cdot \prod_{j=1}^{R} \sup_{\overline{\boldsymbol{\theta}}_j(k_r)} f_{\overline{\boldsymbol{\theta}}_j(k_r);n_j(k_r)}(X_j(k_r))$, because the expression $\prod_{j=1}^{R} \sup_{\overline{\boldsymbol{\theta}}_j(k_r)} f_{\overline{\boldsymbol{\theta}}_j(k_r);n_j(k_r)}(X_j(k_r))$ does not depend on r. Hence, the criterion (2.11) is equivalent to

$$\max_{r\in\{1,\ldots,R\}} \max_{k_r\in N_r(k)} \frac{\sup_{\overline{\boldsymbol{\theta}}(k)} f(X(k)|W^{(r)}_{k;k_r})}{\sup_{\overline{\boldsymbol{\theta}}(k)} f_{\overline{\boldsymbol{\theta}}(k);n(k)}(X(k))} \cdot \frac{\sup_{\overline{\boldsymbol{\theta}}_r(k_r)} f(X_r(k_r)|W^{(r)}_{k;k_r})}{\sup_{\overline{\boldsymbol{\theta}}_r(k_r)} f_{\overline{\boldsymbol{\theta}}_r(k_r);n_r(k_r)}(X_r(k_r))}. \tag{2.15}$$

The supremum in (2.15) is reached for the maximal likelihood estimates of the parameters $\boldsymbol{\theta}$. To estimate the conditional density, if the hypothesis $W^{(r)}_{k;k_r}$ is true, the united sample $X^{(r)}_{k;k_r}$ is used. By using the assumption of this theorem about an unbiased maximum likelihood estimate $\hat{\boldsymbol{\theta}}(\widetilde{X})$, the previous equation can be simplified:

$$\max_{r\in\{1,\ldots,R\}} \max_{k_r\in N_r(k)} \frac{f_{\hat{\boldsymbol{\theta}}(X^{(r)}_{k;k_r});n(k)}(X(k)) \cdot f_{\hat{\boldsymbol{\theta}}(X^{(r)}_{k;k_r});n_r(k_r)}(X_r(k_r))}{f_{\hat{\boldsymbol{\theta}}(X(k));n(k)}(X(k)) \cdot f_{\hat{\boldsymbol{\theta}}(X_r(k_r));n_r(k_r)}(X_r(k_r))}. \tag{2.16}$$

The latter criterion can be transformed to the NN rule (2.12) by using the same procedure of transformation of criterion (2.4)–(2.5).

Thus, the criterion (2.12) is the implementation of the parametric approach for the probabilistic model of the piecewise-regular object [21]. It can be implemented very efficiently, as the computation of the Kullback–Leibler divergence (2.6) usually requires $O(p^m)$ operations. For instance, $m = 1$ for polynomial distribution and $m = 3$ for p-variate normal distribution. Hence, the runtime complexity of the criterion (2.12) is equal to $O(p^m \cdot \sum_{r=1}^{R} \sum_{k=1}^{K} |N_r(k)|)$, where $|N_r(k)|$ is the size of the set $N_r(k)$.

2.2.2 Nonparametric Density Estimates

Theorem 2.2. *If the vectors* $\mathbf{x}_j(k)$ *and* $\mathbf{x}_j^{(r)}(k)$ *are i.i.d. random vectors with the densities, which can be represented with the Gaussian–Parzen kernel with the fixed (for all classes) smoothing parameter* σ*, then the rule*

$$\max_{r\in\{1,\ldots,R\}}\sum_{k=1}^{K}\max_{k_r\in N_r(k)}\frac{n(k)^{n(k)}\cdot(n_r(k_r))^{n_r(k_r)}}{(n(k)+n_r(k_r))^{(n(k)+n_r(k_r))}}\cdot\prod_{j=1}^{n(k)}\left(1+\frac{\sum_{j_r=1}^{n_r(k_r)}K(\mathbf{x}_j(k),\mathbf{x}_{j_r}^{(r)}(k))}{\sum_{j_1=1}^{n(k)}K(\mathbf{x}_j(k),\mathbf{x}_{j_1}(k))}\right)$$
$$\times\prod_{j_r=1}^{n_r(k_r)}\left(1+\frac{\sum_{j_1=1}^{n(k)}K(\mathbf{x}_{j_r}^{(r)}(k_r),\mathbf{x}_{j_1}(k))}{\sum_{j_{r;1}=1}^{n_r(k_r)}K(\mathbf{x}_{j_r}^{(r)}(k_r),\mathbf{x}_{j_{r;1}}^{(r)}(k_r))}\right) \tag{2.17}$$

is the asymptotically minimax criterion of testing the complex hypothesis of the segment homogeneity.

We do not show the proof of this Theorem [22], because it is very similar to the proof of the Theorem 2.1. The criterion (2.17) can be implemented in the homogeneity testing PNN (HT-PNN) (Fig. 2.3), which is, in turn, the general case of the PNN in asymptotic ($n_r(k_r)\to\infty$) [19].

Here the input layer contains not only the segment $X(k)$, but also the united sample $\{X(k),X_1(k_1),\ldots,X_R(k_R)\}$. It makes no difference between the new observation and the training samples. In the second, pattern layer the kernel function for an input object is added to a training set. The new division layer is added according to (2.17). In the production layer we multiply not only the features of the segment $X(k)$, but also the features of the r-th segment $X_r(k_r)$.

Unfortunately, the network (Fig. 2.3) has the same disadvantages [18], as the general PNN (Fig. 2.2) [24]. They both require large memory to store all training samples and the classification speed is low as the network is based on an exhaustive search through all training samples. In fact, criterion (2.17) requires the comparison of *all* features of *all* segments of *all* instances. Its runtime complexity can be written as $O(p\cdot\sum_{r=1}^{R}\sum_{k=1}^{K}\sum_{k_r\in N_r(k)}n(k)\cdot n_r(k_r))$, i.e., it is much less computationally efficient than the parametric case (2.12). Thus, the practical implementation of these rules can be unfeasible. It is known [19] that they can be simplified, if the feature vectors are discrete and certain, i.e., their domain of definition is a set $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$, where N is the number of different vectors. In such case, the segment of the input object $X(k)$ can be described with the histogram $H(k)=\{h_1(k),\ldots,h_N(k)\}$. Similarly, the segment $X_r(k)$ of the reference object can be described with the histogram $H^{(r)}(k)=\{h_1^{(r)}(k),\ldots,h_N^{(r)}(k)\}$. This definition allows to use the polynomial distribution, which is known to be of exponential type. Hence, it can be shown that the criterion (2.5) is equivalent to the Kullback–Leibler minimum information discrimination principle [8]:

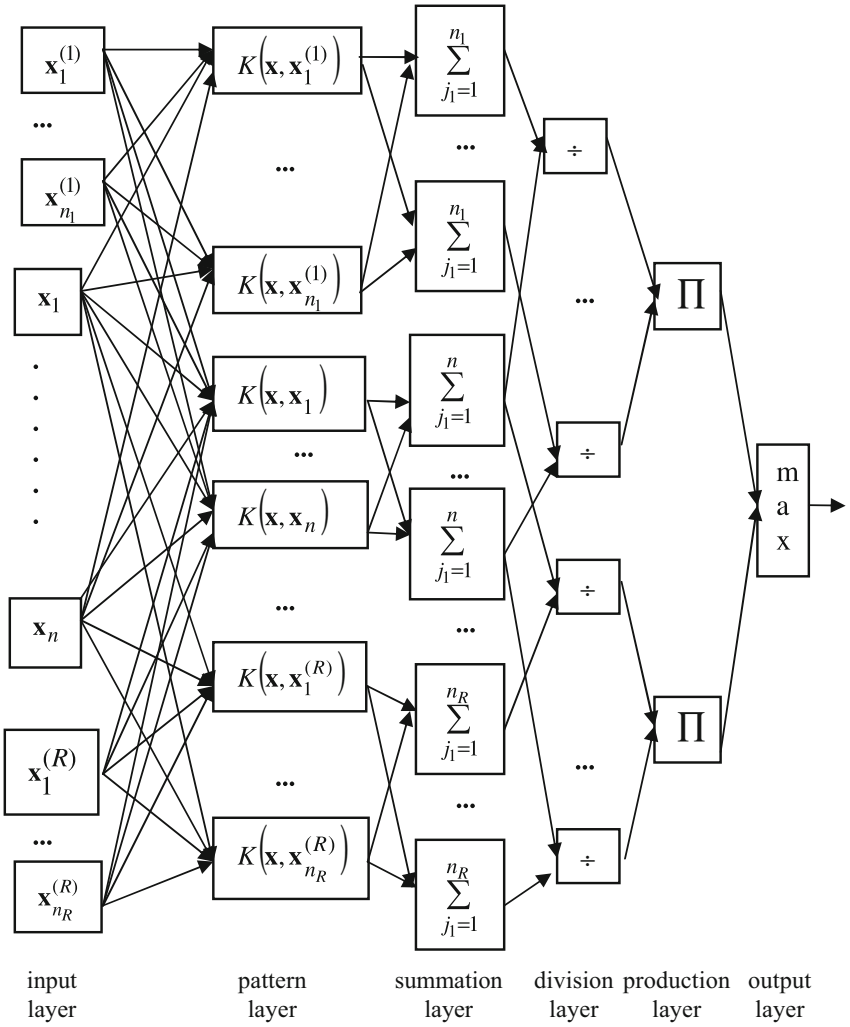

Fig. 2.3 Homogeneity testing probabilistic neural network. The figure is reprinted from [19] with the permission of Elsevier

$$\min_{r\in\{1,\ldots,R\}} \frac{1}{Kn} \sum_{k=1}^{K} \min_{k_r \in N_r(k)} \sum_{i=1}^{N} h_i(k) \ln \frac{h_i(k)}{h_i^{(r)}(k_r)}, \tag{2.18}$$

where $n = \sum_{k=1}^{K} n(k)/K$ is the average size of the segment.

Similarly, the parametric criterion (2.12) based on the homogeneity testing is equivalent to

$$\min_{r\in\{1,\dots,R\}} \frac{1}{Kn} \times \sum_{k=1}^{K} \min_{k_r\in N_r(k)} \sum_{i=1}^{N} \left(n(k)h_i(k) \ln \frac{h_i(k)}{\tilde{h}^{(r)}_{\Sigma;i}(k;k_r)} + n_r(k_r)h^{(r)}_i(k) \ln \frac{h^{(r)}_i(k)}{\tilde{h}^{(r)}_{\Sigma;i}(k;k_r)} \right), \tag{2.19}$$

where $\tilde{h}^{(r)}_{\Sigma;i}(k;k_r) = (n(k)\cdot h_i(k) + n_r(k_r)\cdot h^{(r)}_i(k_r))/(n(k)+n_r(k_r))$. If $n(k) = n_r(k_r)$, this criterion is equivalent to the NN rule with the Jensen–Shannon divergence widely used in various pattern recognition tasks [13].

At the same time, if the nonparametric approach is used, an obvious generalization of the PNN (2.9) can be transformed to the NN rule with the following distance

$$\rho_{\text{PNN}}(X, X_r) = \frac{1}{Kn} \sum_{k=1}^{K} \min_{k_r\in N_r(k)} \sum_{i=1}^{N} h_i(k) \ln \frac{h_{K;i}(k)}{h^{(r)}_{K;i}(k_r)}, \tag{2.20}$$

where $h_{K;i}(k) = \sum_{j=1}^{N} K_{ij}h_i(k)$ and $h^{(r)}_{K;i}(k) = \sum_{j=1}^{N} K_{ij}h^{(r)}_i(k)$ are the convolutions of the histograms with the kernel $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. The HT-PNN (2.17) for the discrete patterns is implemented in the NN rule with the dissimilarity measure [19]

$$\rho_{\text{HT-PNN}}(X, X_r) = \frac{1}{Kn} \times \sum_{k=1}^{K} \min_{k_r\in N_r(k)} \sum_{i=1}^{N} \left(n(k)h_i(k) \ln \frac{h_{K;i}(k)}{\tilde{h}^{(r)}_{\Sigma;K;i}(k;k_r)} + n_r(k_r)h^{(r)}_i(k) \ln \frac{h^{(r)}_{K;i}(k)}{\tilde{h}^{(r)}_{\Sigma;K;i}(k;k_r)} \right) \tag{2.21}$$

where $\tilde{h}^{(r)}_{\Sigma;K;i}(k;k_r) = (n(k)\cdot h_{K;i}(k) + n_r(k_r)\cdot h^{(r)}_{K;i}(k_r))/(n(k)+n_r(k_r))$.

One can notice that expressions (2.18) and (2.19) are the special cases of (2.20) and (2.21), if the discrete delta function is used as a kernel: $K_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$. Runtime complexity of (2.21) is $O(N \cdot \sum_{r=1}^{R} \sum_{k=1}^{K} |N_r(k)|)$, i.e., the computing efficiency is in average $n^2 \cdot p/N$-times higher than the efficiency of the general HT-PNN (2.17).

2.3 Applications in Image Classification

2.3.1 Constrained Face Recognition

Let a set of $R > 1$ gray-scale images $\{X_r\}, r \in \{1, \ldots, R\}$ be given. In image recognition, it is required to assign a new image X to one of the R classes specified by these reference images. At first, every image is put in correspondence with a set of feature descriptors [25]. The common part of most of the modern algorithms is to divide the whole neighborhood into a regular grid of $S_1 \times S_2$ blocks, S_1 rows and S_2 columns (in our previous notation, $K = K_1 = K_2 = \ldots = K_R = S_1 \cdot S_2$), and separately evaluate the histogram $H^{(r)}(s_1, s_2) = [h_1^{(r)}(s_1, s_2), \ldots, h_N^{(r)}(s_1, s_2)]$ of the gradient orientations for each block $(s_1, s_2), s_1 \in \{1, \ldots, S_1\}, s_2 \in \{1, \ldots, S_2\}$, of the reference image X_r [2, 12]. The same procedure is repeated to evaluate the histograms of oriented gradients (HOGs) $H(s_1, s_2) = \{h_1(s_1, s_2), \ldots, h_N(s_1, s_2)\}$ based on the input image X.

The second part is classifier design. According to the model of the piecewise-regular object and in view of the small spatial deviations due to misalignment after object detection, the following dissimilarity measure with the mutual alignment and the matching of the HOGs in the Δ-neighborhood of each segment is used [20, 22]:

$$\min_{r \in \{1,\ldots,R\}} \sum_{s_1=1}^{S_1} \sum_{s_2=1}^{S_2} \min_{|\Delta_1| \le \Delta, |\Delta_2| \le \Delta} \rho_H(H(s_1 + \Delta_1, s_2 + \Delta_2), H^{(r)}(s_1, s_2)). \tag{2.22}$$

Here $\rho_H(H(s_1+\Delta_1, s_2+\Delta_2), H^{(r)}(s_1, s_2))$ is an arbitrary distance between the HOGs $H(s_1 + \Delta_1, s_2 + \Delta_2)$ and $H^{(r)}(s_1, s_2)$. The neighborhood $N_r(k)$ of the cell (s_1, s_2) is described with the set $\{(\tilde{s}_1, \tilde{s}_2) \,|\, |\tilde{s}_1 - s_1| \le \Delta, |\tilde{s}_2 - s_2| \le \Delta\}$.

In this section, we examine the square of the Euclidean (L_2) distance:

$$\rho_2(H(s_1, s_2), H^{(r)}(s_1, s_2)) = \sum_{i=1}^{N} (h_i(s_1, s_2) - h_i^{(r)}(s_1, s_2))^2, \tag{2.23}$$

and the described distances based on statistical approach, namely, the Kullback–Leibler (2.18) and the Jensen–Shannon (2.19) divergences, the PNN (2.20) and the segment homogeneity testing (2.21). Additionally, we use the state-of-the-art SVM classifiers of the HOGs, the SIFT method [12], and face recognition methods from OpenCV library,[1] namely, the eigenfaces [16], the fisherfaces [10], and the LBP (Local Binary Patterns) histograms [11]. All these methods were implemented in the C++ Windows console application,[2] which was compiled in Visual C++ Express 2013 environment (optimization by speed). We use the multithreading to

[1] http://www.opencv.org.

[2] https://sites.google.com/site/andreyvsavchenko/ImageRecognitionTest_VS13.zip.

Fig. 2.4 Sample images from the AT&T dataset

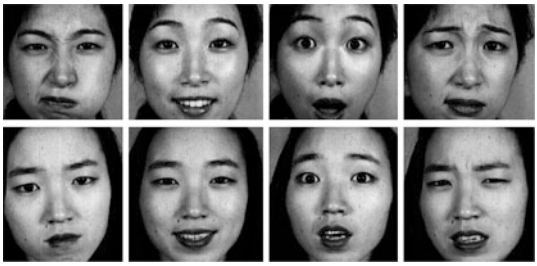

Fig. 2.5 Sample images from the JAFFE dataset

make the brute-force search (2.22) faster. Each thread is implemented with Windows ThreadPool API and operates only on a subset of the database. The whole training sample is divided into 8 distinct parts, i.e., we look for the NN (2.22) in 8 parallel threads. The laptop with the following configuration (CPU: 4 core i7 2 GHz and 6 GB RAM) was used to run this application.

The experimental study deals with the constrained face recognition task [10]. Three popular datasets were used. The AT&T (former ORL, Fig. 2.4) dataset[3] is well known by a various face foreshortening on the image. It contains 400 photos of 40 persons (10 photos per person). The Japanese Female Facial Expression (JAFFE, Fig. 2.5) database[4] contains 213 images of 10 female persons (more than 20 photos per person). The latter dataset is used in either face classification or the facial expression recognition tasks. The FERET dataset (Fig. 2.6)[5] is the standard set to estimate of constrained face recognizers. From this database, 2720 frontal facial images of $C = 994$ persons were selected.

Instead of the standard methodology of tuning the parameters by splitting the whole dataset into the training, validation, and testing sets, we used large Essex face database (7900 images, 395 persons).[6] In fact, the similar idea is popular in training the DNN-based face recognizers [27]. The tenfold cross-validation was

[3] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[4] http://www.kasrl.org/jaffe.html.

[5] http://www.itl.nist.gov/iad/humanid/feret/feret_master.html.

[6] http://cswww.essex.ac.uk/mv/allfaces/index.html.

Fig. 2.6 Sample images from the FERET dataset

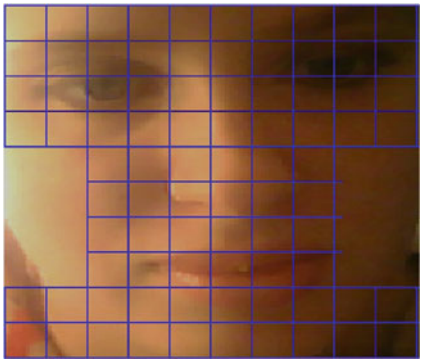

Fig. 2.7 Initial segmentation of the facial image

applied to obtain the following values of parameters. The median filter with the window size (3×3) was applied to remove the noise in the detected faces. The number of bins in the HOG is equal to $N = 8$. The Gaussian kernel smoothing parameter $\sigma = 0.71$. The following neighborhood sizes were tested: $\Delta = 0$ and $\Delta = 1$. All facial images are divided into regular segments (blocks) by 10×10 grid ($S_1 = S_2 = 10$), if $\Delta = 1$, and by 5×5 grid ($S_1 = S_2 = 5$), if $\Delta = 0$. Next, we use the prior information about the domain: the mouth, the noise, and the eyes regions are extracted in the facial image (Fig. 2.7). In the case of 10×10 grid size, the eyes region is covered by the top 4×10 cells, the nose and the mouth regions are covered by (overlapping) 6×6 and 2×10 cells, respectively, (Fig. 2.7). The dissimilarity of two facial images is estimated as the weighted sum of the dissimilarities (2.22) between the corresponding regions.

The accuracy is estimated by the following cross-validation procedure. At first, the number of photos per one person n_p is fixed. We randomly choose n_p photos for each person and put them into the reference database. Other photos from the dataset form the testing set. Then we estimate the error rate of the testing set classification. This experiment is repeated 20 times. Finally, we estimate the mean and the standard deviation of the error rate for all experiments. The number of instances is not the same for different classes in FERET dataset. Hence, in this case, we fix the size of the training sample R instead of n_p.

The error rates are presented in Tables 2.1, 2.2, and 2.3. The lowest error rate for the fixed size of the training set and each group of face recognition methods (conventional classifiers, NN rule with $\Delta = 0$ and $\Delta = 1$) is highlighted in bold. Here we do not show the results of the fisherfaces method applied to the FERET dataset (Table 2.3), as this method cannot be used, if the training set contains only one instance for *any* class. The average time to classify one photo $\bar{t}$ for a fixed size of the training sample R is shown in Table 2.4. The proposed methods are marked by bold in this table.

Here, firstly, the quality of the face recognizers from OpenCV library (eigenfaces, fisherfaces, and the LBP histograms) is not appropriate in most cases. For instance, the accuracy of eigenfaces is 10–15 % higher, when compared with the HT-PNN (2.21) for FERET dataset. The results significantly depend on the dataset: eigenfaces is the best choice for AT&T dataset, fisherfaces is appropriate for JAFFE and $n_p \geq 3$, and the LBP histograms are much preferable for complex FERET dataset. It is necessary to emphasize that the state-of-the-art SIFT is one of the best methods for AT&T dataset. It is even better than the comparison of the HOGs (2.22) without their alignment ($\Delta = 0$). However, the performance of the SIFT method is several orders of magnitude worse (see Table 2.4). Hence, it is impossible to use SIFT in practice, if the real-time processing is required. However, the SIFT accuracy in our experiments is much higher, when compared with other local descriptors (SURF, FAST, etc.).

Secondly, the error rate of the state-of-the-art SVM is the lowest, only in the case of large training sample for simple AT&T and JAFFE databases. At the same time, even in this case the accuracy of the alignment of HOGs ($\Delta = 1$) is 2.3–3.2 % higher than the accuracy of SVM.

Thirdly, the error rate of the NN rule ($\Delta = 0$) with the Euclidean distance (2.23) is too high. Moreover, we confirmed that the accuracy of the PNN (2.20) is less than the accuracy of criterion based on the homogeneity testing (2.21). According to the McNemar's test with the confidence level 0.05, this improvement of the classifier (2.21) is statistically significant. In fact, the Jensen–Shannon divergence is a special case of the dissimilarity measure (2.21) with the segment homogeneity testing, if $\sigma \to 0$. Our experimental results confirm that our approach with the segment homogeneity testing is much more robust to the deviation of the smoothing parameter than the PNN [18].

Finally, the most important conclusion here is that the segment homogeneity testing is the best choice in most cases. And the alignment of the HOGs ($\Delta = 1$) is characterized by statistically significant higher accuracy than the conventional approach ($\Delta = 0$). Unfortunately, this alignment leads to a worse performance: traditional distance computation ($\Delta = 0$) is 9 ($(2 \cdot 1 + 1)^2$)-times faster than the HOGs alignment ($\Delta = 1$).

In the next experiment we measure the influence of the noise presence in the test set, as it is required in the objective function (1.1). We artificially add a random noise from the range $[-x_\eta; x_\eta]$ to *each* pixel of the image from the test set, where $x_\eta \in \{0, 3, 5\}$. Error rate was estimated by 100-times repeated random sub-sampling cross-validation. Dependence of estimated error rates of the criterion (2.22) on x_η

Table 2.1 Face recognition error rate $\overline{\alpha}_\eta$ [%], AT&T dataset

Algorithm	$n_p = 1$	$n_p = 2$	$n_p = 3$	$n_p = 4$	$n_p = 5$
SIFT	**26.3±2.3**	**12.9±2.1**	**7.8±2.2**	**4.4±2.1**	**2.9±2.3**
Eigenfaces	28.9±2.5	17.2±3.2	10.2±1.7	7.7±1.9	5.6±1.7
Fisherfaces	31.4±2.0	21.6±2.3	12.1±1.5	9.3±1.7	8.4±1.7
LBP histograms	41.5±3.6	26.0±2.8	16.7±3.2	11.8±3.8	8.3±1.8
HOG, SVM	31.0±3.1	16.9±3.3	**8.8±1.4**	**5.6±2.1**	**3.4±1.1**
HOG (2.22), Euclidean (2.23), $\Delta = 0$	30.9±4.1	16.3±3.2	11.1±2.4	7.7±1.6	5.4±1.6
HOG (2.22), PNN (2.20), $\Delta = 0$	32.1±4.1	15.7±3.2	11.0±2.2	7.8±2.2	4.5±1.2
HOG (2.22), Kullback–Leibler (2.18), $\Delta = 0$	34.1±3.8	15.5±2.9	10.4±2.6	7.2±2.1	4.6±0.9
HOG (2.22), Jensen–Shannon (2.19), $\Delta = 0$	30.8±4.4	15.3±3.1	10.5±2.4	7.8±1.6	4.2±1.5
HOG (2.22)**, HT-PNN** (2.21)**,** $\Delta = 0$	**29.4±3.0**	**15.1±3.0**	9.3±2.9	6.8±1.4	4.1±1.7
HOG (2.22), Euclidean (2.23), $\Delta = 1$	23.5±2.6	9.8±2.8	7.1±2.5	3.4±1.5	2.4±1.4
HOG (2.22), PNN (2.20), $\Delta = 1$	21.9±2.2	9.2±2.4	6.1±1.9	2.8±0.9	1.9±1.0
HOG (2.22), Kullback–Leibler (2.18), $\Delta = 1$	24.3±2.7	9.6±2.4	7.5±2.2	3.8±1.3	2.3±1.4
HOG (2.22), Jensen–Shannon (2.19), $\Delta = 1$	21.0±3.2	8.9±2.4	5.3±2.2	2.3±1.0	1.2±0.7
HOG (2.22)**, HT-PNN** (2.21)**,** $\Delta = 1$	**20.5±2.3**	**8.6±2.3**	**5.3±2.1**	**2.4±0.9**	**1.1±0.7**

Table 2.2 Face recognition error rate $\overline{\alpha}_\eta$ [%], JAFFE dataset

Algorithm	$n_p = 1$	$n_p = 2$	$n_p = 3$	$n_p = 4$	$n_p = 5$
SIFT	**21.1±8.7**	**11.5±4.7**	7.3±6.4	5.3±4.9	4.0±4.2
Eigenfaces	24.2±7.3	14.0±6.7	11.2±7.2	8.3±6.6	7.3±4.9
Fisherfaces	28.5±7.6	14.5±7.9	**6.2±6.0**	**4.3±4.0**	**2.6±2.8**
LBP histograms	29.5±6.5	12.7±4.2	7.8±3.8	5.6±2.6	3.6±2.6
HOG, SVM	18.8±7.9	9.6±3.9	6.9±5.1	4.6±4.4	2.9±3.1
HOG (2.22), Euclidean (2.23), $\Delta = 0$	18.1±8.2	7.2±5.5	6.3±6.3	4.7±2.9	3.1±2.7
HOG (2.22), PNN (2.20), $\Delta = 0$	19.5±9.5	10.0±6.7	7.0±6.5	5.4±4.4	3.9±4.0
HOG (2.22), Kullback–Leibler (2.18), $\Delta = 0$	19.7±9.2	9.5±7.3	6.1±6.6	4.9±4.6	4.2±4.0
HOG (2.22), Jensen–Shannon (2.19), $\Delta = 0$	17.6±8.2	7.2±5.0	5.8±4.9	4.3±3.2	3.1±2.7
HOG (2.22)**, HT-PNN** (2.21), $\Delta = 0$	**16.8±7.8**	**6.3±5.0**	**4.7±4.9**	**3.5±1.8**	**2.8±2.6**
HOG (2.22), Euclidean (2.23), $\Delta = 1$	17.4±9.6	8.3±6.3	5.9±6.3	3.6±4.4	2.1±2.9
HOG (2.22), PNN (2.20), $\Delta = 1$	13.6±8.8	5.3±4.2	**4.1±4.5**	**2.0±4.6**	**1.0±1.5**
HOG (2.22), Kullback–Leibler (2.18), $\Delta = 1$	16.3±9.3	6.7±5.2	4.6±5.0	2.5±3.2	1.7±1.5
HOG (2.22), Jensen–Shannon (2.19), $\Delta = 1$	12.8±7.6	5.3±3.9	4.2±4.3	2.4±2.6	1.1±1.7
HOG (2.22)**, HT-PNN** (2.21), $\Delta = 1$	**12.2±7.5**	**5.0±3.5**	**4.1±4.1**	2.2±2.2	1.2±1.8

Table 2.3 Face recognition error rate $\overline{\alpha}_\eta$ [%], FERET dataset

Algorithm	$R = 1030$	$R = 1110$	$R = 1370$	$R = 1730$
SIFT	37.0±1.2	30.8±2.0	20.4±2.2	18.7±1.6
Eigenfaces	41.0±0.9	35.5±1.6	27.7±1.3	22.0±1.5
Fisherfaces	–	–	–	–
LBP histograms	**29.5±1.0**	**23.7±0.9**	**15.9±1.1**	**10.5±0.8**
HOG, SVM	29.3±1.4	24.7±1.2	12.8±1.3	13.4±0.4
HOG (2.22), Euclidean (2.23), $\Delta = 0$	29.0±1.2	25.0±1.3	11.7±1.6	9.7±0.3
HOG (2.22), PNN (2.20), $\Delta = 0$	26.8±1.3	25.1±1.0	10.8±1.4	7.8±0.3
HOG (2.22), Kullback–Leibler (2.18), $\Delta = 0$	27.0±1.0	22.7±0.9	11.6±1.4	8.3±0.2
HOG (2.22), Jensen–Shannon (2.19), $\Delta = 0$	26.0±1.1	21.6±0.9	10.2±1.3	7.5±0.3
HOG (2.22)**, HT-PNN** (2.21)**,** $\Delta = 0$	**25.6±1.2**	**21.3±0.7**	**9.4±1.2**	**7.0±0.4**
HOG (2.22), Euclidean (2.23), $\Delta = 1$	25.7±1.4	21.2±1.0	9.9±1.5	7.2±0.2
HOG (2.22), PNN (2.20), $\Delta = 1$	22.8±1.4	17.6±1.1	8.5±1.3	5.2±0.3
HOG (2.22), Kullback–Leibler (2.18), $\Delta = 1$	24.2±0.8	19.5±1.0	9.3±1.4	6.5±0.1
HOG (2.22), Jensen–Shannon (2.19), $\Delta = 1$	22.9±0.7	17.8±0.9	8.3±1.2	5.2±0.6
HOG (2.22)**, HT-PNN** (2.21)**,** $\Delta = 1$	**22.5±1.3**	**16.9±0.5**	**7.7±1.2**	**4.8±0.4**

Table 2.4 Average face recognition time $\bar{t}$ [ms]

Algorithm	AT&T ($R = 200$)	JAFFE ($R = 50$)	FERET ($R = 1730$)
SIFT	110.26±0.51	10.48±0.13	1435.10±0.51
Eigenfaces	4.16±0.09	0.77±0.04	20.46±0.33
Fisherfaces	0.52±0.03	0.113±0.01	–
LBP histograms	15.81±0.15	5.14±0.13	128.61±0.95
HOG, SVM	0.04±0.01	0.01±0.01	2.84±0.06
HOG (2.22), Euclidean (2.23), $\Delta = 0$	0.35±0.01	0.12±0.01	3.00±0.06
HOG (2.22), PNN (2.20), $\Delta = 0$	0.41±0.03	0.17±0.08	5.05±0.22
HOG (2.22), Kullback–Leibler (2.18), $\Delta = 0$	0.40±0.1	0.18±0.04	5.22±0.28
HOG (2.22), Jensen–Shannon (2.19), $\Delta = 0$	0.64±0.24	0.22±0.03	8.12±0.45
HOG (2.22)**, HT-PNN** (2.21)**,** $\Delta = 0$	0.66±0.13	0.22±0.07	8.57±0.43
HOG (2.22), Euclidean (2.23), $\Delta = 1$	0.70±0.02	0.19±0.04	8.28±0.17
HOG (2.22), PNN (2.20), $\Delta = 1$	7.10±0.20	2.06±0.22	93.09±1.01
HOG (2.22), Kullback–Leibler (2.18), $\Delta = 1$	7.43±0.33	1.96±0.31	83.74±1.01
HOG (2.22), Jensen–Shannon (2.19), $\Delta = 1$	14.91±0.89	4.27±0.74	176.84±4.24
HOG (2.22)**, HT-PNN** (2.21)**,** $\Delta = 1$	15.87±0.92	4.13±0.22	179.27±5.63

Table 2.5 Face recognition error rate $\overline{\alpha}_\eta$ [%] in dependence of the noise level, AT&T dataset ($R = 80$), criterion (2.22)

Dissimilarity measure	$x_\eta = 0$	$x_\eta = 1$	$x_\eta = 3$	$x_\eta = 5$	$x_\eta = 10$
Euclidean (2.23), $\Delta = 0$	16.3±3.2	16.5±3.2	16.6±3.4	16.6±3.4	19.1±3.5
PNN (2.20), $\Delta = 0$	15.7±3.2	16.0±3.2	16.0±3.2	16.1±3.2	**18.2±3.4**
Kullback–Leibler (2.18), $\Delta = 0$	15.5±2.9	15.4±3.2	16.0±3.3	17.0±3.2	20.1±3.6
Jensen–Shannon (2.19), $\Delta = 0$	15.3±3.1	**15.0±3.1**	15.1±3.3	15.9±3.4	18.4±4.1
HT-PNN (2.21), $\Delta = 0$	**15.1±3.0**	15.2±3.1	**15.0±3.1**	**15.6±3.1**	18.3±3.7
Euclidean (2.23), $\Delta = 1$	9.8±2.8	10.8±2.8	11.3±2.9	12.3±2.9	16.6±3.4
PNN (2.20), $\Delta = 1$	9.2±2.4	9.8±2.4	10.8±2.4	11.4±2.5	**14.7±2.9**
Kullback–Leibler (2.18), $\Delta = 1$	9.6±2.4	10.8±2.5	11.7±2.4	12.6±2.7	19.7±3.6
Jensen–Shannon (2.19), $\Delta = 1$	8.9±2.4	9.5±2.4	10.6±2.6	11.4±2.7	15.5±3.2
HT-PNN (2.21), $\Delta = 1$	**8.6±2.3**	**9.3±2.3**	**10.4±2.5**	**11.0±2.4**	14.8±2.9

Table 2.6 Face recognition error rate $\overline{\alpha}_\eta$ [%] in dependence of the noise level, FERET dataset ($R = 1370$), criterion (2.22)

Dissimilarity measure	$x_\eta = 0$	$x_\eta = 1$	$x_\eta = 3$	$x_\eta = 5$	$x_\eta = 10$
Euclidean (2.23), $\Delta = 0$	11.7±1.6	12.4±1.6	13.6±1.6	16.8±1.7	35.0±1.9
PNN (2.20), $\Delta = 0$	10.8±1.4	11.5±1.4	12.1±1.5	14.3±1.6	29.0±1.7
Kullback–Leibler (2.18), $\Delta = 0$	11.6±1.4	13.0±1.4	14.2±1.4	17.1±1.5	43.1±1.7
Jensen–Shannon (2.19), $\Delta = 0$	10.2±1.3	10.8±1.4	11.9±1.3	14.7±1.4	32.3±1.5
HT-PNN (2.21), $\Delta = 0$	**9.4±1.2**	**9.9±1.2**	**10.9±1.2**	**12.9±1.3**	**28.3±1.4**
Euclidean (2.23), $\Delta = 1$	9.9±1.5	10.5±1.5	12.4±1.5	18.8±1.9	52.2±1.8
PNN (2.20), $\Delta = 1$	8.5±1.3	9.0±1.3	10.9±1.8	15.6±1.7	**45.8±2.4**
Kullback–Leibler (2.18), $\Delta = 1$	9.3±1.4	9.9±1.4	12.0±1.7	19.9±1.7	62.8±2.6
Jensen–Shannon (2.19), $\Delta = 1$	8.3±1.2	8.6±1.2	9.8±1.4	18.6±1.7	54.9±1.7
HT-PNN (2.21), $\Delta = 1$	**7.7±1.2**	**8.3±1.2**	**9.4±1.3**	**14.5±1.4**	46.0±1.3

for FERET and AT&T datasets is shown in Tables 2.5 and 2.6, respectively. The best results in each group of dissimilarity measure ($\Delta = 0$ and $\Delta = 1$) in each column are highlighted in bold.

Here, firstly, the segment homogeneity testing classifier (2.21) is quite reliable to the small noise in the testing sample. For instance, the accuracy is decreased to 1.5–1.8 % for the HT-PNN, if $x_\eta \leq 3$. Such a decrease is slightly lower, when compared with other dissimilarity measures. Secondly, the alignment of HOGs ($\Delta = 1$) is characterized by the higher accuracy than the most widely used case ($\Delta = 0$) in the case of the small noise ($x_\eta \leq 3$). However, the application of such alignment significantly *decreases* the recognition rate for more complex FERET dataset and the high noise level $x_\eta = 5$. Addition of large noise makes the estimated distribution of the gradient orientation (i.e., the HOG) to be similar to the HOGs of many other segments. By using the known "bias-variance dilemma" [4], it is necessary to use the simple classifiers (e.g., the criterion (2.22) with $\Delta = 0$), if the available training set is not representative. However, even in the case of the high noise ($x_\eta = 10$), the HOGs alignment ($\Delta = 1$) is preferable for AT&T dataset with the small number of classes C (see Table 2.5). The final conclusion in this

experiment is the lowest error rate of the segment homogeneity testing (2.21) in comparison with the other distances for a fixed noise value x_η and neighborhood size Δ.

2.3.2 Deep Neural Networks

In this section we briefly examine the application of modern deep neural networks in the unconstrained face recognition problem [5, 27]. The popular Caffe framework [7] is used in our experiments. The 4096 non-negative features are extracted with the very deep CNN of the Oxford's Visual Geometry Group trained from scratch using over 2.5 million images of celebrities collected from the web [15]. These feature vectors were normalized, which amounts to treating them as probability distributions and matching them with the PNN and the HT-PNN discussed earlier in this Chapter. The error rates of the CNN-based features were compared with the classification of the HOGs (2.22). The parameter Δ was set to be equal to 0. We additionally used the image features, which were successfully applied to the unconstrained face recognition task [14], namely, the union of the HOG, LBP, and four Gabor filters with the PCA-based extraction of 1500 main features. The following classifiers were implemented: k-NN, SVM, and Linearly Approximated Sparse Representation-based Classification (LASRC) algorithm. Further details can be found in the original paper [14] and the Matlab source code at the accompanied web site.[7]

In the first experiment the PubFig83 (Fig. 2.8) database[8] with 13813 images of 83 famous persons was used. The error rates for several sizes of the training set R and 100-times repeated random sub-sampling cross-validation are presented in Table 2.7. The lowest error rate for fixed features in each column is marked in bold.

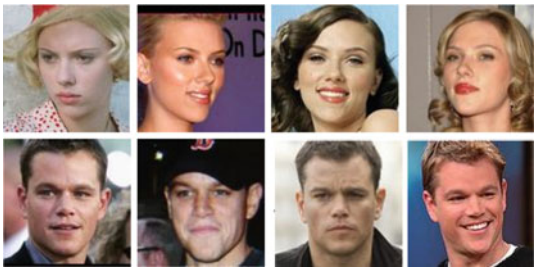

Fig. 2.8 Sample images from the PubFig83 dataset

[7] http://enriquegortiz.com/wordpress/enriquegortiz/research/face-recognition/webscale-face-recognition.

[8] http://vision.seas.harvard.edu/pubfig83.

Table 2.7 Face recognition error rate $\overline{\alpha}_\eta$ [%], PubFig83 dataset

Algorithm	$R = 100$	$R = 1300$	$R = 2700$	$R = 6900$
HOG+LBP+Gabor+PCA, 1-NN	90.7±0.7	76.6±1.1	70.6±1.0	60.8±0.9
HOG+LBP+Gabor+PCA, SVM	**83.3±0.6**	**51.4±0.9**	**41.2±1.0**	**30.3±1.0**
HOG+LBP+Gabor+PCA, LASRC	88.0±0.9	84.5±0.9	60.7±1.1	41.9±1.0
HOG (2.22), Euclidean (2.23)	91.5±0.8	82.0±1.0	76.8±1.1	64.6±1.0
HOG (2.22), PNN (2.20)	90.0±0.7	78.9±0.9	73.0±1.0	62.7±1.2
HOG (2.22)**, HT-PNN** (2.21)	**89.1±0.7**	**75.9±1.1**	**69.3±1.0**	**59.2±1.1**
DNN features, Euclidean (2.23)	28.9±0.3	9.4±0.2	7.5±0.2	5.6±0.2
DNN features, PNN (2.20)	28.3±0.4	9.2±0.2	7.4±0.2	5.6±0.2
DNN features, HT-PNN (2.21)	**28.2±0.3**	**8.8±0.2**	**7.0±0.2**	**5.2±0.1**

Fig. 2.9 Sample images from the LFW dataset

These results support the known superiority of the DNNs in image recognition: their accuracy is 25–60 % higher than the accuracy of the best known conventional image features [14]. Similarly to our study in Sect. 2.3.1, the segment homogeneity testing (HT-PNN) showed the best error rate for the HOG features, but this error rate is extremely high for such a complex task (Fig. 2.8). Finally, the HT-PNN classifier can be successfully applied with the DNN-based features; however, the difference in accuracies with other dissimilarity measures is rather low.

In the second experiment much more complex Local Faces in the Wild (LFW) dataset[9] is explored (Fig. 2.9). This dataset is the standard de facto in the comparison of contemporary face verification methods. We took 1680 persons from this dataset with two or more photos. The resulted dataset contains 9034 facial photos of these persons. The error rates are shown in Table 2.8.

Here SVM classifier was not able to converge, so it was not presented in this table. The LASRC method needs more than one instance per each class and raises an exception in the case of $R = 1680$ photos in the training set. The results are very similar to the previous experiment, though the error rate of the DNN features is 10 % higher. However, the DNNs are 35–60 % more accurate in comparison with other methods. Moreover, our segment homogeneity testing procedure with the HOG features is more preferable than the classification of the complex image features

[9] http://vis-www.cs.umass.edu/lfw/.

Table 2.8 Face recognition error rate $\overline{\alpha}_\eta$ [%], LFW dataset

Algorithm	$R = 1680$	$R = 4550$	$R = 6500$
HOG+LBP+Gabor+PCA, 1-NN	**93.1±1.1**	73.1±1.0	58.9±0.9
HOG+LBP+Gabor+PCA, LASRC	–	**65.7±0.8**	**49.2±0.8**
HOG (2.22), Euclidean (2.23)	91.1±1.0	72.4±0.9	75.0±1.1
HOG (2.22), PNN (2.20)	88.6±1.0	67.9±1.1	69.6±1.0
HOG (2.22)**, HT-PNN** (2.21)	**86.2±1.0**	**63.3±0.9**	**65.5±0.8**
DNN features, Euclidean (2.23)	27.4±0.9	15.5±0.3	17.2±0.4
DNN features, PNN (2.20)	26.7±0.9	15.2±0.4	16.8±0.4
DNN features, HT-PNN (2.21)	**26.5±1.0**	**15.1±0.5**	**16.5±0.5**

(HOG+LBP+Gabor+PCA) in this experiment, because the number of instances per class here is much lower than for the PubFig83 dataset. Finally, one can notice that the accuracy of the DNN features is decreased, when the size of the training set is increased from 4550 to 6500 instances. In fact, many facial images of several persons are very different with the other images of these persons, and the DNNs cannot be used to resolve this uncertainty. However, this effect is not observed for other features due to their low accuracy.

Thus, in this chapter we described the novel segment homogeneity testing classifier. We experimentally showed that its error rate is the worst in most cases, when compared with other popular methods. Unfortunately, the average recognition time of the segment homogeneity testing is usually the worst (Table 2.4), especially, for large size $|N_r(k)|$ of the segment neighborhood. Hence, this approach is usually not suitable in terms of our goal (1.1) for a reasonable choice of the maximal processing time t_0. The main goal of this monograph is to look for the ways to speed-up the exhaustive search procedures (2.12), (2.17), and (2.21) by using the properties of the classifier with the segment homogeneity testing. We explore the possible search techniques in the next two chapters.

References

[1] Borovkov, A.A.: Mathematical Statistics. Gordon and Breach Science Publishers, Amsterdam (1998)
[2] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 886–893 (2005)
[3] Denison, D.G.: Bayesian Methods for Nonlinear Classification and Regression. Wiley Series in Probability and Statistics, vol. 386. Wiley, New York (2002)
[4] Haykin, S.O.: Neural Networks and Learning Machines, 3rd edn. Prentice Hall, Harlow (2008)
[5] Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database for studying face recognition in unconstrained environments. In: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition (2008)

[6] Jenssen, R., Erdogmus, D., Principe, J., Eltoft, T.: Some equivalences between kernel methods and information theoretic methods. J. VLSI Sig. Proc. **45**, 49–65 (2006)
[7] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
[8] Kullback, S.: Information Theory and Statistics. Dover Publications, New York (1997)
[9] Lehmann, E.L., Romano, J.P.: Testing Statistical Hypotheses, 3rd edn. Springer, New York (2008)
[10] Li, S.Z., Jain, A.K. (eds.): Handbook of Face Recognition, 2nd edn. Springer, London/ New York (2011)
[11] Liao, S., Zhu, X., Lei, Z., Zhang, L., Li, S.Z.: Learning multi-scale block local binary patterns for face recognition. In: Lee, S.W., Li, S.Z. (eds.) Proceedings of the International Conference on Advances in Biometrics (ICB), vol. 4642, pp. 828–837. Springer (2007)
[12] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
[13] Martins, A.F.T., Figueiredo, M.A.T., Aguiar, P.M.Q., Smith, N.A., Xing, E.P.: Nonextensive entropic kernels. In: International Conference on Machine Learning, pp. 640–647. ACM (2008)
[14] Ortiz, E.G., Becker, B.C.: Face recognition for web-scale datasets. Comput. Vis. Image Underst. **118**, 153–170 (2014)
[15] Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference (2015)
[16] Ruiz-del Solar, J., Navarrete, P.: Eigenspace-based face recognition: a comparative study of different approaches. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **35**(3), 315–325 (2005)
[17] Rutkowski, L.: Computational Intelligence: Methods and Techniques. Springer, Heidelberg (2010)
[18] Savchenko, A.V.: Statistical recognition of a set of patterns using novel probability neural network. In: Mana, N., Schwenker, F., Trentin, E. (eds.) Proceedings of the International Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR). Lecture Notes in Computer Science, vol. 7477, pp. 93–103. Springer-Verlag Berlin Heidelberg (2012)
[19] Savchenko, A.V.: Probabilistic neural network with homogeneity testing in recognition of discrete patterns set. Neural Netw. **46**, 227–241 (2013)
[20] Savchenko, A.V.: Nonlinear transformation of the distance function in the nearest neighbor image recognition. In: Zhang, Y.J., Tavares, J.M.R.S. (eds.) Proceedings of the International Conference on Computational Modeling of Objects Presented in Images (CompIMAGE), Lecture Notes in Computer Science, vol. 8641, pp. 261–266. Springer International Publishing Switzerland (2014)
[21] Savchenko, A.V.: Fast multi-class recognition of piecewise regular objects based on sequential three-way decisions and granular computing. Knowl.-Based Syst. **91**, 252–262 (2016)
[22] Savchenko, A.V., Belova, N.S.: Statistical testing of segment homogeneity in classification of piecewise-regular objects. Int. J. Appl. Math. Comput. Sci. **25**(4), 915–925 (2015)
[23] Shapiro, L.G., Stockman, G.C.: Computer Vision. Prentice Hall, Upper Saddle River (2001)
[24] Specht, D.F.: Probabilistic neural networks. Neural Netw. **3**(1), 109–118 (1990)
[25] Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic, Burlington/London (2008)
[26] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 511–518 (2001)
[27] Zhou, E., Cao, Z., Yin, Q.: Naive-deep face recognition: touching the limit of LFW benchmark or not? CoRR (2015). abs/1501.04690

Springer

http://www.springer.com/978-3-319-30513-4

Search Techniques in Intelligent Classification Systems
Savchenko, A.V.
2016, XIII, 82 p. 28 illus., 19 illus. in color., Softcover
ISBN: 978-3-319-30513-4