Chapter 6

Pin code Segmentation and Recognition

Pin code Script Identification, Segmentation and Recognition

6.1 Introduction

Localization of pin code was performed on the skew corrected DAB image (Fig. 6.3.(a)). Recognition of identified pin code was performed using Hybrid Classification method and obtained an accuracy of 97%. Performance comparison of most referenced classification methods used in literature were discussed and the results were recorded. It was observed from the experiments that every classification method has its own advantages and disadvantages. It is the efficiency of the researcher to choose the appropriate classification method for best results based on time and computational cost.

6.2 The Proposed Methodology

Identification or localization of the pin code blob among the other address related blob was performed using Template Matching technique. The identified pin code was segmented and processed using Hybrid Classifier for recognition of the digits. As an initial step towards pin code localization the extracted DAB image was segmented into text lines (Line segmentation) and then the text lines were segmented into words (Word segmentation). The proposed system structure is given below in Fig. 6.1.

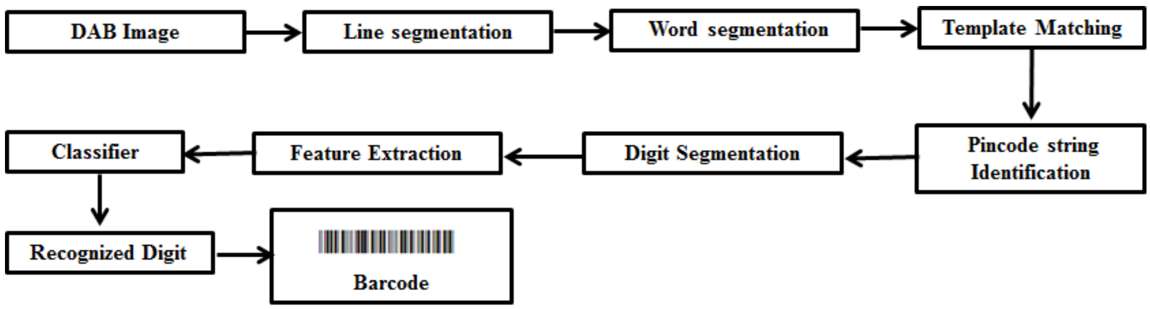

Figure 6.1: System Flow for Pin Code Identification

6.2.1 Line Segmentation

A simple approach for line and word segmentation was proposed in initial stages of the research work [72]. Line segmentation and word segmentation was performed on the DAB segmented image (Fig. 6.3(a)). The image contains n Rows and m Columns (Fig. 6.2). The foreground colour of the image is white (1) or on pixel and background colour is black (0) or off pixel.

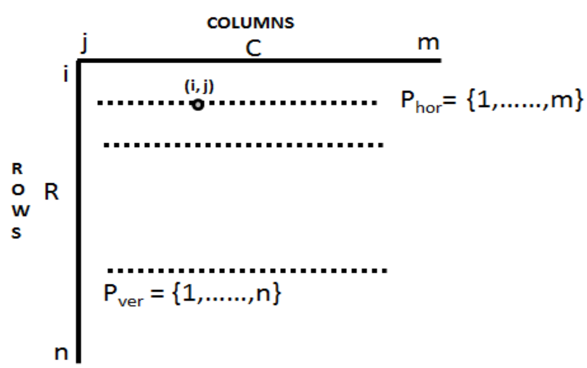

Figure 6.2: Rows and Column Description in Image

- The horizontal projection profile (HPP) of the DAB image (Fig 6.3) was obtained. $HPP_i(\sum_{i=1}^{m} P_{hor} \epsilon R_i)$ indicates the sum of pixels horizontally (row wise) R_i .

- W and H determines the width and height of the image.

Algorithm 7:

Step 1: To extract the text lines the adjacent row values (R_i) current row, (R_{i+1}) next row were computed. If $R_i - R_{i+1} == -1$, when $R_i = 0$ and $R_{i+1} = 1$ then it indicates the start of text line. The text line image is cropped when $R_{i+1} - R_{i+n} == 1$.

Step 2: The cropping coordinates are $(Startl = R_{i+1}, C1, W, HP)$, where $HP = Startl + End$ (i.e. $Endl = R_{i+n}$).

Step 3: The remaining image is repeated with Step 1 to 3 until the last line of the text block is segmented.

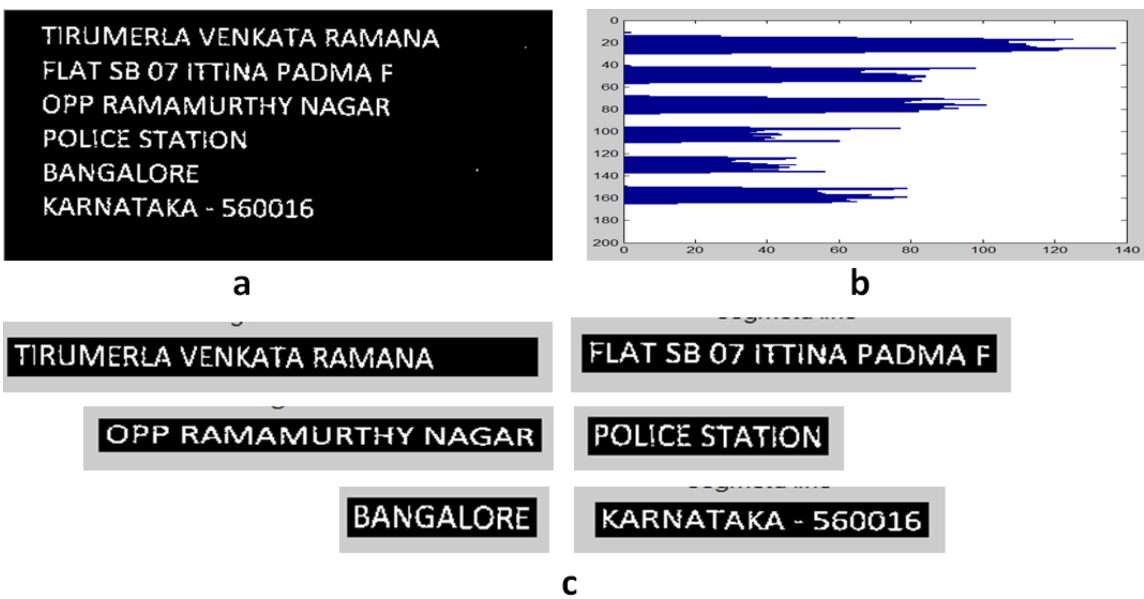

Figure 6.3: a) DAB b) HPP c)Segmented text lines

6.2.2 Word Segmentation

To the line segmented image morphological operation dilation is performed by sizing the structuring element by calculating the average (heights) of CC in the line segmented image. Thus every isolated character in every text line is combined into bigger connected components. Thus word segmentation is performed using connected component analysis using vertical projection profile (VPP).

Algorithm 8:

Step 1: The vertical projection profile (VPP) of the DAB image (Fig. 6.4(c)) was obtained. $VPP_j(\sum_{i=1}^{n} P_{ver}\epsilon C_j)$ indicates the sum of pixels vertically (column wise).

Step 2: To extract the word the adjacent column values C_j and C_{j+1} is computed. If $C_j - C_{j+1} == -1$, when $C_j = 0$ and $C_{j+1} = 1$ then it indicates the start of word component. The word image is cropped when $C_{j+1} - C_{j+m} == 1$.

Step 3: The corresponding row number R_i for C_{j+1} were extracted.

Step 4: The cropping coordinates are $(Startw = C_{j+1}, R_i, WP, HP)$, where $WP = Startw + Endw$.(i.e. $Endw = C_{j+m}$).

Step 5: The remaining image is repeated with Step 2 to 4 until the last CC is segmented.

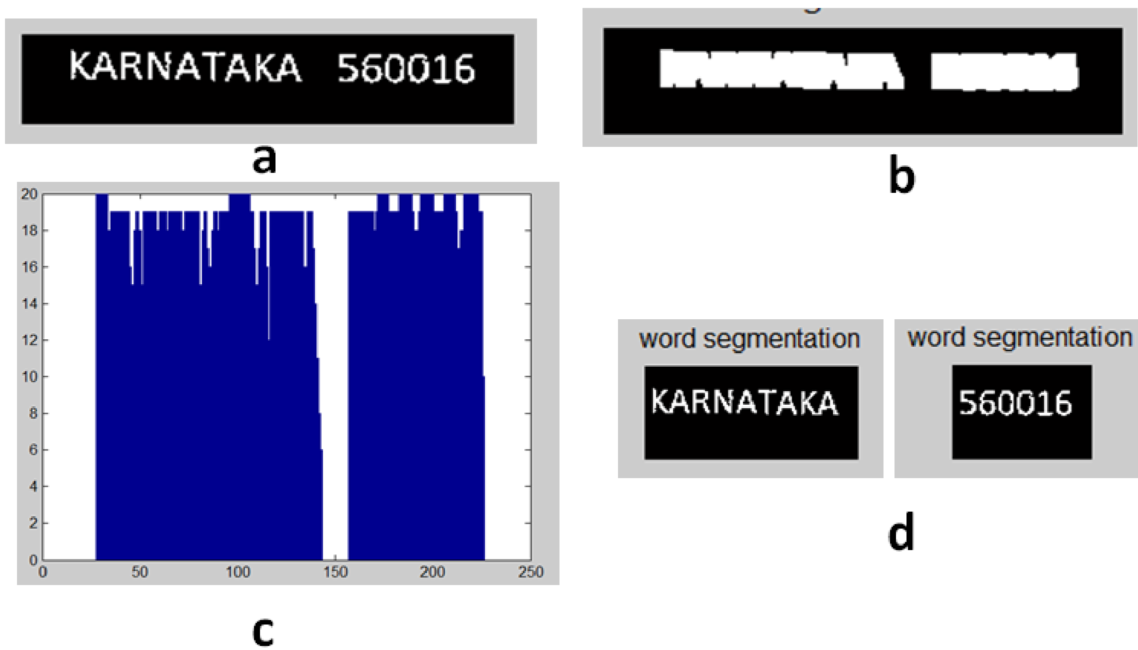

Figure 6.4: a)Segmented Text line b)Dilated CC c)VPP d)Segmented Words

Algorithm 7 and Algorithm 8 is represented in Pseudo Code 7.

Pseudo Code 7 HPP and VPP

Input: $t(x, y), ht, D_{par}, W, H$

Output: segmented lines and words

Procedure HPP and VPP

$\forall R_i \exists P_{hor_i}$ `where` $i = 1, ..., m \epsilon t(x, y)$

$HPP_i = \sum_{i=1}^{m} P_{hor} \epsilon R_i$

`For every Row` R_i

`if` $R_i - R_{i+1} = -1$ ▷ Start of text line

$startl = R_{i+1}$

`if` $R_{i+1} - R_{i+n} = 1$ ▷ End of text line

$stopl = R_{i+n}$

$HP = startl + stopl$ ▷ Height of text line

`Crop` $t(x, y)$ using $\{start, C_1, W, HP\}$ ▷ Line segmentation

`The cropped image is stored in` $L_i(x, y)$.

$L(x, y) \oplus SE$

$\forall C_j \exists P_{ver_j}$ `,where` $j = 1, ..., n \epsilon L(x, y)$

$VPP_j = \sum_{j=1}^{n} P_{ver} \epsilon C_j$

`For every Column` C_j

`if` $C_j - C_{j+1} = -1$

$startw = C_{j+1}$

`if` $C_{j+1} - C_{j+m} = 1$ ▷ End of word

$stopw = C_{j+m}$

$WP = startw + stopw$ ▷ width of word

`Extract the` R_i `for` $startw$ `column`

`Crop` $L_i(x, y)$ using $\{R_i, startw, WP, HP\}$ ▷ Word segmentation

`The cropped image is stored in` $W_i(x, y)$`.`

$RemImgcoor = \{stopl, C_1, W, H - HP\}$ ▷ Remaining Image

$t(x, y) = RemImg$

`if` $RemImg == 0$

`end`

`else`

`GOTO line no 2`

End Procedure

6.3 Pin code Identification

The Indian pin code contains only six digits. Hence, every segmented word image is analysed for the presence of six components in it. Only the word image that contains six components are considered. All the extracted or segmented blob having six components are processed for the presence of alphabet or digits. This is performed using Template Matching technique.

6.3.1 Template Matching

Template Matching is a high level machine vision technique that identifies parts on an image that matches a predefined image or template. Hence two templates were created consisting of alphabets and digits. Template-I constitutes 36 predefined images of 26 (A-Z) alphabet and 10 (0-9) numerals. Template-II constitutes 10 predefined images of 10 (0-9) numerals. These templates contains predefined images of alphabets and numbers that belong to the commonly used fonts.

The Template Matching was performed for the reference image and the input image of normalized size 44 x 26. The predefined templates were compared with the segmented digits of the pin code. The comparison of the predefined or reference image and the input image is performed using Correlation Coefficient (r) .

The Pearson correlation [22] measures the degree and direction of the linear relationship between two variables A and B . It returns the correlation coefficient r between A and B , where A and B are matrices or vectors of the same size.

$$r = \frac{\sum_i (X_i - X_m)(Y_i - Y_m)}{\sqrt{\sum_i (X_i - X_m)^2}\sqrt{\sum_i (Y_i - Y_m)^2}} \tag{6.1}$$

Thus X_i is the intensity of the i^{th} pixel in image A , Y_i is the intensity of the i^{th} pixel in image B , X_m is the mean intensity of the image A and Y_m is the mean intensity of the image B .

The Correlation co-efficient has the value $r = 1$ if the images A and B are absolutely correlated, $r = 0$ if they are completely uncorrelated and if they are anti-correlated (complement image) $r = -1$. Hence the correlation coefficient values ranges between -1 to 1 .The following steps explains the process of template matching technique.

Algorithm 9:

Step 1: The extracted pin code blob is further segmented into individual character image using CCA .

Step 2: The segmented image is resized to 44 x 26.

Step 3: The normalized segmented character image is compared with the Template-I images.

Step 4: Correlation coefficient is calculated using the above formula as in Equation 6.1. The maximum value obtained is found to be the best match value.

Step 5: Correlation coefficient is calculated for Template-II and the input image.

Step 6: The number of occurrence of alphabets and numerals were calculated using Step 4 & 5 for every segmented blob containing six components.

Step 7: If the total number of occurrence of digits is greater than or equal to 9, then that extracted blob is concluded as the blob containing the pin code.

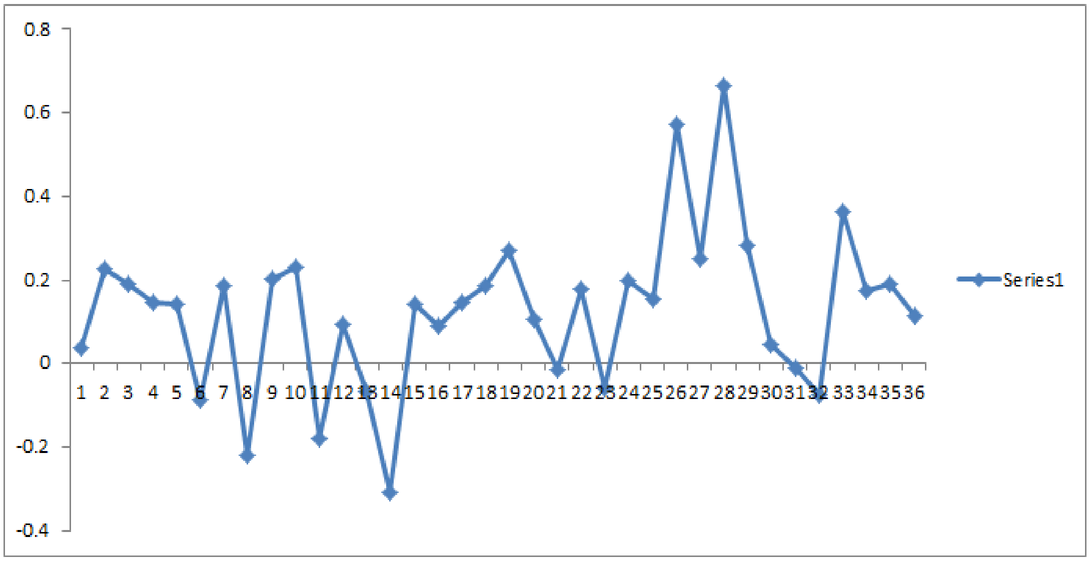

Figure 6.5: Co-relation Co-efficient

In Fig. 6.5, the horizontal axis indicates the template numbers 1 to 26 for alphabets $[A - Z]$ and the numerals $[0 - 9]$ are numbered as 28 to 36. The vertical axis indicates the correlation co-efficient values in the range -1 to 1. The graph shows the higher correlation value (r) for the digit image 2. This method yields 98% accuracy in identifying whether the given character is a digit or alphabet.

6.3.2 Evaluation of Template Matching

300 segmented word images from the IPBME images were tested to evaluate the template matching technique (Algorithm 9). 98% accuracy was obtained in identifying the correct blob that contains the pin code. The accuracy fails when it contains deformed or broken alphabet or number. In order to further

evaluate the accuracy of Template matching technique. The Template-I was tested with 60 set of each containing 26 alphabets (1560-alphabets) of varied font names and sizes. The overall accuracy of 90% was obtained. The results are recorded in Table 6.1 and the confusion matrix is shown in Fig. 6.6.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	5	6	7	8	ACC %
A	58																														100%
B		49																	1					2		1	1	1		3	92%
C			48	2	5							2																1			90%
D				57											1																99%
E					55																					3					97%
F						52			1						1											1			3		94%
G			1	1			50								1													5			92%
H		1			2	1		54																							96%
I									58																						100%
J										51										4									3		93%
K											54													4							96%
L												57														1					99%
M								3					48	3								2			2						90%
N														57								1									99%
O				5											53																95%
P				1		5										52															94%
Q				2										5			51														93%
R															9			49													92%
S		4			5														33								5			1	75%
T									1											57											99%
U																					58										100%
V														1								57									99%
W																						2	56								98%
X																								58							100%
Y																									58						100%
Z																										58					100%

Figure 6.6: Confusion Matrix for Template Matching

Table 6.1: Experimental Observation

Method Used	Testing Samples	Classified Characters	Accuracy
Template-I	1560	(A-Z) and (0-9)	92%
Template-II	160	(0-9)	98 %

6.4 Feature Extraction

Many methods have been used in literature for extracting features like statistical features, structural features, contour features as listed by [99]. Multi or hybrid feature extraction is a combination of various feature extraction methods. In the proposed feature extraction method hybrid feature extraction was performed. Unique features have to be differentiated and extracted from the segmented digits. These extracted features serve as the input for the classifier. The well defined feature set increases the accuracy of the classification.

Various features like Chain code, inner boundary (Holes), outer boundary, vertical line, horizontal line and centroid were extracted from the segmented digit image and serve as input vector for various classifiers. The pin code recognition was performed using the most referenced classification methods in literature like multi-layer perceptron (MLP), Naive Bayes classifier (NBC), K-nearest neighbour KNN and the proposed Hybrid Classification method [73]. The results are recorded.

6.4.1 Freeman Chain Codes

Freeman Chain Codes are used to extract boundary features [55]. It represents the shape of an image and encodes the boundary of the image. Chain Codes can be computed for any shape of image objects. Chain Code features can be used for recognition of characters and digits [37]. This technique possesses many advantages as listed by [80].

Chain codes provide a lossless compression [55] and increases the speed and effectiveness of the analysis patterns.

Two types of boundaries have to be considered, they are 4-connected and 8-connected. For 4-connected boundary, there are four possible directions. For 8-connected, there are eight possible directions (Fig. 6.7(a)).

In the proposed work feature extraction was performed using 8-connected Chain Code algorithm (Fig. 6.7) [30]. By using the Chain Code algorithm a vector containing the directional features were obtained. The Chain code varies based on the size. The values have to be normalized before being sent as an input for recognition. Normalization of Chain Code was performed to reduce the size of the feature vector. Normalization of chain code was performed using the below algorithm.

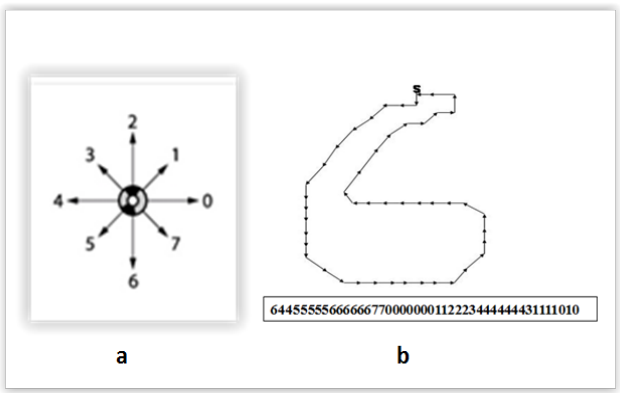

Figure 6.7: a) 8-Connected Boundary b) Image boundary and its Chain code

6.4.2 Normalization of Chain Code Algorithm (8-connected)

Algorithm 10:

Step 1: The first leftmost on pixel is located. Starting direction ($Start_{dir} = 6$) is set to trace the boundary of the image (Fig. 6.7).

Step 2: The following formula $Next_{dir} = Start_{dir} + d \mod 8$, was used to calculate the next direction. Where $d = \{0, 1, 2, 3, 4, 5, 6, 7\}$ the 8 possible direction for locating the next direction.

Step 3: The first occurring on pixel in these directions is updated as the $Next_{dir}$.

Step 4: The co-ordinates and the chain code directional values for every updated direction is stored.

Step 5: This gets repeated until it reaches the starting pixel co-ordinate again.

Step 6: The frequency (CF) of occurrence of every chain code directional values were calculated.

Step 7: The sum value of CF is calculated and it was found equal to the length of the obtained Chain code $(Chain_{len})$ and average length $avg_{len} = \sum CF / Chain_{len}$ was calculated.

Step 8: Further the Chain code was normalized by dividing the individual frequency of occurrence of every chain code direction values by the total Chain code. Thus the length of column vector was reduced.

$$NormalizedValue = \frac{CF_i}{\sum_{i=1}^{n} CF_i} * 10 \tag{6.2}$$

Step 8: Finally eight directional feature vector values $FV_1 = (DV_i = 0, 1, ..., 7)$ were obtained. The above steps were applied to all the digit images and the normalized Chain codes were obtained.

The major advantage of this approach was that the image could be of any size. The obtained Chain code was normalized to eight feature vectors. These feature vectors serves as the input and the training set for the MLP-BP and NBC. Algorithm 10 is represented in Pseudo Code 8.

Pseudo Code 8 Chain Code

Input: $P(x, y)$

Output: Chain code directional values

Procedure Normalization of Chain Code

$P(x, y) = \{p_1 \ldots\ldots\ldots\ldots p_n\}$

`Let co-ordinate of` $P_1 = (S_x, S_y)$ ▷ Leftmost pixel

$start = (S_x, S_y)$

$Start_{dir} = 6$

`for` $k = 0$ `to 7`

$Next_{dir} = Start_{dir} + k \mod 8$

`if` $p_i == 1$

`Extract the co-ordinates` (x, y) `for` $p_i == 1$

$Next_{dir} = k$

▷ Next direction

`if` $S_x \neq x \ \&\& \ S_y \neq y$

$Dir = Next_{dir}$

$Co - or = [x, y]$

`next k`

```
else

break

end for
```

$CF_i = count(Dir == k_i)$

$\sum_{i=1}^{n} CF_i = CF_1 + CF_2 + ... + CF_n$

$Norm.Value = \frac{CF_i}{\sum_{i=1}^{n} CF_i} * 10$

▷ Normalized value

End Procedure

The features that were obtained using the above chain code algorithm are Hole feature ($Hole$) and Outer boundary (OB) feature.

Inner Boundary Feature: $(Hole)$ The binary segmented image is complemented and the connected component is extracted. This gives the co-ordinates of the hole location.

Outer boundary feature: (OB) The chain code is derived or obtained for the outer boundary or shape (Fig. 6.8).

Morphology: The morphological opening was performed using line structuring element (Section 1.4). VL and HL features were extracted to check the presence of vertical line and horizontal line in the image.(Fig. 6.9)[63] [72]

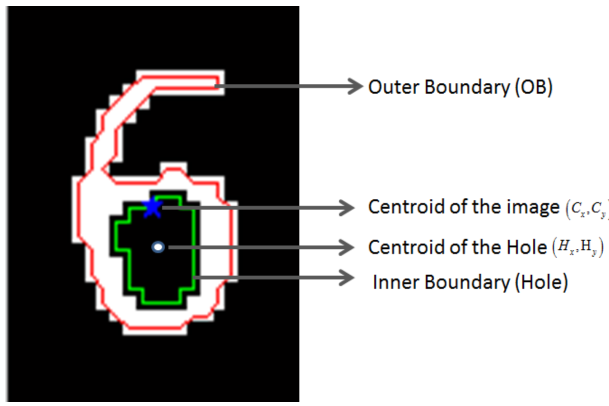

Figure 6.8: Inner and Outer Boundary Features

Figure 6.9: Retained vertical line using Morphological operation

The Feature set (FV_2) contains, Hole co-ordinates $Hole = [H_x, H_y]$, number of holes $Hole_{cnt}$, Centroid of the image (C_x, C_y), Centroid of the hole (CH_x, CH_y), Vertical line using morphological operation VL and end points EP [73]

$$FV_2 = \{Hole_{cnt}, max(H_x), max(H_y), C_x, C_y, CH_x, CH_y, VL, HL\} \quad (6.3)$$

6.4.3 Modified Chain Code Algorithm

Generally the chain code algorithm (Algorithm 10) is used to trace the shape of the boundary for any closed component. Hence the general stopping condition is set that the algorithm stops tracing or encoding the directional value when it reaches the starting pixel again. Sometimes this creates duplicate values when applied for digit images. Hence to avoid this and to extract better feature, a new modified algorithm is proposed.

The Chain Code algorithm (Algorithm 10) is modified in such a way that algorithm stops when it sees the lone pixel as shown. The Fig. 6.10 demonstrates the possible positions of the presence of the pixels. That is when the chain code algorithm is applied to the digit shapes of single pixel width, after travelling forward and encoding the directions, the chain code tends to turn back and stops once it sees the starting pixel again that results in duplicate values.

The digits like 2, 3, 5 and 7 share more similarities in the extracted chain code features. As the main variation lies in the curve path of 2, 3, 5 and 7. Only the forward chain code direction values of the curve paths are extracted.

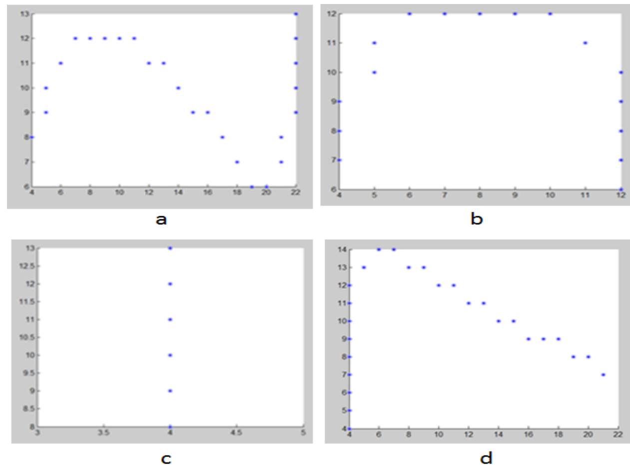

Figure 6.10: Curve path of digits 2,3,5 and 7

A new stopping criteria is used to avoid backward travelling. In Fig. 6.11, as per 8-connected chain code algorithm the presence of lone pixels is found in all the 8 directions. The current direction (NextDir) and previous direction (Dir) is stored. The proposed algorithm represented as in Pseudo Code 9 checks whether the directional values are repeated in the reverse direction. If so it ends the algorithm.

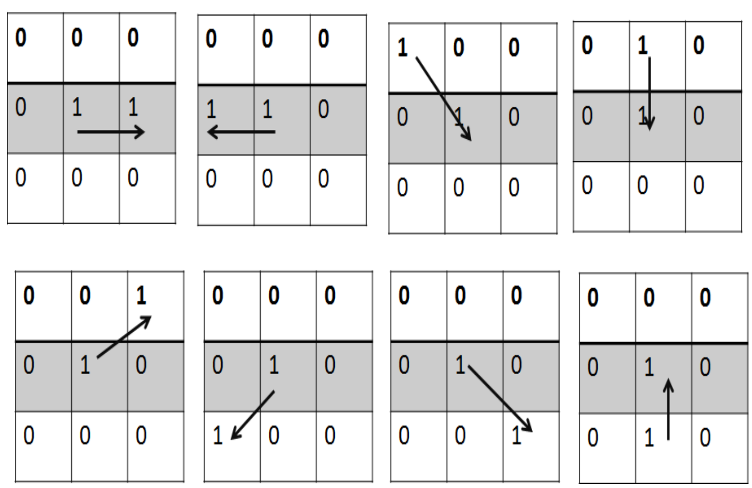

Figure 6.11: Lone Pixel Positions

Pseudo Code 9 Modified chain code

```
Procedure Modified Chain Code
  Next_dir                                          ▷ Current direction
  Dir                                               ▷ Previous direction
  if Dir == 0 && Next_dir == 4||Dir == 4 && Next_dir == 0
  flag = 0
  elseif Dir == 7 && Next_dir == 3||Dir == 3 && Next_dir == 7
  flag = 0
  elseif Dir == 1 && Next_dir == 5||Dir == 5 && Next_dir == 1
  flag = 0
  elseif Dir == 2 && Next_dir == 6||Dir == 6 && Next_dir == 2
  flag = 0
  if flag == 0
  break
End Procedure
```

Thus after obtaining the chain code features, the chain code is normalized using (Algorithm 10). The feature vector FV_3 comprises the normalized directional values for digits (2,3,5 and 7) and classified using K-NN classifier.

$$FV_3 = \{Avg_{len}, DV_i = 0, 1, ...7\} \tag{6.4}$$

6.5 Pin code Recognition

6.5.1 Proposed Hybrid Classification Method

Hybrid Classification method is used classify digit images having holes (0, 4, 6, 9 and 8) using the feature vector (FV_2) [73]. The following steps are applied to the segmented components of the extracted pin code blob.

Algorithm 11:

Step 1: If the hole count is equal to 2 then the digit is classified as 8.

Step 2: Centroids of the image C_x, C_y and hole H_x, H_y are compared for classification of digit 0 or 6 or 9.

Step 3: If $Hole_{cnt} == 1$ and $VL \neq 0$ && $HL \neq 0$. The digit is classified as '4'.

Step 4: If $Hole_{cnt} == 1$ and $C_x == H_x$ && $C_y == H_y$. The digit is classified as '0'.

Step 5: If $Hole_{cnt} == 1$ and $C_x \geq H_x$ && $C_y \geq H_y$. The digit is classified as '6'.

Step 6: If $Hole_{cnt} == 1$ and $C_x \leq H_x$ && $C_y \leq H_y$. The digit is classified as '9'.

Step 7: If $Hole_{cnt} == 0$ and $VL \neq 0$. The digit is '1'.

Step 8: If none of the above conditions matches then Template matching technique is performed for classification of digit image into 2 or 3 or 5 or 7.

The following Fig. 6.12 represents the proposed Hybrid Classifier. The above mentioned Algorithm 11 is represented in Pseudo Code 10.

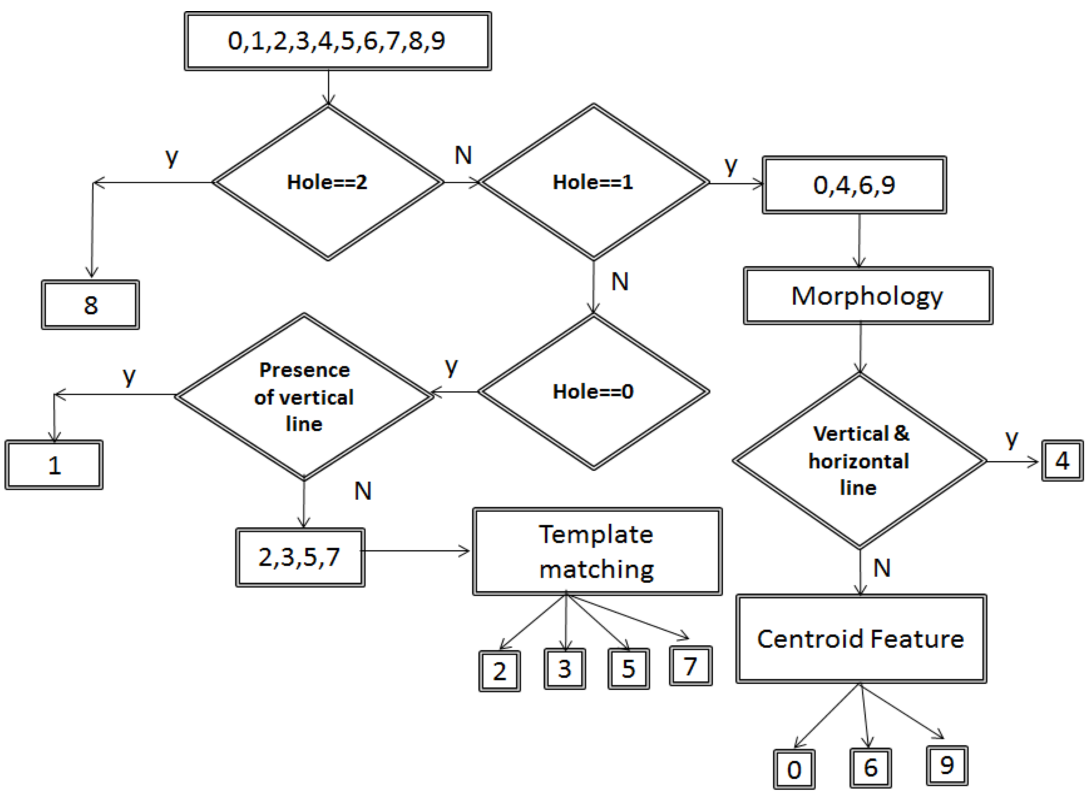

Figure 6.12: Proposed Method

Pseudo Code 10 Hybrid Classifier

Input: $P(x, y)$

Output: $RecognizedDigit$

Procedure Hybrid Classification

$P(x, y) = \{c_1 .... c_6\}$

```
for c = 1 to 6

if c_i contains holes

Hole = [H_x, H_y]

if Hole_cnt == 2

The digit is '8'

end if

if Hole_cnt == 1

if C_x == H_x && C_y == H_y

The digit is '0'

else if C_x ≥ H_x && C_y ≥ H_y

The digit is '6'

else if C_x ≤ H_x && C_y ≤ H_y

The digit is '9'
```

```
else if VL ≠ 0 && HL ≠ 0

The digit is '4'

end if

else if Hole_cnt == 0

if VL ≠ 0

The digit is '1'

else

Apply Template-II to classifiy 2 or 3 or 5 or 7
```

End Procedure

The above method yields an higher accuracy of 97%. The accuracy decreases when the digits are deformed or broken.

6.5.2 KNN

K-nearest neighbour classifier (KNN) is a non-parametric classifier. The new sample is classified to the most common class of the K nearest neighbours. There is no training time for K-NN [110]. However, the testing is time-consuming and has high requirements on memory as it include both the training and test data sets. Distance between the new sample and all the training sample is calculated.

For the proposed method using K-NN classifier, Euclidean distance measure was used to calculate the K nearest neighbours. According to Euclidean distance formula , the distance between two points in the plane with co-ordinates (X, Y) and (A, B) is given by:

$$ED = \sqrt{(X - A)^2 + (Y - B)^2} \tag{6.5}$$

The classification accuracy of KNN is affected by the value of K. If K is small, the noise will have a high influence on the classification result. However, if K is large, the result may not be consistent with the idea behind KNN since KNN focuses on the neighbouring pixels.

To evaluate the performance of KNN, different parameters value for K (K = 1, 3, 5, 7, 9) was used. The highest accuracy of 92% was achieved when K = 5. In the proposed method KNN classifier was used for recognition of digits like 2, 3, 5 and 7. The features were extracted using the modified chain code algorithm (Pseudo Code 9).

6.5.3 Multi-Layer Feed Forward Network

Neural networks are parallel, distributed information processing structures. The feed forward network was used for solving non-linearly separable problems (Fig. 6.13). The feed forward network is referred as MLP. The feed forward network contains nodes that works as computing units and perform information transfer from node to node.

The MLP architecture constitutes Input layer, Hidden layer and Output layer. There can be any number of hidden layers. But, increasing the number of perceptrons in the hidden layer increases the number of weights that must be estimated in the network, which in turn increases the execution time for the network [4]. Training was performed to minimize the training mean square error for all the training patterns. Sigmoid functions was used as the activation function.

Supervised learning was followed since the training of networks was performed by providing known sets of input and output data. The weights were adjusted using the back propagation algorithm until the network gives satisfactory output when provided with the known input [88]. The features (FV_1) were used for the classification of (0 to 9) digits.

$$FV_3 = \{Avg_{len}, DV_i = 0, 1, ...7\} \tag{6.6}$$

$$FV_4 = \{DV_i = 0, 1, ...7\} \tag{6.7}$$

6.5.4 Architecture of BPNN

The MLP works by initializing weights by applying Back Propagation (BP) algorithm. The MLP network was constructed with 8 input neurons, 1 hidden layer and 10 output neurons.

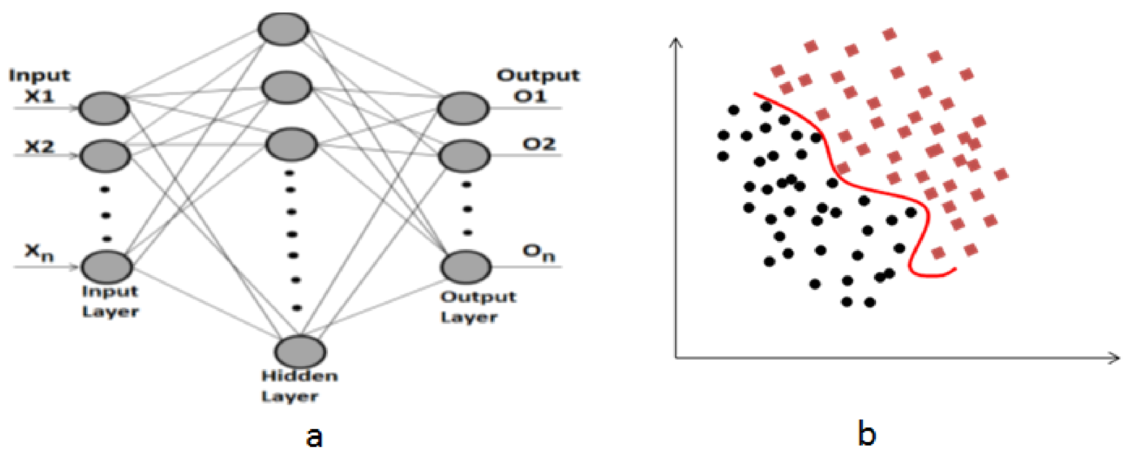

Figure 6.13: a)MLP b)Non-linearly Separable

Back Propagation Algorithm:

Step 1: Each input unit $x_{i(i=1,2,....,n)}$ receives an input signal and transmits this signal to each of the hidden units $h_{j(j=1,2,....,p)}$.

Step 2: Each hidden unit $h_{j(j=1,2,....,p)}$ sums its weighted input signal and apply activation function and send signals to each output unit $o_{k(k=1,2,....,m)}$ which is responsible for transformation of input into output. (vo_j) is the bias on the hidden unit.

$$h_j = f\left(vo_j + \sum_{i=1}^{n} x_i v_{ij}\right) \tag{6.8}$$

Step 3: The output unit $o_{k(k=1,2,....,m)}$ sums its weighted input signals and applies activation function to calculate the output (wo_k) is the bias on the output unit.

Step 4: During back propagation of errors each output unit $o_{k(k=1,2,....,m)}$ receives a target pattern t_k corresponding to an input pattern. Error at the output unit is calculated as follows.

$$\delta_k = (t_k - o_k)\, o_k \tag{6.9}$$

Step 5: Each hidden unit $h_{j(j=1,2,....,p)}$ sums its delta inputs to the output layer and calculates the error in the hidden unit as

$$\delta_j = \sum_{k=1}^{m} \delta_j w_{jk} h_j \tag{6.10}$$

Step 6: Updation of weights and biases are performed using the error factor and the activation function. The output unit and the hidden unit update its weight and bias respectively. Weight and bias update for output unit are as follows:

$$w_{jk}(new) = w_{jk}(old) + \alpha\delta_k z_j, w_{ok}(new) = w_{ok}(old) + \alpha\delta_k \tag{6.11}$$

Step 7: The weight and bias update for the hidden unit as follows:

$$v_{ij}(new) = v_{ij}(old) + \alpha\delta_j x_i, v_{oj}(new) = v_{oj}(old) + \alpha\delta_j \quad (6.12)$$

Step 8: Updation of weights is performed until it reaches the minimal or closer error rate with the destined target.

6.5.5 Naive Bayes Classifier

The Naive Bayes Classifier technique was based on the Bayesian theorem and used when the number of inputs are high. Prediction is performed for the test sample by computing the posterior probability of that sample belonging to each class. The method then classifies the test sample according to the largest posterior probability.

NBC is based on estimating $P(X|Y)$, the probability or probability density of features X given class Y. The NBC uses normal (Gaussian), kernel, multinomial and multivariate multinomial distributions. Different distributions can be used for different features. Kernel Distribution was used in our experiments as the features had continuous values.

The general Bayes rule states that, let C be the class variable and let X_i is the attributes where $i = 1$ to n. The conditional probabilities were calculated using the following formula and the class is chosen based on the maximum posterior probability obtained.

$$P(C|X_1, X_2, ...X_n) = \frac{\left(\pi_{i=1}^{n} P(X_i|C)\right) P(C)}{P(X_1, X_2, ...X_n)} \quad (6.13)$$

The NBC computes a separate kernel density estimate for each class based on the training data for that class. A kernel distribution is a non-parametric representation of the probability density function (PDF) of a random variable. This distribution is defined by a smoothing function and a bandwidth value that controls the smoothness of the resulting density curve. A random partition was performed for holdout validation on n observations. The feature vectors FV_2 obtained using the chain code algorithm was used as features for classification of digits (0 to 9).

6.6 Results and Discussions

The proposed work was tested with MLP, NBC and K-NN classifier. 531 digit samples (0-9) were obtained from 100 address images of IPBME. Normalized Chain codes features were obtained as feature vectors (FV_1) [72].

The MLP network was constructed with 8 input neurons, 1 hidden layer, 40 hidden neurons and 10 output neurons. The network was trained with 400 digit samples and tested with 131 digit samples. It was noted that 95% accuracy was achieved.

NBC was trained with 240 samples and tested with 160 samples. 93.8% accuracy was obtained. The recognition accuracy is given in (Table 6.2). It was observed that NBC obtained higher accuracy in less time compared to MLP-BP. Even when the training samples were increased for NBC it out performed by consuming lesser time. Hence NBC was computationally simple and possessed less execution time compared to MLP-BP.

KNN was trained using 150 samples of digits (2, 3, 5 and 7) using feature vectors FV_3. By setting K=5, 80 digits samples were tested and 92% accuracy was obtained. The graph in (Fig. 6.14) explains the accuracy of above mentioned classifiers.

The resulting recognized pin code is compared with the pin code database. If the pin code is found then it is exported to the excel and converted into 1-D bar code otherwise diverted to visual inspection.

Table 6.2: Experimental Observation

METHODS	TRAINING SET	TESTING SET	ACCURACY
MLP	400	131	95%
Naive Bayes	240	160	93%
KNN	150	80	92%
Hybrid Classification	-	440	97%

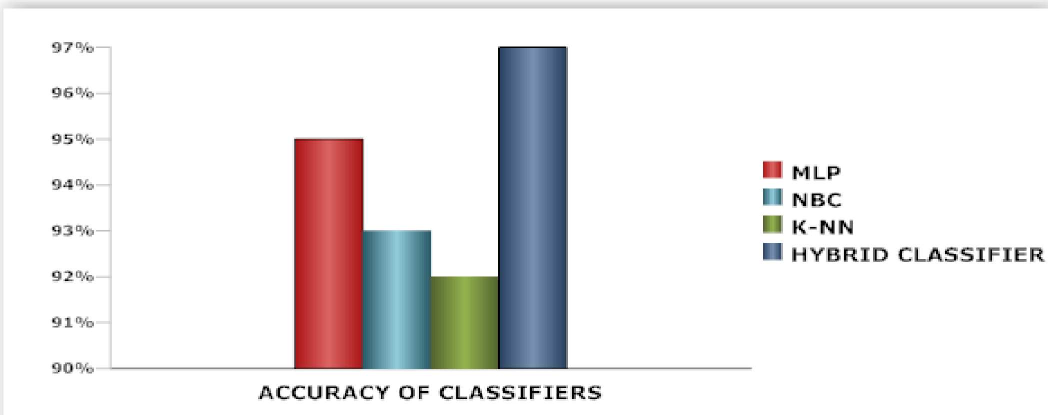

Figure 6.14: Accuracy of Various Classifiers

6.7 Summary

Four approaches were provided for pin code digit recognition. The MLP classifier produced higher accuracy but was expensive and complex in terms of computation and time. Though the NBC classifier is not as good as MLP in generalization, the time and computational complexity were much less.

The simple, cost effective, easy to implement K-NN classifier obtained a higher accuracy of 92% when K=5. But K-NN classifier was little time consuming and the training data set occupied more memory. The training dataset needs to be carefully framed as there was no learning involved other than the euclidean distance measure.

It was observed that the proposed hybrid classification approach combined with Template-II produced higher accuracy when compared with other classifiers. The proposed system was found to be very effective and fast in recognizing the pin code digits.