Chapter 7

Feature Identification and Extraction

Feature identification and extraction is perhaps the most important stage in most pattern recognition problems and specifically HCR. While, in theory, one can use the pixels in the character image as features for recognition, this is not useful in practice, due to the wide variation in the images of a single character. Choosing features that help to distinguish effectively among any given pair of characters, and still not err in recognizing all the possible variations of a given character is a major challenge. The inability to systematically characterize the possible and acceptable variations for a character makes this problem harder, often driven by the variations observed in whatever corpus is used in the experiment. As mentioned in chapter 5, our character corpus provides an extensive range of variations of all types from ink to writer. Thus, choosing an appropriate feature set that performs well over this corpus became a major challenge and the core part of the work.

This chapter first explains the features used (sections 7.1 to 7.4) for basic character set with both vowels and consonants. The feature identification and extraction for Kagunita recognition is discussed next (section 7.5). We identify each feature set with an abbreviation (written in parenthesis in the heading) to refer them in the figures and experiments. A full list is provided in the appendix III for easy reference.

7.1 Curvature Features (C)

Indian language scripts are rich in curves, and these curves can play an important role in identifying the character. Curvature features are, however, sensitive to noise and extracting
them is a complex task as compared to statistical features. Through observations and comparative studies, we have identified a few curvature related features, as described below.

Kannada characters have curve shaped (concave or convex) contours when observed from a boundary towards center of the image and vice versa. For example, when character े is observed from the four boundaries towards center, it has one long curve close to the left boundary, one small and one long curve close to the right boundary, one long curve close to the bottom boundary and two small curves close to the top boundary. Similarly, character ङ has one small close-by curve and one small far-away curve to the left boundary, one long and one small close-by curves to the right boundary, two small close-by curves at the bottom boundary and two long close-by curves to the top boundary.

Even though the length and the position of each curve may vary due to handwriting variations, number of such curves, relative position of the curves – close or away from the observation boundary and relative lengths of these curves usually remain unaffected after dynamic preprocessing. Hence they can be considered as useful features for the identification of Kannada characters. Therefore, we have formulated a number of features focusing on the curvature information like curve position and curve length.

### 7.1.1 Curvature position information (CPB, CPC)

To find the number of curves closer to a boundary and the number of curves away from it for all the eight direction observations, following method is proposed.

1. Find four different distance vectors from the $N \times N$ ($N = 64$ in our case) input image $f(x,y)$. A distance vector of size $N$ is given by $g(u_1, u_2, \ldots, u_N)$ where each $u_i$ for $i=1,\ldots,N$ with $i$ representing each row or column based on observation direction, is the distance of the first black pixel from a boundary in the $i^{th}$ row or column. This way we find the first four distance vectors $g$ named as $glb, grb, gbb$ and $gtb$ from all the four boundaries searching for the first black pixel towards center.

Similarly, we find another four distance vectors $glc, grc, gbc$ and $gtc$ searching for the first black pixel from the center line towards four boundaries of the image as shown in the figure 7.1.
2. A second order derivative function (7.1) is applied to the distance vector $g(u)$. As we use 1D distance information than 2D positional information from the image, this method reduces the amount of computation (1D uses only 2 neighbors whereas 2D uses 4 or 8 immediate neighbors for the second order derivative computation) to find the directional information of the curve. Eight derivative vectors are computed from eight distance vectors to find the curvature details.

$$g''(u) = g(u+1) + g(u-1) - 2g(u)$$ (7.1)

3. To find the number of individual concave or convex curves, we find the discontinuity of the contours with respect to change in the curve direction from a direction vector computed using second order derivative function. We consider the absolute value of $g''(u)$ with 2 or less as “smooth curve variation”. Value of 2 is considered to make it robust for the boundary ruggedness that may influence the smooth variations of the curve. Otherwise, if the value is $> 2$ then there is high variation. Therefore we consider this as one curve ends and another new curve starts. Using the distance vector, find the minimum distance of the recognized curve from the boundary in the first four cases. Sample values of $g(u)$ and $g''(u)$ for a specific character image are shown below to illustrate the curve computation.

<table>
<thead>
<tr>
<th>Shape</th>
<th>$g_{lb}(u)$</th>
<th>$g_{lb''}(u)$</th>
<th>curves</th>
</tr>
</thead>
</table>
|       | 10 6 4 2 2 3 6 8 11 16 13 12 10 8 6 4 3 3 5 7 10 | 2 0 2 1 2 -1 -1 2 8 2 -1 0 0 0 1 1 2 1 0 | 2 curves of type 1 - minimum distance of these curves are 2 and 3 respectively and are close to left boundary.
4. If a curve is closer to the boundary with some portion of curve touching a boundary or curve is within 1/3\textsuperscript{rd} of N (number of rows/columns) with respect to observation boundary or relatively close to boundary with another previously recognized curve of type 1 in the same observation direction, then it is marked as type 1 to indicate that the curve is close to the boundary. Otherwise it is marked as type 2 indicating that the curve is far from the boundary.

5. As a maximum of 3 counts of each type is expected (based on the manual character shape observation), 2 bits is sufficient for encoding. With two bits for type 1 and two bits for type 2 features computed from the distance vectors from boundary $glb$, $grb$, $gbb$ and $gtb$, the feature vector size is 16. This feature set is called CPB.

6. Similarly curve positions are extracted from the distance vectors $glc$, $grc$, $gbc$ and $gtc$ from center of the image towards boundary using steps 2, 3 and 4 as above and encoded using 2 bits each resulting in another 16 features. This feature set is called CPC.

\subsection*{7.1.2 Curve Length (CLB, CLC)}

This feature considers the length of the curves of each of the types (type1 and type2) as defined above. The 8 length features CLB are lengths of CPB curves observed from the boundary towards the center and another 8 length features CLC are the lengths of CPC curves observed from center toward boundary. While finding the curve types from each direction of observation, the length information is also computed by counting the pixels under each type.

\subsection*{7.2 Statistical Features (S)}

Statistical features are used to capture the density of the information and its distribution in the image. All characters are not uniformly distributed in density; some are top heavy, some could be left heavy etc. These features help to group characters based on such properties.
7.2.1 Global statistical features

We have identified two types of features of a statistical nature that capture information about the full image: information content and aspect ratio. These are described below.

7.2.1.1 Information content (GI)

Some characters have more number of curvatures making image to have more information in the same space. Thus some characters are denser than others. Based on this, we define GI as the ratio of total number of black pixels to the total number of pixels in the image using equation (7.2), where $T_{black}$ is the count of black pixels and $T_{all \, pixels}$ is the total pixels in the image.

$$GI = \frac{T_{black}}{T_{all \, pixels}} \quad (7.2)$$

7.2.1.2 Aspect Ratio (GA)

The observation of Kannada basic characters shows that the characters vary in shape spanning thin, wide, squared etc. As the image is size normalized without affecting the original shape by padding background to make a square image, this shape is largely preserved in the preprocessed image. The minimum and the maximum positional values of x and y in the image $f(x,y)$ indicating the rectangle in which the shape is confined are used in the computation given by equation (7.3). For example, a square shaped character with size

$$AR = \frac{y_{max} - y_{min}}{\sqrt{(y_{max} - y_{min})^2 + (x_{max} - x_{min})^2}} \quad (7.3)$$

7.2.2 Zonal statistical features

The image can be divided into a number of Zones or windows to extract the features based on information in certain parts of the image. The zones can be independent or overlapping. They can be of fixed or varying sizes. As zones localize the information for feature extraction, the location dependent geometrical, positional, and statistical etc, features can be extracted. We have used two zonal statistical features: zonal density feature and zonal probability distribution. These are described below.
7.2.2.1 Zonal density feature (ZD)

Kannada characters have varying density of information across different parts/zones of the image, based on the shape. We use zonal approach with horizontal and vertical histograms of foreground density to build features characterizing this information. The feature extraction method is as follows:

- The density is measured using horizontal and vertical histograms.
- The image is divided into zones by dividing the image into 1x3 zones to extract vertical zonal density and 3x1 zones to extract horizontal zonal density.
- In each region, count the number of black pixels in each row.
- Five buckets are used to hold respectively the count of the rows having 1 to 2 pixels, 3 to 4 pixels, 5 to 6 pixels, 7 to 8 pixels and more than 8 pixels per region. The normalized count within each bucket is then, used as feature.

With 5 buckets, the number of features is 5. With 3 horizontal regions, we have 15 features. Similarly from 3 vertical regions, another 15 features are extracted. This gives a set of 30 features.

7.2.2.2 Zonal Probability Distribution (ZPD)

The Zonal probability distribution of information is calculated using equation (7.4) where \( T_{\text{zone}} \) is the count of black pixels in zone and \( T_{\text{total}} \) is the total black pixels in the image resulting in one feature per zone. In our experiments the number of zones considered to extract this feature varies depending on the image (whole image or portion of the whole image (cut image)).

\[
ZPD(z) = \frac{T_{\text{zone}}}{T_{\text{total}}} \tag{7.4}
\]

7.3 Moments (M)

Moments and functions of moments have been used as pattern features in a number of applications to achieve invariant recognition of two-dimensional image patterns. Hu [43] first introduced moment invariants in 1961, based on methods of algebraic invariants. Using nonlinear combinations of regular moments (regular moments will be referred to as geometric
moments henceforth), he derived a set of invariant moments which has the desirable properties of being invariant under image translation, scaling, and rotation. Moments can be applied to binary or grey level images, defined in 2D, 3D and higher dimensional space, and also to edges and primitives extracted through a preprocessing stage.

The geometric moment is the projection of \( f(x,y) \) onto the monomials \( x^py^q \). Unfortunately, the basis set \( \{x^py^q\} \) is not orthogonal. Consequently, these moments are not optimal with regard to the information redundancy and hence it is difficult to recover the original image from its moments. Still the geometric moments are most widely used in image analysis and pattern recognition tasks due to their simplicity, the invariance properties and geometric meaning of the low order moment values. We use geometric moments and central moments in our experiments; these are described below.

### 7.3.1 Geometric Moments (GM)

The geometric moments are as mentioned, basically projections of the image function \( f(x,y) \) onto the monomials \( x^py^q \). For a digital image \( f(x,y) \) of size \( M \times N \), the geometric moment \( m_{pq} \) of order \( p+q \) is given by (7.5).

\[
m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x,y) \quad \text{with} \quad p,q = 0,1,\cdots,\infty
\]

(7.5)

The zero order moment \( m_{00} \) can be seen to be the total mass (or power) of the image and gives the number of pixels comprising the object. The first order moments, \( \{m_{10}, m_{01}\} \), give the displacement of the object in \( x \) and \( y \) directions and are used in finding the centroid (Center of Moment – COM) of the object. The coordinates of the centroid \( (\bar{x}, \bar{y}) \) give the intersection of the lines, \( x = \bar{x} \) and \( y = \bar{y} \), parallel to the \( x \) and \( y \) axes respectively, about which the first order moments are zero. In terms of moment values, the coordinates of the centroid are as in (7.6)

\[
\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{and} \quad \bar{y} = \frac{m_{01}}{m_{00}}
\]

(7.6)

The second order moments, \( \{m_{02}, m_{11}, m_{20}\} \), known as the moments of inertia, give the balancing factor (central-ness) and the third order moments \( \{m_{30}, m_{03}\} \) give the symmetry information of the character contour.
As the image is not normalized with respect to translation, rotation and scale, the features will have the influence of all these variations in them reflecting the true moments of the object shape before normalization. Hence the features like displacement, central-ness and symmetry are all reflecting the actual shape of the character. Using the 0th, 1st, 2nd and 3rd order moments of the character images, four GM features are computed using functions (7.7) [108]. As these features give the overall $x$, $y$ and diagonal direction variations with respect to displacement, central-ness and symmetry, we chose to use them as features.

\begin{align}
    f_1 &= m_{20} + m_{02} + m_{00} \\
    f_2 &= \sqrt{(m_{20} - m_{02})^2 + m_{11}} \\
    f_3 &= \sqrt{(m_{10} - m_{01})^2} \quad \text{and} \\
    f_4 &= m_{30} + m_{03}
\end{align}

(7.7)

### 7.3.2 Central Moments and HU’s moments (CM and HU)

To make the features translation invariant, the $M \times N$ image plane $f(x,y)$ is first mapped onto a square defined by $x \in [-1, +1]$ and $y \in [-1, +1]$ and from the mapped image the central moments are computed using (7.8).

\begin{equation}
    \mu_{pq} = \sum_{x=-l}^{+l} \sum_{y=-l}^{+l} (x - \bar{x})^p (y - \bar{y})^q f(x,y) \quad \text{with } p,q = 0,1,\ldots,\infty
\end{equation}

(7.8)

Each moment extracts some information from the image that is not affected by the position of the object in the image.

- $\mu_{00}$ – area or information content of the image
- $\mu_{20}$ – horizontal central-ness
- $\mu_{02}$ – vertical central-ness
- $\mu_{11}$ – diagonality - indication of the quadrant with more mass with respect to centroid
- $\mu_{03}$ – vertical deviation (symmetry)
- $\mu_{30}$ – horizontal deviation (symmetry)

We use these moments to define the features relating to central-ness and deviation in the image shapes.
Three Non linear functions using central moments (CM) invariant to translation are given by equation (7.9) [108]. These constitute 3 CM features.

\[
g_1 = \mu_{20} + \mu_{02} + \mu_{00} \\
g_2 = \sqrt{\left(\mu_{20} - \mu_{02}\right)^2 + \mu_{11}} \quad \text{and} \\
g_3 = \mu_{30} + \mu_{03}
\]

Invariance to scale is achieved by calculating the new set of moments \( \eta_{pq} \) from central moments using (7.10).

\[
\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\nu} \quad \text{with} \quad \nu = \frac{p + q + 2}{2}
\]

The sign and the magnitude of the central moments provide useful clues for character recognition based on symmetry properties. The shapes that are symmetric either about the \( x \) or \( y \) axes will produce \( \eta_{11} = 0 \). For shapes symmetrical about the \( y \) axis, \( \eta_{12} = 0 \) and \( \eta_{30} = 0 \). However for shapes symmetric about the \( x \) axis, \( \eta_{03} = 0 \) and \( \eta_{12} \) is positive. Some examples are shown in the figure 7.2 with 7.2(a) highlighting symmetry about \( y \) axis and 7.2(b) the symmetry about \( x \) axis. Table 7.1 shows qualitative values for some of these parameters for the two English characters shown in figure 7.2, illustrating their role. The English typed characters are used in the example to demonstrate the perfect symmetry around the axis.

![Figure 7.2 Axes of symmetry](image)

<table>
<thead>
<tr>
<th>Character</th>
<th>( \eta_{11} )</th>
<th>( \eta_{20} )</th>
<th>( \eta_{02} )</th>
<th>( \eta_{21} )</th>
<th>( \eta_{12} )</th>
<th>( \eta_{30} )</th>
<th>( \eta_{03} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>0</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>+</td>
<td>+</td>
<td>0</td>
</tr>
</tbody>
</table>

In order to make the central moments invariant to rotation, Hu defined 7 non linear functions (HU) using the scale and translation invariant moments \( \eta_{pq} \), as shown in equations (7.11).
\( \phi_1 = \eta_{20} + \eta_{02} \)
\( \phi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2 \)
\( \phi_3 = (\eta_{30} + 3\eta_{12})^2 + (3\eta_{21} + \eta_{03})^2 \)
\( \phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \)
\( \phi_5 = (\eta_{30} + 3\eta_{12})(\eta_{30} + \eta_{12})[2(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \)
\( \phi_6 = (\eta_{30} - \eta_{02})[2(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \)
\( \phi_7 = (3\eta_{12} - \eta_{30})(\eta_{30} + \eta_{12})[2(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \)

Each Hu’s moment represents some feature of the character shape in the image that are not influenced by the position, size and the angle of orientation of the character shape in the image. They are as follows:

\( \phi_1 \): The sum of horizontal and vertical variance. If the character shape is more distributed towards horizontal and vertical axes, the values are enlarged.

\( \phi_2 \): The covariance value of vertical and horizontal axes.

\( \phi_3 \): Emphasizes the object inclination to left/right and upper/lower axes.

\( \phi_4 \): Emphasizes the object counterbalancing to left/right and upper/lower axes.

\( \phi_5, \phi_6, \phi_7 \): These values, invariant against size, rotation, and location, help to distinguish mirror images \( g(x,y) = f(M - x, y) \) for \( M \times N \) image \( f(x,y) \) of otherwise identical images.

The numerical values of \( \phi_1 \) to \( \phi_7 \) are very small. Hence, the logarithm of the absolute values of these functions is used as Hu’s features. The inter class mean and the Standard Deviation (SD) of the features are used for the normalization.

Sample moments and the feature values for two Kannada characters are as follows:

<table>
<thead>
<tr>
<th>Image</th>
<th>Geometric Moments</th>
<th>Central Moments</th>
<th>GM features</th>
<th>CM features</th>
<th>HU features</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{00} = 177 )</td>
<td>( \mu_{00} = 177 )</td>
<td>( f_1 = 424134 )</td>
<td>( g_1 = 261.9 )</td>
<td>( \phi_{1} = -2.567 )</td>
<td></td>
</tr>
<tr>
<td>( m_{01} = 5161 )</td>
<td>( \mu_{01} = 0.0 )</td>
<td>( f_2 = 247038 )</td>
<td>( g_2 = 2.428 )</td>
<td>( \phi_{2} = -5.739 )</td>
<td></td>
</tr>
<tr>
<td>( m_{10} = 5746 )</td>
<td>( \mu_{10} = 0.0 )</td>
<td>( f_3 = 585 )</td>
<td>( g_3 = -0.725 )</td>
<td>( \phi_{3} = -9.059 )</td>
<td></td>
</tr>
<tr>
<td>( m_{11} = 184755 )</td>
<td>( \mu_{11} = 16.809 )</td>
<td>( f_4 = 1858553 )</td>
<td>( \phi_{4} = -10.090 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( m_{20} = 243120 )</td>
<td>( \mu_{20} = 55.259 )</td>
<td></td>
<td>( \phi_{5} = -20.810 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( m_{02} = 180837 )</td>
<td>( \mu_{02} = 29.640 )</td>
<td></td>
<td>( \phi_{6} = -13.242 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 7.4 Directional Images (G)

The directional images are the images representing the contours of the characters in a selected direction as foreground. We use directional images to represent the curvature information in the character images. To find the directional information in the image, we used Gabor transformation as it is mathematically well defined and can be fine tuned. They are less sensitive to noise and small amount of translation, rotation and scaling – the kind of problems usually present in handwriting.

#### 7.4.1 Gabor Filter

Gabor filter decomposes an input image into a number of filtered images, each of which contains intensity variations over a narrow range of frequency and orientation. A 2D Gabor filter can be described by the impulse response function (7.12) at a sampling point \((x,y)\) with wavelength \(\lambda\), oscillation frequency \(f_0 = 1/\lambda\) and oscillation orientation \(\theta\).

\[
h(x, y, \lambda, \theta) = g(x', y') e^{j2\pi (u_0x + v_0y)}
\]

with

\[
g(x, y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}((x/\sigma)^2 + (y/\sigma)^2)}
\]

where \(x' = x\cos\theta + y\sin\theta\) and \(y' = -x\sin\theta + y\cos\theta\) are the rotated coordinates and \(\sigma = \sigma_x = \sigma_y\) is the standard deviation along the x-direction and y-direction which decides the spread (scale or window size) of the 2D Gaussian envelope. The spatial frequencies are \(u_0 = \ldots\)
\( f_0 \cos \theta \) and \( v_0 = f_0 \sin \theta \). \( \sigma_x \) and \( \sigma_y \) can be different and are the function of \( \lambda \) which decides the feature stability for scaling, translation and rotation. If we increase \( \sigma \), we look at a large space axis (increase in the window size) but at a smaller frequency axis and vice versa.

The word wavelet means a small wave with finite energy that has the ability to allow simultaneous time and frequency analysis. Hence the Gabor filter is a wavelet with 2 free parameters \( f_0 \) and \( \theta \) that help in generating a large number of Gabor wavelets. In our case since the images are thinned, one scale (window size) is sufficient to compute the contour directional information. Since the shape, direction and the amount of curvature of the handwritten characters vary, the fine angular responses do not convey the character details. So we are interested in only 4 major directions. Selection of one scale and 4 direction combinations constitutes a set of 4 Gabor wavelets.

### 7.4.2 Gabor Transformation

Gabor Transformation is the process of Fourier Transformation of the input image with a Gaussian window; it transforms the input image into Gabor Transform domain. For a character image \( f(x,y) \), its Gabor transform result is obtained by applying Gabor wavelet \( h(x,y,\lambda,\theta) \) with window size \( x \in (0..M) \) and \( y \in (0..N) \). The response output is given by equation (7.14).

\[
I(x, y; \lambda, \theta) = \sum_{x = x_M / 2}^{x + M / 2} \sum_{y = y_N / 2}^{y + N / 2} f(x', y') \cdot h(x, y, \lambda, \theta)
\] (7.14)

These outputs exhibit perfect local space characteristics, frequency characteristics and orientation selectivity of the image. We do not use the Gabor directional feature information as features directly. Instead we further extract features from these directional images. The directional feature information is again converted into a binary image and from these images we extract moment features and statistical features. The directional images of Kagunita characters with different aspect ratio are shown in figure 7.3.
7.5 Cut images for Kagunita (C)

The Kannada Kagunita is a set of 510 compound characters formed by combining one of the 15 vowels with each of the 34 basic consonants. Recognizing Kagunita means recognizing 510 different aksharas, a very large number of classes and so is a very complex task. Since Kagunita is a combination of vowels and consonants, if we identify these two components in a given akshara, then we can recognize a Kagunita. Hence the problem reduces to recognizing a vowel from 15 vowels and a consonant from 34 consonants from a Kagunita. This is the approach we explore in our research.

However, the recognition of vowel in the presence of the 34 variations of consonant and the recognition of a consonant under the presence of 15 vowel variations is a difficult task. This raises issues of looking at two types of feature extraction – one suitable for identifying the vowel element and another set for the consonant element. In order to derive clues distinguishing these two, we analyze the various Kagunita shapes.

The addition of a vowel to a consonant modifies the consonant shape in a Kagunita. Such variation in shape normally happens at the periphery of the consonant. Kannada script Kagunita analysis shows that it does not have any Kagunita character with a left-side dependent vowel sign where the left side of the consonant is modified by a vowel. The vowel sign is on the top, bottom or right of the consonant character as shown in the 2nd row of figure 7.4. We find four top variations of same size and only two variations in the bottom. Most of the matra variation is observed on the right of the consonant and can be grouped into four categories as shown in the 6th row in figure 7.4.
Category 1: no matra to the right
Category 2: matra with one curve to the right
Category 3: matra with two curves to the right
Category 4: matra with three curves to the right

Figure 7.4 grouping of matra based on the matra position

To make the vowel information prominent for its recognition, a portion can be cut, say around 20%, on the top and bottom respectively giving two different images. To recognize the right side vowels, experimentally, we found four different size images more efficient with one original image and 3 cut images of sizes 20%, 40% and 60% from the right so that at least one category can best represent the original matra. Hence there will be around 6 images to help in the recognition of vowel in the Kagunita as shown in the figure 7.5. In the figure 20T 20R and 20B means 20% top, 20% right and 20% bottom cut images respectively.

Figure 7.5 % cuts on the original images for vowel information
We use such cut images to recognize the consonant in the Kagunita also. As the vowel matra size varies to the right and the images are size normalized, the size of the consonant also varies. The four right vowel matra sizes result in four different size consonants. If there is no vowel matra to the right then, consonant occupies the whole image. If vowel matra is 20%, then consonant occupies 80% of the image from left and so on. Therefore, to recognize the consonant we considered again four images with one original and three cut images of sizes 40%, 60% and 80% as shown in the figure 7.6. The notation 40, 60 and 80 means three cut images with sizes 40%, 60% and 80% of the original image. As this size is decided manually, different % cut image sizes are experimented to find the suitable cut size for both vowel and consonant recognition in a Kagunita.

![Figure 7.6 % cuts on the original images for consonant information](image)

Features of the type outlined in previous section of this chapter, are extracted from these cut images also. Based on various studies (described in section 2.4), we expect to narrow down on an appropriate set of features and the selection of cut images; for Kannada HCR.

Based on study of relevant literature, the nature of Kannada alphabet including the nature of Kagunita and various intermediate empirical investigations, we have been able to arrive at a comprehensive set of features useful for many Indian language scripts. These features span from simple image based observations like curvature and zonal density to highly specialized moments. The base images used include the preprocessed image, directional images and cut images (for Kagunita). After discussing the last component in the HCR pipeline, the recognition engine, in the next chapter, we present the various studies involving these features in chapter 9.