CHAPTER 10

Image Annotation by Moments

Mustapha Oujaoura, Brahim Minaoui and Mohammed Fakir

The rapid growth of the Internet and multimedia information has generated a need for technical indexing and searching of multimedia information, especially in image retrieval. Image searching systems have been developed to allow searching in image databases. However, these systems are still inefficient in terms of semantic image searching by textual query. To perform semantic searching, it is necessary to be able to transform the visual content of the images (colours, textures, shapes) into semantic information. This transformation, called image annotation, assigns a legend or keywords to a digital image. The traditional methods of image retrieval rely heavily on manual image annotation which is very subjective, very expensive and impossible given the size and the phenomenal growth of currently existing image databases. Therefore it is quite natural that the research has emerged in order to find a computing solution to the problem. It is thus that research work has quickly bloomed on the automatic image annotation, aimed at reducing both the cost of annotation and the semantic

Mustapha Oujaoura, Brahim Minaoui, Mohammed Fakir
Laboratory of Information Processing and Telecommunications, Computer Science Department
Faculty of Science and Technology, Sultan Moulay Slimane University
Mghila, PO Box. 523, Béni Mellal, Morocco
e-mail: M.Mustapha.Oujaoura@ieee.org, bra_min@yahoo.fr, fakfad@yahoo.fr

Editor: G.A. Papakostas, *Moments and Moment Invariants - Theory and Applications*

DOI: 10.15579/gcsr.vol1.ch10, GCSR Vol. 1,

gap between semantic concepts and digital low-level features. One of the approaches to deal with image annotation is image classification. From the segmented image, the feature vector is calculated and fed to the classifier in order to choose the appropriate keyword for each region that represents the image content. In this chapter, the use of Hu, Zernike and Legendre moments as feature extraction will be presented.

10.1 Introduction

Due to the large amounts of multimedia data prevalent on the Web, searching digital information archives on the Internet and elsewhere has become a significant part of our daily lives. Amongst the rapidly growing body of information there is a vast number of digital images. The task of automated image retrieval is complicated by the fact that many images do not have adequate textual descriptions. Retrieval of images through analysis of their visual content is therefore an exciting and a worthwhile research challenge.

With regard to the long standing problem of the semantic gap between low-level image features and high-level human knowledge, the image retrieval community has recently shifted its emphasis from low-level features analysis to high-level image semantics extraction. Therefore, image semantics extraction is of great importance to content-based image retrieval because it allows users to freely express what images they want. Semantic content annotation is the basis for semantic content retrieval. The aim of image annotation is to automatically obtain keywords that can be used to represent the content of images.

Automatic object recognition and annotation are essential tasks in these image retrieval systems. Indeed, annotated images play a very important role in information processing; they are useful for image retrieval based on keywords and image content management [7]. For that reason, a lot of research efforts have aimed at annotating objects contained in visual streams. Image content annotation facilitates conceptual image indexing and categorization to assist text-based image search, which can be semantically more meaningful than search in the absence of any text [26, 30].

Manual annotation is not only boring but also not practical in many cases due to the abundance of information. Most images are therefore available without adequate annotation. Automatic image content annotation becomes a recent research interest [26, 32]. It attempts to explore the visual characteristics of images and associate them with image contents and semantics in order to use textual request for image retrieval and searching; automatic image annotation is an effective technology for improving the image retrieval.

The algorithms and systems used for image annotation are commonly divided into those tasks:

- Image Segmentation
- Feature extraction
- Image Classification and Annotation

After segmentation of the query image into several regions that are supposed to represent the object contained in that image, the features vector can be extracted carefully in order to keep the maximum information while reducing the size of features vector. This features vector is fed to the input of the trained classifier in order to choose the appropriate keyword that can be used for image content indexing and retrieval. Those tasks will be presented and discussed later in the following section.

10.2 Image Segmentation

The features' vector, which is extracted from the entire image, loses local information. So, it is necessary to segment an image into regions or objects of interest and make use of local characteristics. The image segmentation is the process of partitioning a digital image into multiple segments. It is very important in many applications for any image processing, and it still remains a challenge for scientists and researchers. So far, the efforts and attempts are still being made to improve the segmentation techniques. With the improvement of computer processing capabilities, there are several possible segmentation techniques of an image: threshold, region growing, active contours, level sets, etc.. [7]. After dividing the original image into several distinct regions that correspond to objects in a scene, the feature vector can be extracted from each region and can be considered as a representation of an object in the entire image.

10.3 Extraction and Computation of Moments

When the input data to an algorithm is too large to be processed, it will be transformed into a reduced representation of features' set. Transforming the input data into the set of features is called features' extraction. The extraction task transforms rich contents of images into various features. In order to perform this task using this reduced representation instead of the full size input, we need to extract carefully these features [5]. It enhances not only the retrieval and annotation's accuracy, but also the annotation's speed. So a large image database can be organized according to the classification rule and therefore, the searching can be performed [31]. The moments are widely used in many applications for feature's extraction due to their invariance to scale, rotation and reflection change [19, 22, 20, 21]. The use of moments for image analysis and pattern recognition was inspired by Hu [11]. The most common used moments are:

- Hu moments
- Legendre moments
- Zernike moments

The features vector of each one of these moments can be computed and extracted for the 3 planes of color images. So, in addition to the shape information described by moments, the color information contained in the image is also considered and contributes to the improvement of image annotation accuracy.

10.3.1 Hu Moments

Hu moments are widely used in the image processing and pattern recognition. They are derived and calculated from geometric moments. The two-dimensional geometric moments of order $(p+q)$ of an image, that is represented by a real valued measurable function $f(x,y)$ in the interval range $[a_1,a_2]\times[b_1,b_2]$, are defined as:

$$M_{pq}=\int_{a_1}^{a_2}\int_{b_1}^{b_2}x^p y^q f(x,y)\,dxdy, \tag{10.1}$$

where $p,q=0,1,2,...,\infty$. The monomial product $x^p y^q$ is the basis function for this moment definition. A set of n moments consists of all M_{pq} for $p+q\leq n$. The zero order moment, M_{00}, of the function $f(x,y)$

$$M_{00}=\int_{a_1}^{a_2}\int_{b_1}^{b_2}f(x,y)\,dxdy, \tag{10.2}$$

represents the total of mass of the given image. The two first order moments

$$M_{10}=\int_{a_1}^{a_2}\int_{b_1}^{b_2}xf(x,y)\,dxdy, \tag{10.3}$$

$$M_{01}=\int_{a_1}^{a_2}\int_{b_1}^{b_2}yf(x,y)\,dxdy, \tag{10.4}$$

represent the centre of mass of the given image. In terms of moment values, the coordinates of the centre of mass are

$$\bar{x}=\frac{M_{10}}{M_{00}},\ \bar{y}=\frac{M_{01}}{M_{00}}. \tag{10.5}$$

The central moments of an image, that is represented by $f(x,y)$, are defined as:

$$a_{pq}=\int_{a_1}^{a_2}\int_{b_1}^{b_2}(x-\bar{x})^p(y-\bar{y})^q f(x,y)\,dxdy, \tag{10.6}$$

where $\bar{x}$ and $\bar{y}$ are defined in Eq.(10.5).

The central moments a_{pq} defined in Eq.(10.6) are invariant under the translation of coordinates. They can be normalized to preserve the invariance by scaling. For $p+q=2,3,...$The Normalized central moments of an image are given by:

$$\mu_{pq}=\frac{a_{pq}}{a_{00}^{\gamma}},\ with\,\gamma=\frac{p+q}{2}+1. \tag{10.7}$$

Based on the theory of algebraic invariance, Hu [11] derived relative and absolute combinations of moments that are invariant to scale, position and orientation. Hu defined the following seven functions, computed from the normalized central moments through the order three, that are invariant to scale, translation and rotation changes:

$$\phi_1 = \mu_{20} + \mu_{02} \tag{10.8}$$

$$\phi_2 = (\mu_{20} + \mu_{02})^2 + 4\mu_{11}^2 \tag{10.9}$$

$$\phi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \tag{10.10}$$

$$\phi_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \tag{10.11}$$

$$\begin{aligned}\phi_5 = \quad & (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})\left[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2\right] + \\ & (3\mu_{12} - \mu_{03})(\mu_{21} + \mu_{03})\left[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2\right]\end{aligned} \tag{10.12}$$

$$\begin{aligned}\phi_6 = \quad & (\mu_{20} - \mu_{02})\left[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2\right] + \\ & 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})\end{aligned} \tag{10.13}$$

$$\begin{aligned}\phi_7 = \quad & (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})\left[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2\right] - \\ & (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})\left[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2\right]\end{aligned} \tag{10.14}$$

The function ϕ_1 through ϕ_6 are invariant to rotation and reflection while ϕ_7 changes sign under reflection.

10.3.2 Legendre Moments

Legendre moments, were first introduced by Teague [28]. Legendre moments are orthogonal moments. They were used in several patterns' recognition [4].

The $(p+q)$-th order of Legendre moment, of an image with intensity function $f(x,y)$ is defined on the square $[-1,1] \times [-1,1]$, by:

$$L_{pq} = \lambda_{pq} \int_{-1}^{+1} \int P_p(x) P_q(y) f(x,y)\, dxdy, \tag{10.15}$$

where $\lambda_{pq} = \frac{(2p+1)(2q+1)}{4}$, $p,q = 0,1,2,...,\infty$, and $P_p(x)$ is the p-th order Legendre polynomial defined by:

$$P_p(x) = \sum_{k=0}^{p} \alpha_{pk} x^k = \frac{1}{2^p p!} \frac{d^p}{dx^p} \left(x^2 - 1\right)^p, \tag{10.16}$$

or

$$P_p(x) = \sum_{k=0}^{p} \left\{ \frac{(-1)^{\frac{p-k}{2}} x^k (p+k)!}{2^p k! \left(\frac{p-k}{2}\right)! \left(\frac{p+k}{2}\right)!} \right\}_{p-k=even}. \tag{10.17}$$

The Legendre polynomials have the generating function:

$$\frac{1}{\sqrt{1-2rx+r^2}} = \sum_{p=0}^{\infty} r^p P_p(x), \; r \prec 1. \tag{10.18}$$

By deriving the two parts of the generating function above, the recurrent formula of the Legendre polynomials can be acquired straightforwardly:

$$\frac{d}{dr}\left(\frac{1}{\sqrt{1-2rx+r^2}}\right) = \frac{d}{dr}\left(\sum_{p=0}^{\infty} r^p P_p(x)\right) \Leftrightarrow$$

$$\frac{1}{\sqrt{1-2rx+r^2}} \times \frac{x-r}{1-2rx+r^2} = \sum_{p=0}^{\infty} p r^{p-1} P_p(x).$$

Then we have:

$$(x-r)\sum_{p=0}^{\infty} r^p P_p(x) = \left(1-2rx+r^2\right)\sum_{p=0}^{\infty} p r^{p-1} P_p(x).$$

And, the recurrent formula of the Legendre polynomials is:

$$\begin{cases} P_{p+1}(x) = \frac{2p+1}{p+1} x P_p(x) - \frac{p}{p+1} P_{p-1}(x) \\ P_1(x) = x, \; P_0(x) = 1 \end{cases} \tag{10.19}$$

The Legendre polynomials are a complete orthogonal basis set on the interval $[-1, 1]$:

$$\int_{-1}^{+1} P_p(x) P_q(x) dx = \frac{2}{2p+1}\delta_{pq}, \tag{10.20}$$

where

$$\delta_{pq} = \begin{cases} 1 & if \; p = q \\ 0 & if \; p \neq q \end{cases},$$

is the Kronecker symbol.

The orthogonal property of Legendre polynomials implies no redundancy or overlapping of information between the moments with different orders. This property enables the contribution of each moment to be unique and independent from the information in an image [28].

To compute Legendre moments from a digital image, the integrals in Eq.(10.15) are replaced by summations and the coordinates of the image must be normalized into $[-1, 1]$. Therefore, the numerical approximate form of Legendre moments, for a discrete image of $M \times N$ pixels with intensity's function $f(x, y)$, is [10]:

$$L_{pq} = \lambda_{pq} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} P_p(x_i) P_q(y_j) f(x_i, y_j), \tag{10.21}$$

where $\lambda_{pq} = \frac{(2p+1)(2q+1)}{M \times N}$, x_i and y_j denote the normalized pixel coordinates in the range of $[-1, 1]$, which are given by:

$$x_i = \frac{2i - (M-1)}{M-1},\ y_j = \frac{2j - (N-1)}{N-1}. \tag{10.22}$$

The formula defined in Eq.(10.21) is obtained by replacing the integrals in Eq.(10.15) by summations and by normalizing the pixel coordinates of the image into the range of $[-1, 1]$ using Eq.(10.22).

Figure 10.1 shows the pseudo code for computing Legendre moments of order $(p+q)$ by equation defined in Eq.(10.21) and by using direct method for calculating Legendre polynomials. In this work the recurrent formula is used for calculating Legendre polynomials in order to increase computation speed. Other fast and accurate computation method of Legendre moments are presented in [23].

10.3.3 Zernike Moments

Zernike moments, as a type of moment function, are the mapping of an image onto a set of complex Zernike polynomials. As these Zernike polynomials are orthogonal to each other, Zernike moments can represent the properties of an image with no redundancy or overlapping of information between the moments [12]. Due to these characteristics, Zernike moments have been used as features set in many applications [29].

The computation of Zernike moments from an input image consists of three steps: computation of radial polynomials, computation of Zernike polynomials, and computation of Zernike moments by projecting the image onto the Zernike polynomials.

The procedure of obtaining Zernike moments from an input image begins with the computation of radial polynomials. The real-valued radial polynomial is defined as:

$$R_{p,q}(r) = \sum_{s=0}^{(p-|q|)/2} \frac{(-1)^s (p-s)! r^{p-2s}}{s! \left(\frac{p+|q|}{2} - s\right)! \left(\frac{p-|q|}{2} - s\right)!}, \tag{10.23}$$

with $R_{p,q}(r) = R_{p,-q}(r)$.

In Eq.(10.23), p and q are generally called respectively order and repetition. The order p is a non-negative integer, and the repetition q is an integer satisfying $p - |q| = even$ and $|q| \leq p$. The radial polynomials satisfy the orthogonal properties for the same repetition q,

Function LegendrePolynomial (x, p)
px=0 ;
for k=0 **to** p
if mod (p-k,2)=0

$$c = \frac{(-1)^{\frac{p-k}{2}} x^k (p+k)!}{2^p k! \left(\frac{p-k}{2}\right)! \left(\frac{p+k}{2}\right)!};$$

px=px+c;
end if
end for
return px ;
Function LegendreMoments (p, q)
L=0;
for i=0 **to** (M-1)
for j=0 **to** (N-1)

$$x_i = \frac{2i-(M-1)}{M-1};\ y_j = \frac{2j-(N-1)}{N-1};$$

px= LegendrePolynomial (x_i, p) ;
py= LegendrePolynomial (y_i, q) ;

$$L = L + f(x_i, y_j) * px * py;$$

end for
end for
return $\frac{L\,(2p+1)(2q+1)}{M \times N}$;

Figure 10.1: Pseudo code for computing Legendre moments.

$$\int_0^{2\pi}\int_0^1 R_{p,q}(r,\theta)\,R_{p',q}(r,\theta)\,r dr d\theta = \frac{\delta_{pp'}}{2(p+1)}, \tag{10.24}$$

where $\delta_{pp'}$ is the Kronecker symbol defined in the previous section.

Using the radial polynomial, complex-valued 2-D Zernike polynomials, which are defined within a unit circle, are formed by:

$$V_{pq}(x,y) = V_{pq}(r\sin\theta, r\cos\theta) = R_{p,q}(r)\,e^{jq\theta}, \tag{10.25}$$

where, $j=\sqrt{-1}$, $|r|\le 1$ is the length of the vector from the origin to the pixel at (x,y), and θ is the angle between vector r and the x axis.

The Zernike polynomials are a complete set of complex-valued functions orthogonal on the unit disk $x^2+y^2\le 1$.

$$\iint_{x^2+y^2\le 1} [V_{nm}(x,y)]^* V_{pq}(x,y)\,dxdy = \frac{\pi\delta_{mp}\delta_{nq}}{m+1}, \tag{10.26}$$

or, in polar coordinates:

$$\int_0^{2\pi}\int_0^1 [V_{nm}(r,\theta)]^* V_{pq}(r,\theta)\,r dr d\theta = \frac{\pi\delta_{mp}\delta_{nq}}{m+1}, \tag{10.27}$$

where the asterisk (*) denotes the conjugated complex.

The complex Zernike moments of order p with repetition q for an image function $f(x,y)$ are finally defined as:

$$Z_{pq} = \frac{p+1}{\pi}\iint_{x^2+y^2\le 1} [V_{pq}(x,y)]^* f(x,y)\,dxdy, \tag{10.28}$$

or, in polar coordinates:

$$Z_{pq} = \frac{p+1}{\pi}\int_0^{2\pi}\int_0^1 [V_{pq}(r,\theta)]^* f(r,\theta)\,r dr d\theta, \tag{10.29}$$

According to this definition, the procedure to compute Zernike moments can be seen as an inner product between the image's function and the Zernike polynomials.

To compute Zernike moments from a digital image, the integrals in Eq.(10.28) and in Eq.(10.29) are replaced by summations in addition to the coordinates of the image which must be normalized into $[0,1]$ by a mapping transform. The two commonly used cases of the transformations are shown in Fig.(10.2b) the image is over a unit circle and Fig.(10.2c) the image is inside a unit circle. Based on Fig.(10.2b), the pixels, which are located on the outside of the circle, are not involved in the computation of the Zernike moments. Accordingly, Zernike moments, which are computed by the mapping transformation, do not describe the properties of the outside of the unit circle in the image. This can be considered as a default while calculating Zernike moments.

The discrete form of the Zernike moments of an image size $M \times N$ is expressed as follows:

$$\begin{aligned} Z_{pq} &= \frac{p+1}{\lambda} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[V_{pq}\left(x,y\right)\right]^{*} f\left(x,y\right) \\ &= \frac{p+1}{\lambda} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} R_{pq}\left(r_{xy}\right) e^{-jq\theta_{xy}} f\left(x,y\right) \end{aligned} , \tag{10.30}$$

where $0 \leq r_{xy} \leq 1$ and λ is a normalization factor.

In the discrete implementation of Zernike moments, the normalization factor λ must be the number of pixels located in the unit circle by the mapping transformation and corresponds to the area of a unit circle π in the continuous domain. The transformed θ_{xy} phase and the distance r_{xy} at the pixel of coordinates (x, y) are given by:

For Fig.(10.2b):

$$\theta_{xy} = \tan^{-1}\left(\frac{\left(2y-(N-1)\right)/\left(N-1\right)}{\left(2x-(M-1)\right)/\left(M-1\right)}\right) , \tag{10.31}$$

$$r_{xy} = \sqrt{\left(\frac{2x-(M-1)}{M-1}\right)^{2} + \left(\frac{2y-(N-1)}{N-1}\right)^{2}} . \tag{10.32}$$

For Fig.(10.2c):

$$\theta_{xy} = \tan^{-1}\left(\frac{2y-(N-1)}{2x-(M-1)}\right) \tag{10.33}$$

$$r_{xy} = \sqrt{\frac{\left(2x-(M-1)\right)^{2} + \left(2y-(N-1)\right)^{2}}{\left(M-1\right)^{2} + \left(N-1\right)^{2}}} . \tag{10.34}$$

Figure 10.3 shows the pseudo code to compute Zernike moments of order p with repetition q by Eq.(10.31) and by using direct method for radial polynomials.

Most of the computation time of Zernike moments is because of computation of radial polynomials. Therefore, researchers have proposed faster methods that reduce the factorial terms by utilizing the recurrence relations on the radial polynomials. Prata [24] proposed a recurrence's relation that uses radial polynomials of lower order than p as follows:

$$R_{pq}\left(r\right) = \frac{2rp}{p+q} R_{(p-1)(q-1)}\left(r\right) - \frac{p-q}{p+q} R_{(p-2)q}\left(r\right) . \tag{10.35}$$

It is quite evident from the precedent equation that we can't compute all cases of p and q while computing the radial polynomials. It is not possible to use Prata's equation in cases where $q = 0$ and $p = q$. Those cases can be obtained by other methods. The direct method can be used in cases where $q = 0$, whereas the equation $R_{pp}\left(r\right) = r^{p}$ is used for $p = q$. The usage of direct method to compute radial polynomials in the case of $q = 0$ will considerably increase the computation time, especially when p is large.

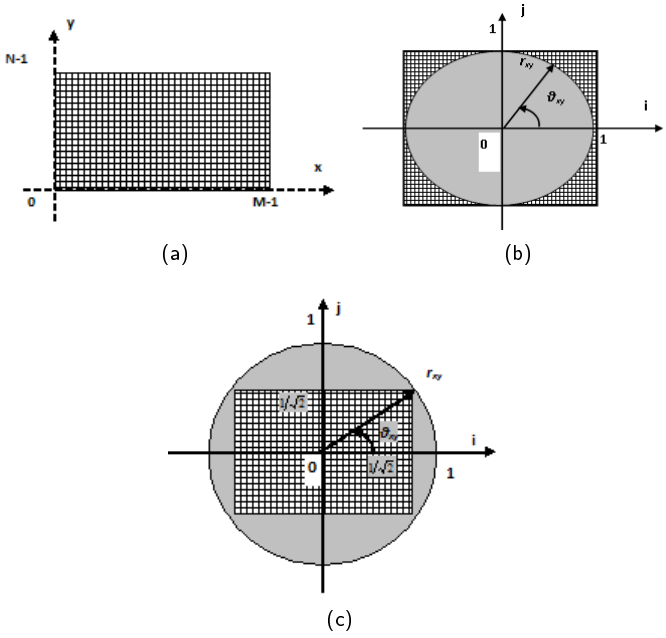


Figure 10.2: (a) Image $M \times N$, (b) mapping of image over a unit circle and (c) mapping of image inside a unit circle.

Function RadialPolinomial (r, p, q)

radial=0 ;

for s=0 **to** (p-q)/2

$$c = \frac{(-1)^s (p-s)!}{s!\left(\frac{p+|q|}{2}-s\right)!\left(\frac{p-|q|}{2}-s\right)!};$$

radial=radial + c * r^{p-2s};

end for

return radial ;

Function ZernikeMoments (p, q)

Zr=0; Zi=0;

Count=0;

for x=0 **to** (M-1)

for y=0 **to** (N-1)

$$r = \sqrt{\left(\frac{2x-(M-1)}{M-1}\right)^2 + \left(\frac{2y-(N-1)}{N-1}\right)^2};$$

$$\theta = \tan^{-1}\left(\frac{(2y-(N-1))/(N-1)}{(2x-(M-1))/(M-1)}\right);$$

If $r \le 1$

Radial= RadialPolinomial (r, p, q) ;

$Zr = Zr + f(x,y) * radial * \cos(q * \theta)$;

$Zi = Zi + f(x,y) * radial * \sin(q * \theta)$;

Count=Count+1;

end if

end for

end for

return $\frac{(p+1)(Zr + i * Zi)}{Count}$;

Figure 10.3: Pseudo code for computing Zernike moments.

Kintner [14] proposed another recurrence's relation that uses polynomials of a varying low-order p with a fixed repetition q to compute the radial polynomials as shown below:

$$R_{pq}(r) = \frac{\left(K_2 r^2 + K_3\right) R_{(p-2)q}(r) + K_4 R_{(p-4)q}(r)}{K_1} \tag{10.36}$$

The coefficients K_1, K_2, K_3 and K_4 are given by

$$\begin{aligned}
K_1 &= \frac{(p+q)(p-q)(p-2)}{2} \\
K_2 &= 2p(p-1)(p-2) \\
K_3 &= -q^2(p-1) - p(p-1)(p-2) \\
K_4 &= \frac{-p(p+q-2)(p-q-2)}{2}
\end{aligned}$$

Like the equation in Eq.(10.36), Kintner's method cannot be applied in cases where $p = q$ and $p - q = 2$. For these two cases, in the normal approach, it is better to use the direct method, although it takes too much time to compute. The following two relations are used to avoid the involvement of direct method. For $p = q$ the equation $R_{pp}(r) = r^p$ is used, and for $p - q = 2$ the recurrent relation below is used:

$$R_{pq}(r) = pR_{pp}(r) - (p-1) R_{qq}(r) \tag{10.37}$$

This improved version of Kintner's method is denoted as modified Kintner's method.

Recently, Chong [3] presented the *q-recursive* method, which uses a relation of the radial polynomials of fixed order p and varying repetition of q. The relation of the radial polynomial is defined as:

$$R_{pq}(r) = H_1 R_{p(q+4)}(r) + \left(H_2 + \frac{H_3}{r^2}\right) R_{p(q+2)}(r) \tag{10.38}$$

where

$$\begin{aligned}
H_1 &= \frac{(q+4)(q+3)}{2} - (q+4) H_2 + \frac{H_3 (p+q+6)(p-q-4)}{8} \\
H_2 &= \frac{H_3 (p+q+4)(p-q-2)}{4(q+3)} + (q+2) \\
H_3 &= \frac{4(q+2)(q+1)}{(p+q+2)(p-q)}
\end{aligned}$$

As the order p is fixed in Eq.(10.38), the individual order of Zernike moments can be calculated independently without referring to higher or lower order moments. All these precedent methods focus only on the computation of Zernike radial polynomials

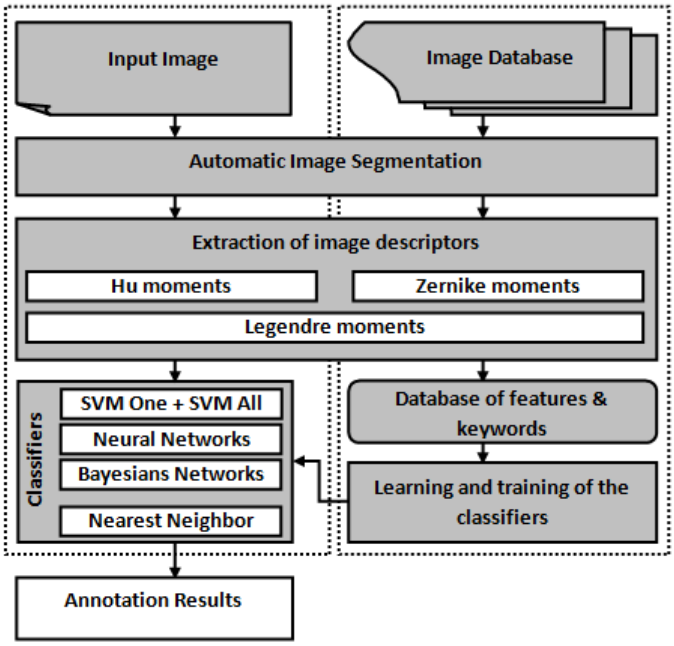


Figure 10.4: Block diagram of the proposed annotation system.

and have some limitations if just a single Zernike moment is required because they use recurrence's relations.

From the experiments in [3], the combined use of both the *q-recursive* method and modified Kintner's method takes the shortest time to compute a full set of Zernike moments followed by Kintner's method. Chong's method is much faster than other methods especially in computing Zernike moments with a fixed order. Therefore Chong's method is more effective in cases, where only selected orders of Zernike moments are used as feature vectors. The Zernike moments can be obtained fastly using the hybrid method. The *q-recursive* method and modified Kintner's method can be combined and used for fast calculation of the Zernike radial polynomials.

10.4 Image Classification and Annotation

The image annotation can be approached by the model or the classifier generated and trained to bridge the gap between low-level feature vectors and high-level concepts; a function is learned which can directly correspond the low-level feature sets to high-level conceptual classes. There are several types of classifier that are used singly for classification. Each classifier is found suitable to classify a particular kind of feature vectors depending on their characteristics. The Neural Networks, Support Vector Machines (SVM) and K-Nearest Neighbour classifiers are used in this chapter. The block diagram of the image annotation system adopted in this work is shown in Fig.(10.4).

The system contains two parts: the first one is reserved to a reference database of

image already annotated by experts (offline and manual annotation). This database is used for modelling and training the classifiers (Neural Network, Bayesian Network and Multiclass SVM). The second one can be considered as the direct image annotation (online annotation), which is the subject of this chapter. To achieve this goal, firstly, the query image is segmented into regions that represent objects in the image, secondly, the features vector of each region is computed and extracted from the image, and those features are finally fed into input of the classifiers (Neural Network, Bayesian Network, Multiclass SVM and Nearest Neighbour Classifiers). These classifiers decide and choose the appropriate keywords for annotation tasks of the object.

10.4.1 Nearest Neighbor Classifier

The nearest neighbour classifier is used to compare the feature vector of the input image and the feature vectors stored in the database. It is obtained by finding the distance between the prototype image and the database. The class is found by measuring the distance between a feature vector of input image and feature vectors of images in reference database. The Euclidean distance measurement is used in this paper, but other distance measurements can also be used [1].

Let $X_1, X_2, ..., X_k$ be the k class features vectors in the database and X_q the feature vector of the query image. The feature vector with the minimum distance is found to be the closest matching vector. It is given by:

$$d\left(X_q, X_j\right) = \underset{j\in\{1,2,\ldots,k\}}{min}\left\{\sqrt{\sum_i \left(x_q\left(i\right) - x_j\left(i\right)\right)^2}\right\} \tag{10.39}$$

The nearest neighbour classifier does not need any training phase. But, if the database is very large, it takes a considerable time to calculate all the distances between the query image and database classes.

10.4.2 Neural Networks

Neural networks (or artificial neural networks) learn by experience, generalize from previous experiences to new ones, and can make decisions [2, 27]. A multilayer network consists of an input layer including a set of input nodes, one or more hidden layers of nodes, and an output layer of nodes. Figure 10.5 shows an example of a three layer network used in this paper, having an input layer formed by M nodes, one hidden layer formed by L nodes, and an output layer formed by N nodes. This neural network is trained to classify inputs according to target classes. The training input data are loaded from the reference database while the target data should consist of vectors of all zero values except for a one element, where its index is the class they are to represent. The transfer function used in this tree layer neural network is hyperbolic tangent sigmoid transfer function defined by:

$$f\left(x\right) = 2/\left(1 + e^{-2x}\right) - 1 \tag{10.40}$$

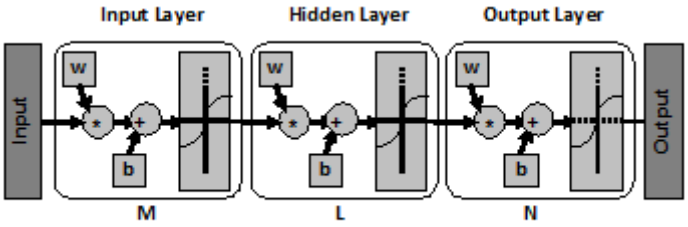


Figure 10.5: The three layer neural network.

According to authors in [15], the number of neurons in the hidden layer is approximately equal to:

$$L = E\left(1 + \sqrt{M\,(N+2)}\right) \tag{10.41}$$

where

- $E\,(x)$ denotes the integer part of x
- M and N are respectively the number of neurons in the input and output layers

10.4.3 Support Vector Machines (SVM)

Support vector machines (SVMs) were originally designed for binary classification. SVM is a classification method which is based on finding a hyper-plan that separates data sets into two classes. Several methods have been proposed to construct a multiclass SVM classifier by combining one-against-one or one-against-all binary classifiers as shown in Fig.(10.6). The data sets can be linearly separable or nonlinearly separable. The nonlinearly separable cases require the use of kernel function in order to obtain linearly separable data sets [8, 25]. The one-against-one and the one-against-all binary classifier can be used. These classifiers are based on the Gaussian kernel function defined by:

$$K\,(x,y) = exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{10.42}$$

where $\sigma = 1$. Many other kernel functions can be used for each binary classifier.

10.4.3.1 One-against-all binary classifier

The one-against-all binary classifier contains N binary classifier, where N is the number of class in data sets. The i^{th} binary SVM is trained with all of the data examples in the i^{th} class with positive labels, and all other data examples with negative labels. To construct a one-against-all multiclass SVM model from binary classifier, the classes are divided into two groups: the first group is formed by one class, and the second group

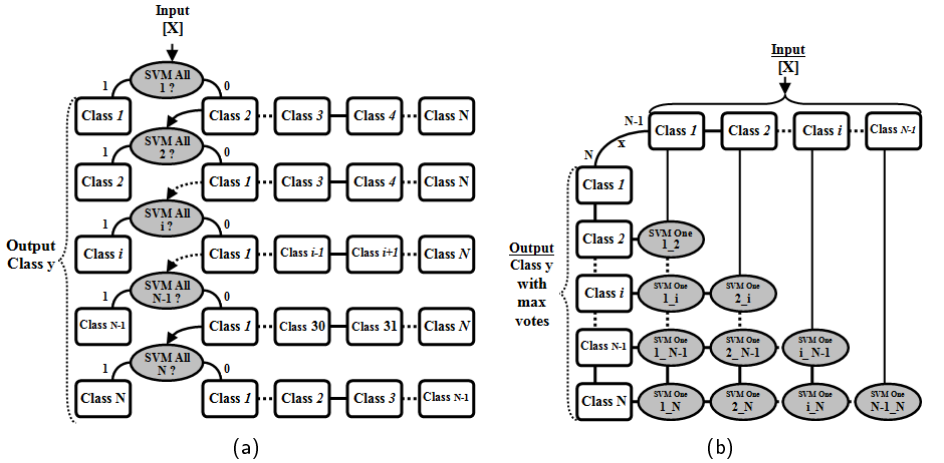


Figure 10.6: Structure of the multiclass SVM classifier, a) one-against-all , b) one against-one.

is all the other classes. The obtained SVM binary classifier is trained to decide if the class is from the first group or it belongs to the second group of classes. This process is repeated for the second group that contains more than two classes until having only one class for each group. The process must stop there. So, by following this way, multiclass SVM is transformed to a multiple SVM binary classifier. Each SVM binary classifier is trained using a matrix of training data, where each row corresponds to features extracted as an observation from a class. After the training phase, the multiclass SVM model is able to decide the correct class for an input features vector. To classify an object, its input features vector is presented iteratively to the i^{th} against all binary classifier from the first to the N^{th} classifier while the result is negative. When the i^{th} binary classifier gives a positive result, the process stops. This means that the object belongs to the i^{th} class.

10.4.3.2 One-against-one binary classifier

Another major method is called the one-against-one binary classifier. From N class in data sets, this method constructs $N(N-1)/2$ binary classifiers where each one is trained on data from two classes. To design and extend SVM binary classifier into a one-against-one multiclass SVM, two groups of data examples are constructed from two classes. The obtained SVM binary classifier is trained to decide if the class is from the first class or it belongs to the second class. This process is repeated for another couple of classes until finishing all the possible couples of the classes from data sets. So, by following this way, multiclass SVM is transformed to a multiple $N(N-1)/2$ SVM binary classifier. Each SVM binary classifier is trained using a matrix of training data, where each row corresponds to the features extracted as an observation from a class. When classifying an object with an input features vector, each binary classifier from the multiclass SVM one-against-one model decides and votes for only one class.

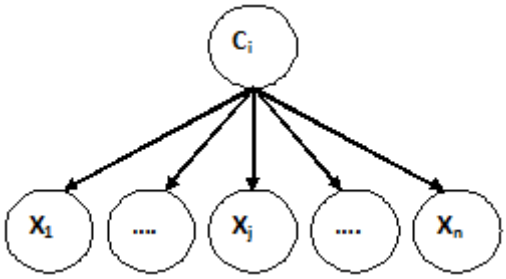


Figure 10.7: Naive Bayes classifier structure.

The class with the majority votes is the correct class which the object belongs to.

10.4.4 Bayesian Network

The construction of a Bayesian network consists of finding a structure or a graph and estimating its parameters by machine learning. In the case of the classification, the Bayesian network can have a class node C_i and many attribute nodes X_j. The naive Bayes classifier is used in this paper due to its robustness and simplicity. Figure 10.7 illustrates its graphical structure.

To estimate the Bayesian network parameters and probabilities, Gaussian distributions are generally used. The conditional distribution of a node relative to its parent is a Gaussian distribution whose mean is a linear combination of the parent's value and whose variance is independent of the parent's value [13] :

$$P\left(X_i|Pa\left(X_i\right)\right) = \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left\{\frac{-1}{2\sigma_i^2}\left(x_i - \left(\mu_i + \sum_{j=1}^{n_i}\frac{\sigma_{ij}}{\sigma_j^2}\left(x_j - \mu_j\right)\right)\right)^2\right\} \tag{10.43}$$

where

- $Pa\left(X_i\right)$are the parents of X_i
- μ_i, μ_j, σ_i and σ_j are the means and variances of the attributes X_i and X_j respectively without considering their parents
- n_i is the number of parents
- σ_{ij} is the regression matrix of weights.

After the learning of the parameters and structure of a Bayesian network, the Bayesian inference is used to calculate the probability of any variable in a probabilistic model from the observation of one or more other variables. So, the chosen class C_i is the one that maximizes these probabilities [16, 18]:

$$P(C_i|X) = \begin{cases} P(C_i) \prod_{j=1}^{n} P(X_j|Pa(X_j), C_i) & if\ X_j\ has\ parents \\ P(C_i) \prod_{j=1}^{n} P(X_j|C_i) & else \end{cases} \quad (10.44)$$

For the naive Bayes classifier, the absence of parents and the variables independence assumption are used to write the posterior probability of each class as given in the following equation [17]:

$$P(C_i|X) = P(C_i) \prod_{j=1}^{n} P(X_j|C_i) \quad (10.45)$$

Therefore, the decision rule d of an attribute X is given by:

$$\begin{aligned} d(X) &= \underset{C_i}{argmax} P(C_i|X) \\ &= \underset{C_i}{argmax} P(X|C_i) P(C_i) \\ &= \underset{C_i}{argmax} P(C_i) \prod_{j=1}^{n} P(X_j|C_i) \end{aligned} \quad (10.46)$$

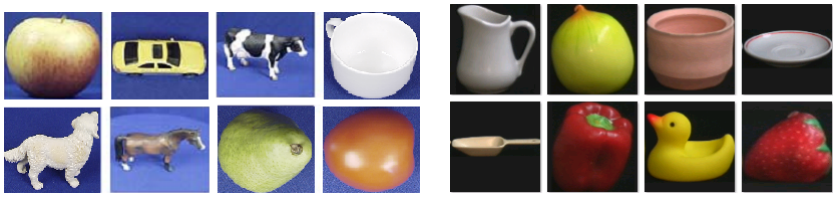
The class with maximum probability leads to the suitable character for the input image.

10.5 Experiments and Results

In our experiments, for each region that represent an object from one channel of the query image, the number of input features extracted using Hu invariants features extraction method is 7 (hu1, hu2, hu3, hu4, hu5, hu6, hu7) while the number of input features extracted using the order 4 of Zernike moments is 9 (Z_{00}, Z_{11}, Z_{20}, Z_{22}, Z_{31}, Z_{33}, Z_{40}, Z_{42}, Z_{44}) and the number of input features extracted using the order 3 of Legendre moments is 10 (L_{00}, L_{01}, L_{02}, L_{03}, L_{10}, L_{11}, L_{12}, L_{20}, L_{21}, L_{30}). So, in the case of color image, the resulted features vector is composed from 21 elements for Hu moments, 27 elements for Zernike moments and 30 elements for Legendre moments. These inputs are presented and fed to the classifier; which is the multiclass SVM, the neural network or the nearest neighbour classifier for testing to do matching with the feature values in the reference database.

Figure 10.8 shows some examples of image objects from ETH-80 [9] image database and COIL-100 [6] image database used in our experiments. The experiments were performed based on different classes of objects.

The accuracy of image annotation is evaluated by the precision rate which is the number of correct results divided by the number of all returned results. All the experiments are conducted using two databases: ETH-80 database containing a set of 8 different object images [9] and COIL-100 database which contains color images of 100 objects with 72 different angle views [6]. The proposed image annotation system has



(a) (b)

Figure 10.8: Some examples of objects from: a) ETH-80 image database, b) COIL-100 image database.

Table 10.1: Single classifier and descriptor mean annotation rates and total execution time for image from ETH-80 and COIL-100 databases.

	Database	Descriptor	Classifier				
			K-NN	SVM-One	SVM-All	Neural Network	Bayesian Network
Annotation Rates	ETH-80	Hu	59.80%	53.75%	58.13%	61.53%	70.00%
		Zernike	68.41%	62.50%	65.00%	69.70%	76.32%
		Legendre	78.55%	73.75%	75.00%	79.91%	82.50%
	COIL-100	Hu	50.00%	43.75%	43.75%	61.25%	65.00%
		Zernike	71.25%	60.00%	65.00%	73.75%	75.50%
		Legendre	73.50%	70.00%	75.00%	77.50%	80.00%
Execution Time in seconds	ETH-80	Hu	121.96	259.08	82.89	146.72	205.28
		Zernike	3213.23	3820.65	3405.48	3338.17	3482.07
		Legendre	10110.55	10490.77	12743.99	13245.28	11408.25
	COIL-100	Hu	84.91	158.42	74.35	144.3	108.13
		Zernike	3088.51	3216.42	2641.45	2674.8	3101.21
		Legendre	11976.21	12127.88	12225.79	12866.25	9802.82

been implemented and tested on a core 2 Duo personnel computer using MATLAB software. The annotation results for each classifier and each type of moments for image from ETH-80 and COIL-100 databases are presented in Table 10.1.

We observe from the above results that the Legendre moments with Bayesian networks classifier give good results.

Figure 10.9 presents the annotation rates for each classifier and each type of moments for image from ETH-80 and COIL-100 databases, while Fig.(10.10) presents the approximate execution times for each classifier and each type of moments for image from ETH-80 and COIL-100 databases

The confusion matrix given in Fig.(10.11) shows improperly annotated objects (indicated by the red color) in the case of using Legendre moments as descriptors and Bayesian networks as classifier for images from ETH-80 database.

Figure 10.12 shows the confusion matrix in the case of using Legendre moments as a descriptor and the Bayesian network as classifier for images from the COIL-100 image database.

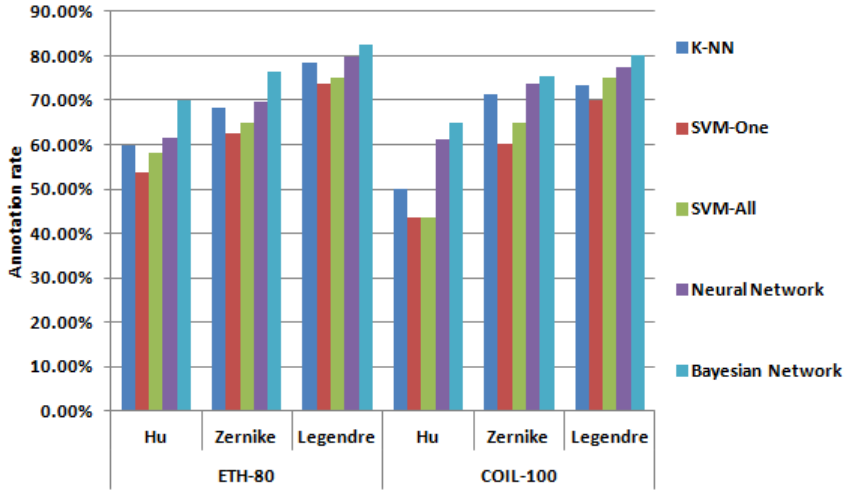


Figure 10.9: Mean annotation rates for each classifier and each type of moments for image from ETH-80 and COIL-100 databases.

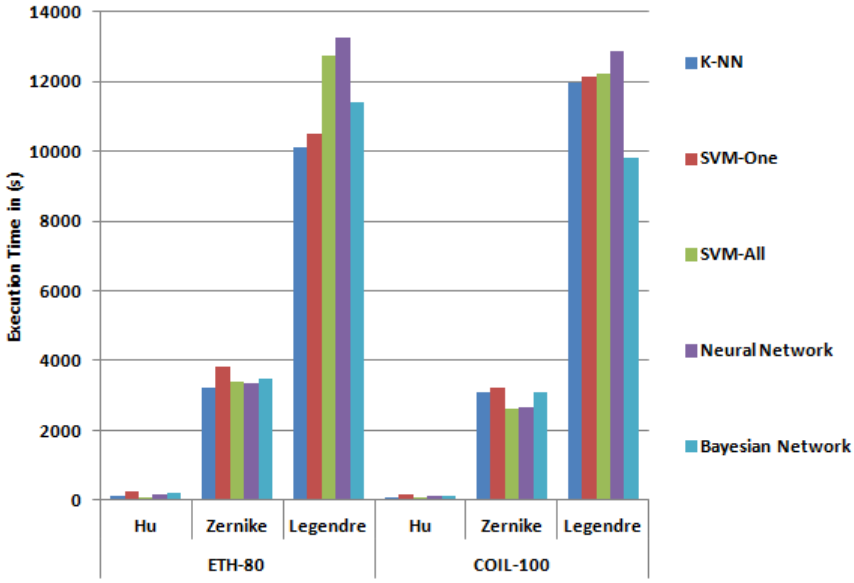


Figure 10.10: Total execution times for each classifier and each type of moments for image from ETH-80 and COIL-100 databases.

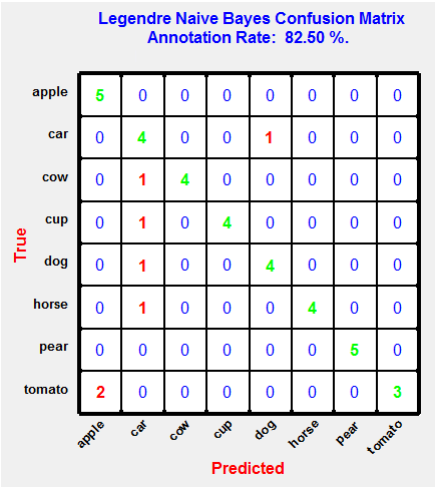


Figure 10.11: Confusion matrix using Legendre moments and Naïve Bayes network for image from ETH-80 database.

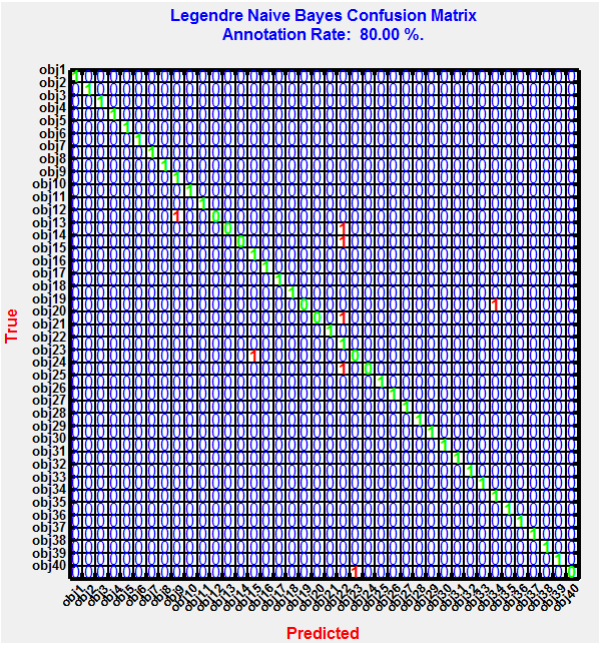
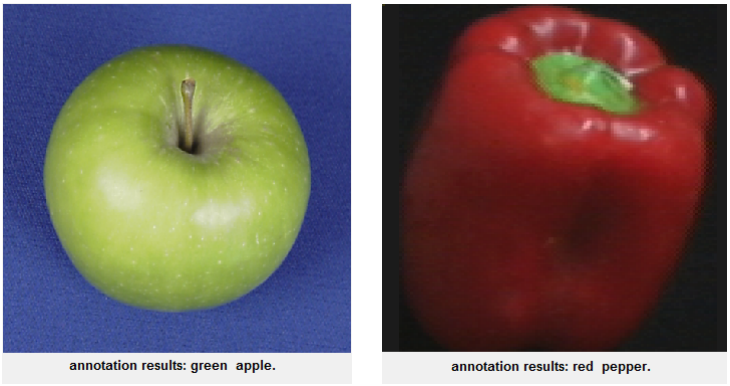


Figure 10.12: Confusion matrix using Legendre moments and Naïve Bayes network for image from COIL-100 database.



(a) (b)

Figure 10.13: Example of image annotation results using a set of 8 images for training the classifiers from : a) ETH-80 database and b) COIL-100 database.

For each classifier and for both images databases ETH-80 and COIL-100 containing objects in general well-defined by shapes, Legendre descriptors allow to get the best image annotation rate.

Figure 10.13 presents an example of annotation results obtained by using the presented system while the Graphical User Interface is illustrated in Fig.(10.14).

The image annotation results are also affected by the accuracy of the image segmentation method; in most cases, it is very difficult to have an automatic ideal segmentation. This problem decreases the annotation rates. Therefore, any annotation attempt must consider image segmentation as an important step, not only for automatic image annotation system, but also for the other systems which require its use. The Legendre moments and Zernike moments are very expensive regarding the processing and computation time, so any use of them in real time for an online image annotation system will be difficult and impracticable.

10.6 Conclusion

In this chapter, we have discussed image annotation via moment's method and compared them to each other. The experimental results showed that the annotation system based on Legendre moments with Bayesian networks gives good results for images that are well and properly segmented. However, Image segmentation remains a challenge that needs more attention in order to increase precision and accuracy of the image annotation system.

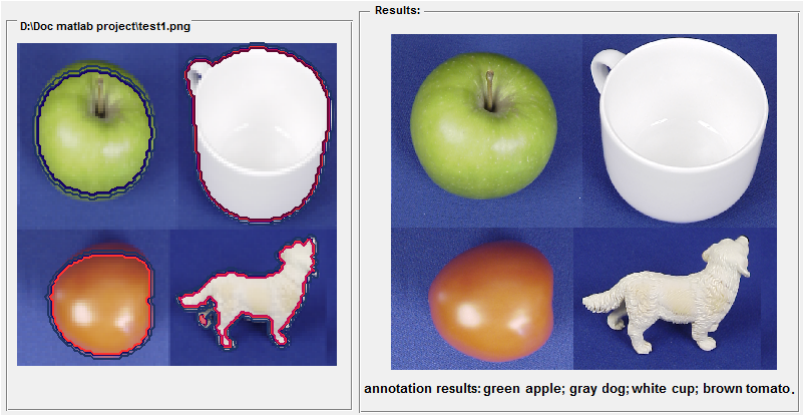


Figure 10.14: Graphical User Interface (GUI).

References

[1] O. Boiman, E. Shechtman, and M. Irani. In defense of Nearest-Neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008.

[2] Y. Cao, X. Liu, J. Bing, and L. Song. Using neural network to combine measures of word semantic similarity for image annotation. In *IEEE International Conference on Information and Automation (ICIA)*, pages 833–837, June 2011.

[3] C.W. Chong, P. Raveendran, and R. Mukundan. A comparative analysis of algorithms for fast computation of Zernike moments. *Pattern Recognition*, 36(3):731–742, 2003.

[4] C.W. Chong, P. Raveendran, and R. Mukundan. Translation and scale invariants of Legendre moments. *Pattern Recognition*, 37(1):119–129, 2004.

[5] R.S. Chora. Image feature extraction techniques and their applications for CBIR and biometrics systems. *International Journal of Biology and Biomedical Engineering*, 1(1):6–16, 2007.

[6] COIL-100. Columbia university image library. [online]. available: http://www.cs.columbia.edu/cave/software/softlib/coil-10image.

[7] R. Datta, D. Joshi, J. Li, and J.Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):5:1–5:60, 2008.

[8] K.B. Duan and S.S. Keerthi. Which is the best multiclass SVM method? An empirical study. In *Multiple Classifier Systems*, volume 3541 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2005.

[9] ETH-80. ETH Zurich image set [online]. available: http://www.d2.mpi-inf.mpg.de/datasets/eth80.

[10] K.M. Hosny. Exact Legendre moment computation for gray level images. *Pattern Recognition*, 40(12):3597–3605, 2007.

[11] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[12] S.K. Hwang and W.Y. Kim. A novel approach to the fast computation of Zernike

moments. *Pattern Recognition*, 39(11):2065–2076, 2006.

[13] G.H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *11th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 338–345, 1995.

[14] E.C. Kintner. On the mathematical properties of the Zernike polynomials. *Optica Acta: International Journal of Optics*, 23(8):679–680, 1976.

[15] R. Lepage and B. Solaiman. Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur. Technical report, Ecole de technologie supérieure, 2003.

[16] P. Leray. *Réseaux bayésiens : apprentissage et modélisation de systèmes complexes, Habilitation A Diriger Les Recherches, Spécialité Informatique, Automatique et Traitement du Signal*. PhD thesis, Université de Rouen, 2006.

[17] T. Mitchell. Generative and discriminative classifier: Naïve bayes and logistic regression. In *Machine learning*. McGraw Hill, 2010.

[18] P. Naïm, P.H. Wuillemin, P. Leray, O. Pourret, and A. becker. *Réseaux bayésiens*. Eyrolles, 3ème édition, Paris,, 2008.

[19] M. Oujaoura, B. Minaoui, and M. Fakir. Image annotation using moments and multilayer neural networks. *IJCA Special Issue on Software Engineering, Databases and Expert Systems (SEDEX)*, 1:46–55, 2012.

[20] M. Oujaoura, B. Minaoui, and M. Fakir. A semantic approach for automatic image annotation. In *International Conference on Intelligent Systems: Theories and Applications (SITA)*, pages 1–8, 8-9 May 2013.

[21] M. Oujaoura, B. Minaoui, M. Fakir, R. El Ayachi, and O. Bencharef. Recognition of isolated printed Tifinagh characters. *International Journal of Computer Applications*, 85(1):1–13, 2014.

[22] M. Oujaoura, B. Minaoui, M. Fakir, B. Bouikhalene, R. El Ayachi, and O. Bencharef. Invariant descriptors and classifiers combination for recognition of isolated printed Tifinagh characters. *International Journal of Advanced Computer Science and Applications*, 3(2):22–28, 2013.

[23] G.A. Papakostas, E.G. Karakasis, and D.E. Koulouriotis. Accurate and speedy computation of image Legendre moments for computer vision applications. *Image and Vision Computing*, 28(3):414–423, 2010.

[24] A. Prata and W. Rusch. Algorithm for computation of Zernike polynomials expansion coefficients. *Appl. Opt.*, 28(4):749–754, 1989.

[25] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[26] W. Shao, G. Naghdy, and S.L. Phung. Automatic image annotation for semantic image retrieval. In *Advances in Visual Information Systems*, volume 4781 of *Lecture Notes in Computer Science*, pages 369–378, 2007.

[27] P.Y. Simard and D. Steinkraus ang J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 958–963, August 2003.

[28] M.R. Teague. Image analysis via the general theory of moments. *J. Opt. Soc. Am.*, 70(8):920–930, 1980.

[29] R.S. Tumen, M.E. Acer, and T.M. Sezgin. Feature extraction and classifier combination for image-based sketch recognition. In *Sketch-Based Interfaces and Mod-*

eling Symposium (SBIM), pages 63–70, 2010.

[30] N. Vasconcelos and M. Kunt. Content-based retrieval from image databases: current solutions and future directions. In *International Conference on Image Processing (ICIP)*, volume 3, pages 6–9, 2001.

[31] Z. Yang and C.C.J. Kuo. Survey on image content analysis, indexing, and retrieval techniques and status report of MPEG-7. *Tamkang Journal of Science and Engineering*, 2(3):101–118, 1999.

[32] L. Ye, P. Ogunbona, and J. Wang. Image content annotation based on visual features. In *IEEE International Symposium on Multimedia (ISM)*, pages 62–69, 2006.